

Since GATE Overflow (GO) started in August 2014, a lot of people have dedicated their time and effort in bringing this book now. Initiated by Omesh Pandita and Arjun Suresh as a Q/A platform for CSE students, Kathleen Bankson was instrumental in getting all previous year GATE questions here. Then experts like Praveen Saini, Happy Mittal, Sankaranarayanan P.N., Suraj Kumar etc. have contributed a lot to the answers in GO. Pragy Agarwal even after topping GATE has continuously contributed in GO with his knowledge as well as in making the contents beautiful with fine latex skills. We also have to thank the work done by our editors Jotheeswari Arasakumar, Misbah Ghaya, Ishrat Jahan, Nataliyah Ahmad, Kanza Bayad, Andrijana Milicevik and Pavan Singh who are continuously adding and keeping the contents in GO neat and clean. There are also many toppers of GATE 2015-18 who are contributing a lot in GO. The list of all the contributors can be found on GO site but even that does not include the contributions of some like Arif Ali Anapparakkal in helping design this book, Arvind Devaraj and others who have provided guidance and help, Aravind S.R. who helped fixing the page numbers of this book, Resmi A.C. who lend her Macbook for this book making, Marlyn Joseph in providing support for book printing and Akshat Joshi for book design support. Our final proof reading and editing were done with the help of M. Chandrakiruthiga, Pooja Khatri, Shikha Mallick, Manoja Rajalakshmi Aravindakshan, Subarna Das and Sukanya Das.

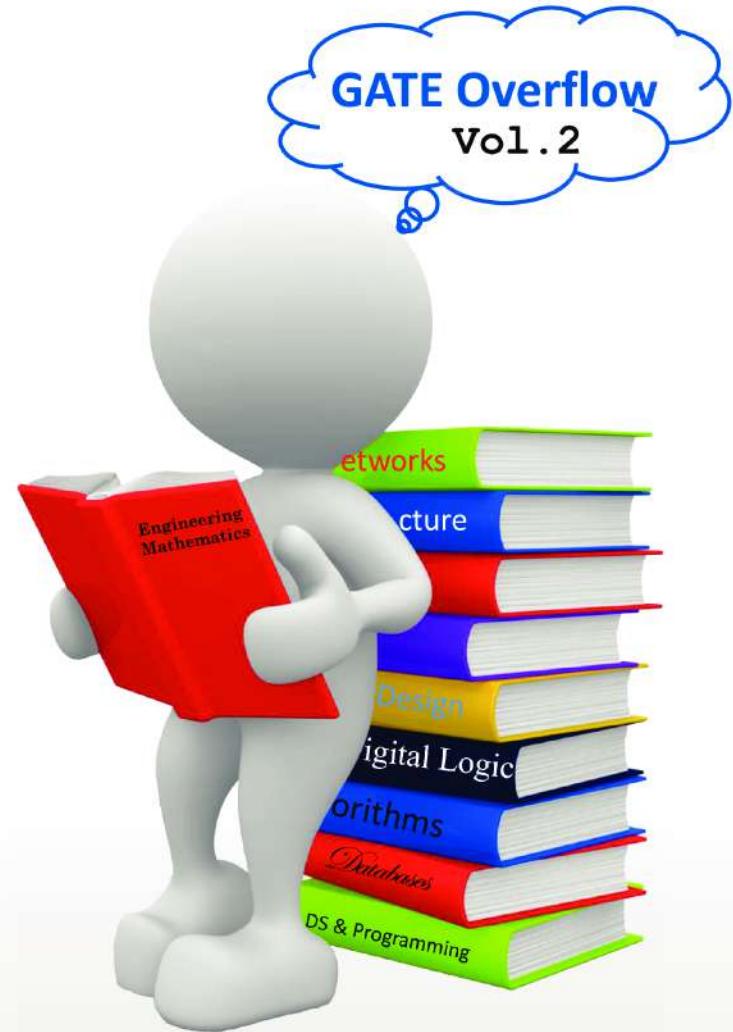
From the GO family, we thank the contributions of Silpa V.S., Digvijay Pandey, Rahul Kumar Yadav and others for getting the GATECSE Last-Rank page maintained. Bikram Ballav is behind most of the exams on GO (<http://mockgate.com>) and our exam interface is contributed by Arindam Sarkar. Pragy Agarwal is also behind the rank and score predictor tool, (<http://mymarks.gatecse.in>) used by GO which has 99-100% accuracy over the last 3 years. Special thanks to Sachin Mittal for making the How to Use GO videos, Silpa V.S. for classifying the questions topic-wise for the book, Pooja Palod for making the GATE 2018 schedule and Debarish Deka, Priti Sharma for GO classroom contributions. We also thank all the GATE toppers who took time to write a review for GO. Finally, on behalf of GO family and all GATE aspirants, we thank the wonderful faculties of IISc., IITs and other premier institutes who are behind the creation of such wonderful questions.



MRP : ₹. 600/-

GATE Overflow for GATECSE 2019 Vol-2

Chackla Printers 0471 2726151



GATE Overflow for GATECSE 2019 Computer Science & IT



Scan me



Scan me



Scan me

This book was created programmatically by GATE Overflow for GATE CSE 2019. This book is having more than 3000 questions which were all from previous year standard exams like GATE, TIFR, CMI and ISI. Our site is having more than 10 times questions but due to difficulty in organizing them and filtering bad ones as of now we do not have them available in any book form but you can see all of them by visiting the [site](#). If you feel any doubt regarding the answer, click on the question link and give a comment on the site. Studying all these questions might get you 60 marks in GATE but that might not be enough for an IIT. So, read [standard books](#) whenever you have a doubt, solve exercise questions and use these questions for cementing the concepts and aim 85+. At least if you are not getting the solution to a given problem first refer standard book and if not [standard videos](#). For more preparation doubts you may see the [FAQ](#) section. If any error is found on any question it shall be updated at <https://gateoverflow.in/corrections>. If you want a hardcopy of this book you can get it from [this link](#).

PDFs for the remaining subjects will be made available at <http://classroom.gateoverflow.in> and you can enroll in [this course](#) to get notification for the same. Enrollment is free and account details are of GATE Overflow with a new password which have been sent to all registered emails on GATE Overflow. New users will receive this email within a day of confirming their email address.

You can now join our [Facebook group](#) for GATE CSE discussions.

You can visit <http://playlists.gatecse.in> for high quality videos for GATE CSE and how to use GO site/ebook.

This book consists of only previous year GATE, TIFR, ISI and CMI questions (CS from 1987 and all 5 years of IT) all of which are relevant for GATE. Out of syllabus subjects as of GATE 2019 are removed from this book except in rare cases.

Since GATE Overflow (GO) started in August 2014, a lot of people have dedicated their time and effort in bringing this book now. Initiated by Omesh Pandita and Arjun Suresh as a Q/A platform for CSE students, Kathleen Bankson was instrumental in getting all previous year GATE questions here. Then experts like Praveen Saini, Happy Mittal, Sankaranarayanan P.N., Suraj Kumar etc. have contributed a lot to the answers here. Pragy Agarwal even after topping GATE has continuously contributed here with his knowledge as well as in making the contents beautiful with fine latex skills. We also have to thank the work done by our editors Jotheeswari Arasakumar, Misbah Ghaya, Ishrat Jahan, Nataliyah Ahmad, Kanza Bayad, Andrijana Milicevik and Pavan Singh who are continuously adding and keeping the contents in GO neat and clean. There are also many toppers of GATE 2015-18 who are contributing a lot in GO. The list of all the contributors can be found [here](#) but even that does not include the contributions of some like Arif Ali Anapparakkal in helping design this book, Arvind Devaraj who has provided as guidance and help, Resmi A.C. for her Macbook which was used for this book making, Aravind S.R. for constant support and help in solving technical issues with our hardcopy, Marlyn Joseph in printing support and Akshat Joshi for page design help. Last but not the least, we thank all the users of GATE Overflow for their contributions and encouragements. The final proof reading and editing of this book were done with the support of M.Chandrakiruthiga, Pooja Khatri, Shikha Mallick, Manoja Rajalakshmi Aravindakshan, Subarna Das and Sukanya Das.

From the GO family we thank the contributions of Silpa V.S., Digvijay Pandey, Rahul Kumar Yadav and others for getting the GATECSE Lastrank page maintained. Bikram Ballav is behind most of the exams on GO (<http://mockgate.com>) and Arindam Sarkar made the interface for it. Pragy Agarwal is also behind the rank and score predictor tool, (<http://mymarks.gatecse.in>) used by GO which has 99-100% accuracy over the last 3 years.

Special thanks to Sachin Mittal for making the How to Use GO videos, Silpa V.S. for classifying the questions topicwise for the book, Pooja Palod for making the GATE 2018 schedule and Debashish Deka and Priti Sharma for GO classroom contributions.

Also thanks to all the GATE toppers who took time to write a review for GO.

Best Books for GATE CSE

Standard books are recommended for a good understanding of Computer Science subjects which is very essential for GATE. For each subject you may choose ANY ONE of the books and not all. You do not need to read every chapter or even every line of a chapter as GATE is not an essay writing examination. What to cover and how to cover for each subject is given in [GO Classroom](#). For full list of recommended standard books please visit [Best books by GATE CSE](#).

Best Videos for GATE CSE

Videos are for those who are not able to follow standard books. Like standard books, to top GATE one must follow standard videos. You should not go behind popular GATE videos which are publicized by marketing guys because nothing can beat quality videos from great professors like Shai Simonson. Also some people try to instigate fear in students that they cannot study from standard videos - again this is just to take your money and fetch you a low rank in GATE. So, if you want to score well in GATE and struggling with some topics, please watch the videos listed in [GO Youtube Channel](#).

Is Coaching necessary for GATE?

I have seen a lot of questions regarding COACHING. It is something which is currently being injected to the blood of students that without coaching success is not possible. When questions are being asked from a predefined text those who practice them a lot can succeed. But GATE is different. Only with proper knowledge and problem solving skills one can crack GATE. In doing this coaching might help a few people. Again coaching alone does not do everything for you to top GATE. You may read [here](#) to know more about how coaching is useful for GATE.

What is GO Classroom for GATE 2019?

Since a lot of aspirants were looking for schedule and materials for GATE 2019, we are adding all the stuffs to GO Classroom (All GO users get GO classroom access once they confirm their email, if you do not have a GO account you can create [here](#)). For GATE 2019, the classroom materials are updated daily based on the live discussions happening at the offline location in Thiruvananthapuram. Materials include, reading part of standard text books, other reference slides/web pages, problems and programming assignments to make the understanding of topics crystal clear. While GO creates a lot of toppers aim of GO Classroom is not to create any loser - all deserving people should get to IITs.

Is there any necessary condition to start GATE preparation?

Only required condition is the dedication and attitude to top GATE. You can read more [here](#).

How to crack GATE?

First of all, do not say that you want to crack GATE. Cracking GATE is very difficult- there are more than 100,000 people trying to do it and only about 2000 seats including NITs. But if you study your Bachelor subjects in the right way, GATE will automatically come to you. No extra preparation is needed. Take a look at GATE syllabus, note down the important ones, and note down the important topics in them. For those subjects at least, use Standard text books, understand the topics well by doing practice problems at the end of the chapters (like doing exercises in Cormen). For practice you should use previous year GATE papers as that tells you how GATE questions are framed. Strictly avoid badly framed test series questions while preparing. For more tips you can see [here](#)

How to use GATE Overflow to top GATE?

Since GATE Overflow was started in 2014, we have added a lot of features and some of these are unknown to many. You may watch [these videos](#) to know about these and how to use GATE Overflow site and PDF in the best way.

What rank I need to get to old IITs?

There are a lot of misconceptions regarding this. As of now one can get direct admits to old IITs upto around 330 GATE rank in general category and around 1000 rank for OBC NCL. But general category people with even above 1500 rank can get to old IITs for Research admits via interviews. This is one more reason why one should prepare using standard resources and do good problems and assignments. You can see the respective tabs [here](#) to know the exact criteria and cutoffs of each IIT and other institutes.

Does GATE Overflow provide test series for GATE CSE?

You can find all the tests on GO [here](#). Under Tests by Mentors, you can find all the tests made by Bikram Ballav which includes 4 full length tests and 2 subject wise tests for each subject. Under GATE Overflow tests we will be adding subjectwise tests as well as full length tests for ensuring every important point is covered. For more details on tests you may refer [here](#).

Table of Contents

Table of Contents	1
Contributors	5
1 Algorithms (323)	6
1.1 Algorithm Design(8)	6
Answers: Algorithm Design	8
1.2 Algorithm Design Techniques(5)	12
Answers: Algorithm Design Techniques	13
1.3 Asymptotic Notations(22)	15
Answers: Asymptotic Notations	22
1.4 Dynamic Programming(12)	32
Answers: Dynamic Programming	36
1.5 Graph Algorithms(48)	43
Answers: Graph Algorithms	60
1.6 Graph Connectivity(1)	80
Answers: Graph Connectivity	80
1.7 Greedy Algorithm(8)	81
Answers: Greedy Algorithm	83
1.8 Huffman Code(2)	88
Answers: Huffman Code	89
1.9 Identify Function(37)	90
Answers: Identify Function	106
1.10 Minimum Maximum(4)	120
Answers: Minimum Maximum	121
1.11 Minimum Spanning Trees(2)	123
Answers: Minimum Spanning Trees	124
1.12 Numerical Computation(2)	125
Answers: Numerical Computation	125
1.13 P Np Npc Nph(12)	126
Answers: P Np Npc Nph	130
1.14 Recurrence(32)	134
Answers: Recurrence	144
1.15 Searching(8)	159
Answers: Searching	162
1.16 Sorting(55)	167
Answers: Sorting	182
1.17 Spanning Tree(31)	204
Answers: Spanning Tree	214
1.18 Time Complexity(34)	228
Answers: Time Complexity	240
2 Compiler Design (175)	255
2.1 Abstract Syntax Tree(1)	255
Answers: Abstract Syntax Tree	255
2.2 Assembler(6)	255
Answers: Assembler	257
2.3 Code Optimization(4)	258
Answers: Code Optimization	259
2.4 Compilation Phases(9)	261
Answers: Compilation Phases	264
2.5 Expression Evaluation(2)	267
Answers: Expression Evaluation	268
2.6 Grammar(39)	269
Answers: Grammar	282
2.7 Intermediate Code(8)	299
Answers: Intermediate Code	301
2.8 Left Recursion(1)	305
Answers: Left Recursion	306
2.9 Lexical Analysis(6)	307

Answers: Lexical Analysis	308
2.10 Linking(3)	310
Answers: Linking	311
2.11 Live Variable(1)	312
Answers: Live Variable	312
2.12 Macros(3)	313
Answers: Macros	314
2.13 Parameter Passing(14)	314
Answers: Parameter Passing	320
2.14 Parsing(43)	325
Answers: Parsing	341
2.15 Register Allocation(2)	359
Answers: Register Allocation	360
2.16 Runtime Environments(18)	361
Answers: Runtime Environments	367
2.17 Static Single Assignment(2)	372
Answers: Static Single Assignment	373
2.18 Syntax Directed Translation(6)	374
Answers: Syntax Directed Translation	376
2.19 Target Code Generation(4)	379
Answers: Target Code Generation	380
2.20 Variable Scope(2)	382
Answers: Variable Scope	384
2.21 Viable Prefix(1)	384
Answers: Viable Prefix	385
3 Programming & DS: DS (208)	386
3.1 Abstract Data Type(1)	386
Answers: Abstract Data Type	386
3.2 Arrays(14)	386
Answers: Arrays	392
3.3 Binary Search Tree(29)	399
Answers: Binary Search Tree	407
3.4 Binary Tree(50)	420
Answers: Binary Tree	435
3.5 Graph Search(1)	455
Answers: Graph Search	455
3.6 Graphs(6)	456
Answers: Graphs	457
3.7 Hashing(17)	460
Answers: Hashing	466
3.8 Heap(24)	473
Answers: Heap	480
3.9 Infix Postfix(2)	489
Answers: Infix Postfix	490
3.10 Linked Lists(20)	491
Answers: Linked Lists	498
3.11 Priority Queue(1)	507
Answers: Priority Queue	507
3.12 Queues(12)	507
Answers: Queues	512
3.13 Stack(16)	517
Answers: Stack	523
3.14 Trees(15)	529
Answers: Trees	534
4 Programming & DS: Programming (104)	542
4.1 Aliasing(1)	542
Answers: Aliasing	542
4.2 Goto(2)	542
Answers: Goto	543
4.3 Identify Function(4)	543

Answers: Identify Function	545
4.4 Loop Invariants(8)	547
Answers: Loop Invariants	550
4.5 Parameter Passing(2)	554
Answers: Parameter Passing	555
4.6 Programming Constructs(1)	556
Answers: Programming Constructs	557
4.7 Programming In C(62)	557
Answers: Programming In C	586
4.8 Programming Paradigms(2)	617
Answers: Programming Paradigms	618
4.9 Pseudo Code(1)	619
Answers: Pseudo Code	619
4.10 Recursion(18)	620
Answers: Recursion	627
4.11 Structures(1)	637
Answers: Structures	638
4.12 Type Checking(1)	638
Answers: Type Checking	638
4.13 Variable Binding(1)	639
Answers: Variable Binding	639
5 Theory of Computation (281)	641
5.1 Closure Property(10)	641
Answers: Closure Property	643
5.2 Context Free Language(31)	646
Answers: Context Free Language	656
5.3 Countable Set(1)	667
Answers: Countable Set	667
5.4 Decidability(23)	667
Answers: Decidability	674
5.5 Finite Automata(65)	682
Answers: Finite Automata	704
5.6 Grammar(2)	734
Answers: Grammar	735
5.7 Identify Class Language(33)	736
Answers: Identify Class Language	745
5.8 Minimal State Automata(2)	756
Answers: Minimal State Automata	757
5.9 Non Determinism(7)	757
Answers: Non Determinism	759
5.10 P Np Npc Nph(5)	761
Answers: P Np Npc Nph	763
5.11 Pumping Lemma(1)	765
Answers: Pumping Lemma	765
5.12 Pushdown Automata(13)	765
Answers: Pushdown Automata	769
5.13 Recursive And Recursively Enumerable Languages(13)	775
Answers: Recursive And Recursively Enumerable Languages	779
5.14 Regular Expressions(27)	783
Answers: Regular Expressions	790
5.15 Regular Grammar(3)	799
Answers: Regular Grammar	800
5.16 Regular Languages(37)	802
Answers: Regular Languages	813
5.17 Turing Machine(8)	827
Answers: Turing Machine	830

Contributors

User	Answers	User	Added	User	Done
Arjun Suresh	6217,196	Kathleen Bankson	441	kenzou	474
Praveen Saini	2037,69	Jotheeswari	210	Milicevic3306	367
Gate Keeda	1312,60	makhdoom ghaya	129	Shikha Mallick	86
Akash Kanase	786,38	Ishrat Jahan	120	Arjun Suresh	43
Prashant Singh	592,32	Arjun Suresh	58	Pooja Khatri	27
Vikrant Singh	536,15	gatecse	35	Subarna Das	14
Amar Vashishth	508,20	Rucha	32	Puja Mishra	11
Bhagirathi Nayak	505,22	Shelke	25	KritiGiga2101	9
Rajarshi Sarkar	495,28	Sandeep Singh	24	Digvijay	7
Pragy	452,17	Akash Kanase	11	Prashant Singh	7
Agarwal		Madhav kumar	8		
gatecse		khush tak			
Rajesh	444,12				
Pradhan	421,21				
Sankaranarayanan P.N	405,20				
Pooja	383,18				
Palod					
Kalpana Bhargav	371,11				
Digvijay	338,20				
Debabish Deka	334,9				
srestha	277,23				
Abhilash Panicker	262,6				
Ankit	242,10				
Rokde					
Manu Thakur	237,10				
Umang Raman	218,14				
Anurag	214,8				
Semwal					
Jayendra	209,10				
Mithlesh Upadhyay	201,7				
Monanshi Jain	194,8				
Anoop	191,6				
Sonkar					
Ahwan Mishra	186,10				
sonam vyas	184,7				
Himanshu	172,7				
Agarwal					
Sandeep_Uniyal	168,7				
Keith	167,7				
Kr					
Kalpish Singhal	150,4				
Sachin	150,5				
Mittal					
Aditi Dan	141,5				
sriv_shubham	141,5				
pC	140,6				
Srinath Jayachandran	129,2				
ryan sequira	129,4				
suraj	127,5				
Manoj Kumar	126,9				
Danish	123,5				
Shashank Chavan	118,3				
Sona Praneeth	118,5				
Akula					
neha pawar	107,5				
Arnabi	101,4				
Bej					
Anu	97,4				
Happy	97,5				
Mittal					
Shyam Singh	96,2				
Ashish Deshmukh	95,2				
Vicky	94,4				
Bajoria					
Shreyans Dhankhar	92,3				
anshu	92,5				
gate_asp	91,3				
Riya Roy(Arayana)	86,5				
Muktinath Vishwakarma	85,6				
Prasanna Ranganathan	84,4				
papesh	83,6				
Harsh181996	78,2				
sumit kumar singh dixit	78,4				
Shaun Patel	75,2				
2018	75,4				
saurabhrk	70,2				
kirti singh	70,5				
Ashis Kumar Sahoo	66,3				
Sourav Basu	65,1				
rahulkr	64,2				
Viral Kapoor	64,2				
Prajwal Bhat	64,4				

1

Algorithms (323) top

1.1

Algorithm Design(8) top

1.1.1 Algorithm Design: CMI2010 - 6 top

<https://gateoverflow.in/215034>

You are given a list of positive integers along with a sequence of operations from the set $\{*, +\}$. You construct expressions from these two lists so that:

- The numbers in the expression are drawn from the first list, without repetition and without altering their order.
- All the operators in the second list are used in the expression, in the same order.

For example, if the two lists are $[1, 3, 2, 1, 4]$ and $['*', '+']$ the set of possible expressions you can form are

$1 * 3 + 2, 1 * 3 + 1, 1 * 3 + 4, 1 * 2 + 1, 1 * 2 + 4, \dots, 2 * 1 + 4, 1 * 3 + 2, 1 * 3 + 1, 1 * 3 + 4, 1 \dots$
For each expression, the value is computed by bracketing operators from the right. That is, the expressions above are evaluated as

$1 * (3 + 2), 1 * (3 + 1), 1 * (3 + 4), 1 * (2 + 1), 1 * (2 + 4), \dots, 2 * (1 + 4), 1 * (3 + 2), 1 * (3 + 1)$
The aim is to determine maximum value among these expressions. In this example, the maximum value is 18, from the expression $3 * 2 + 4$, which is evaluated as $3 * (2 + 4) = 3 * 6 = 18$.

You may assume that the length of the first list is more than the length of the second list.
Describe an algorithm to solve this problem.

[algorithms](#) [descriptive](#) [cmi2010](#) [algorithm-design](#)**Answer**

1.1.2 Algorithm Design: CMI2012-B-03a top

<https://gateoverflow.in/46563>

Let A be an array of n integers, sorted, so that $A[1] \leq A[2] \leq \dots \leq A[n]$. Suppose you are given a number x and you wish to find out if there are indices k and l such that $A[k] + A[l] = x$.

- Design an $O(n \log n)$ time algorithm for this problem.

[cmi2012](#) [descriptive](#) [algorithms](#) [algorithm-design](#)**Answer**

1.1.3 Algorithm Design: CMI2012-B-03b top

<https://gateoverflow.in/47088>

Let A be an array of n integers, sorted so that $A[1] \leq A[2] \leq \dots \leq A[n]$. Suppose you are given a number x and you wish to find out if there exist indices k and l such that $A[k] + A[l] = x$.

- Design an $O(n)$ algorithm for this problem.

[descriptive](#) [cmi2012](#) [algorithms](#) [algorithm-design](#)**Answer**

1.1.4 Algorithm Design: GATE1992-8 top

<https://gateoverflow.in/587>

Let T be a Depth First Tree of an undirected graph G . An array P indexed by the vertices of G is given. $P[V]$ is the parent of vertex V , in T . Parent of the root is the root itself.

Give a method for finding and printing the cycle formed if the edge (u, v) of G not in T (i.e., $e \in G - T$) is now added to T .

Time taken by your method must be proportional to the length of the cycle.

Describe the algorithm in a PASCAL (C) – like language. Assume that the variables have been suitably declared.

gate1992 | algorithms | descriptive | algorithm-design

[Answer](#)

1.1.5 Algorithm Design: GATE2006-17 [top](#)

<https://gateoverflow.in/978>

An element in an array X is called a leader if it is greater than all elements to the right of it in X . The best algorithm to find all leaders in an array

- A. solves it in linear time using a left to right pass of the array
- B. solves it in linear time using a right to left pass of the array
- C. solves it using divide and conquer in time $\Theta(n \log n)$
- D. solves it in time $\Theta(n^2)$

gate2006 | algorithms | normal | algorithm-design

[Answer](#)

1.1.6 Algorithm Design: GATE2006-54 [top](#)

<https://gateoverflow.in/1832>

Given two arrays of numbers a_1, \dots, a_n and b_1, \dots, b_n where each number is 0 or 1, the fastest algorithm to find the largest span (i, j) such that $a_i + a_{i+1} + \dots + a_j = b_i + b_{i+1} + \dots + b_j$ or report that there is not such span,

- A. Takes $O(3^n)$ and $\Omega(2^n)$ time if hashing is permitted
- B. Takes $O(n^3)$ and $\Omega(n^{2.5})$ time in the key comparison mode
- C. Takes $\Theta(n)$ time and space
- D. Takes $O(\sqrt{n})$ time only if the sum of the $2n$ elements is an even number

gate2006 | algorithms | normal | algorithm-design | time-complexity

[Answer](#)

1.1.7 Algorithm Design: ISI2015-CS-2a [top](#)

<https://gateoverflow.in/47271>

You are given two strings S and T , each of length α , consisting only of lower case English letters (a, b, \dots, z). Propose an $O(\alpha)$ -time algorithm to decide whether S can be obtained by permuting the symbols of T . For example, if $S = \text{algorithm}$, $T = \text{logarithm}$, your algorithm should return YES; but if $S = \text{trainee}$, $T = \text{retinaa}$, your algorithm should return NO.

descriptive | isi2015 | algorithms | algorithm-design

[Answer](#)

1.1.8 Algorithm Design: TIFR2011-B-29 [top](#)

<https://gateoverflow.in/20576>

You are given ten rings numbered from 1 to 10, and three pegs labeled A , B , and C . Initially all the rings are on peg A , arranged from top to bottom in ascending order of their numbers. The goal is to move all the rings to peg B in the minimum number of moves obeying the following constraints:

- i. In one move, only one ring can be moved.
- ii. A ring can only be moved from the top of its peg to the top of a new peg.
- iii. At no point can a ring be placed on top of another ring with a lower number.

How many moves are required?

- A. 501
- B. 1023
- C. 2011
- D. 10079
- E. None of the above.

[tifr2011](#) | [algorithms](#) | [algorithm-design](#)

[Answer](#)

Answers: Algorithm Design

1.1.1 Algorithm Design: CMI2010 - 6 top

<https://gateoverflow.in/215034>



Selected Answer

Suppose the first list has m numbers and the second list has n operators, here ($m > n$).

So, at each expression we will have $(n + 1)$ numbers.

Note that maximum value of the expression occurs when $(n + 1)$ maximum numbers out of m numbers in the first list will be present in the expression.

So, our goal is to find $(n + 1)$ maximum numbers out of m numbers in the first list.

Now, finding the maximum number out of m numbers in the first list can be done in $O(m)$ time.

Build a max heap and extract the root node. Building a max heap takes $O(n)$ time.

Now, the remaining n maximum numbers can be determined in $O(\log n)$ time. (Call max-heapify on the max-heap and delete the root node).

Note that here n is a constant number of operators in the second list.

So, to find $(n + 1)$ maximum numbers out of m numbers in the first list can be done in $n + O(\log n)$ time.

So, to determine the maximum value among all the expressions can be done in $n + O(\log n)$ time..

1 votes

-- Kushagra Chatterjee (8.2k points)

1.1.2 Algorithm Design: CMI2012-B-03a top

<https://gateoverflow.in/46563>



Selected Answer

a. Pseudo code for time complexity $O(n \log n)$:

```
for (i=1; i<=n; i++)
{
```

```

    if (binary_search( x- A[i] ) // O(logn)
        return true;
}
return false;

```

b. Pseudo code for time complexity $O(n)$:

```

for ( k=1, l=n; k<l ; )
{
    temp =A[k] + A[l];
    if (temp== x)
        return true;
    else if(temp > x)
        l--;
    else
        k++;
}
return false;

```

👍 2 votes

-- Dhananjay Kumar Sharma (25.7k points)

1.1.3 Algorithm Design: CMI2012-B-03b top

<https://gateoverflow.in/47088>



Selected Answer

Here is the algorithm, which returns **True** if there is a number **x** present such that (**A[k] + A[l] == x**) else returns **False**.

```

// Consider this algorithm is called as
// AlgorithmCheck(A,l,size,x);
// Where A is the sorted array

Bool AlgorithmCheck(A, L, K, x) {
    while(L<K) {
        if(A[L]+A[K] == x)
            return true;
        else if(A[L]+A[K] < x)
            L++;
        else
            K--;
    }
    return false;
}

```

This will take only **O(n)** time to do its work.

👍 15 votes

-- Muktinath Vishwakarma (35.4k points)

1.1.4 Algorithm Design: GATE1992-8 top

<https://gateoverflow.in/587>



Selected Answer

Union-Find Algorithm can be used to find the cycle.

Ref: <http://www.geeksforgeeks.org/union-find/>

 10 votes

-- Rajarshi Sarkar (34.1k points)

1.1.5 Algorithm Design: GATE2006-17

<https://gateoverflow.in/978>



Option B) We can move from right to left, while keeping a note of the maximum element so far (let's call it `current_max`).

- Starting from the rightmost element, we initialize our `current_max` with it, since the rightmost element will always be a leader.
 - Moving from right to left, if an element x is greater than our `current_max`, then x is also a leader. Add this element to list of leaders (or simply print it). Set `current_max` to x and carry-on leftward.

Time Complexity would be $\Theta(n)$.

 33 votes

-- Madhur Rawat (2.3k points)

1.1.6 Algorithm Design: GATE2006-54

<https://gateoverflow.in/1832>



Answer is (C). Following algorithm would do.

Since array is binary, the max sum will go until n and so the sum difference of the two arrays can vary between $-n$ and n . We use array $start$ to keep the starting index of each possible sum (hence of size $2n + 1$) and array end to keep the ending index (these two arrays work like hash tables and since we have only $2n + 1$ possible keys, we can do a perfect hashing). So, our required solution will be $\max(end[i] - start[i])$ provided both are assigned values.

The algorithm works as follows:

1. Initialize $diff$ array to contain the difference of sum of elements of array a and b . i.e., $diff[i] = \sum_{i=0}^n a[i] - b[i]$.
 2. Now $diff[i]$ can have values from $-n$ to n which gives $2n + 1$ possible values and the first occurrence of a $diff$ value marks the beginning of a span and the last occurrence marks the end. We use $start$ and end array for storing these two positions for the $2n + 1$ possible values.
 3. Now, the largest value of $end[i] - start[i]$ for any i , will be the largest span and the start of it will be $start[i] + 1$, and end will be $end[i]$. If the span is starting from first position itself (arrays a and b have same first elements), then it will start from $start[i]$ itself.

```
#include <stdio.h>

#define size 100 //assume n is less than 100
int main()
{
    int n, a[size], b[size];
    int start[2*size+1], end[2*size+1];
    int sum1 = 0, sum2 = 0, i;
    int diff[size];
    printf("Enter n: ");
    scanf("%d", &n);
    for(i = 0; i < n; i++)
    {
        printf("Enter a[%d]: ", i);
        scanf("%d", &a[i]);
    }
}
```

```

for(i = 0; i < n; i++)
{
    printf("Enter b[%d]: ", i);
    scanf("%d", &b[i]);
}

for(i = 0; i < n; i++)
{
    if(a[i]) sum1++;
    if(b[i]) sum2++;
    diff[i] = sum1 - sum2;
}

for(i = 0; i < 2*n; i++)
{
    start[i] = -1, end[i] = -1;
}
start[n] = end[n] = 0;
//initially sum is 0 at the beginning of array and
//the first n-1 elements of start and end are used
//if sum of A till ith element is less than sum of B till ith element
for(i=0; i < n; i++)
{
    if(start[diff[i] + n] == -1) //interested only in the first occurrence of diff[i]
        start[diff[i] + n] = i;
    end[diff[i] + n] = i; //interested in the last occurrence of diff[i]
}

int max = -1;
int savei = -1; //savei is for storing the sum having the largest span

for(i = 0; i < 2*n; i++)
{
    if(start[i] > -1 && (end[i] - start[i] > max))
    {
        max = end[i] - start[i];
        savei = i;
    }
}

if(savei >= 0)
{
    printf("The largest span is from %d to %d\n", start[savei]+(savei != n), end[savei]);
    //when sum zero is having the largest span, span starts from first element itself.
    //Else, the span starts from the next element from which the span does not change
}
else
{
    printf("No span\n");
}
}

```

20 votes

-- Arjun Suresh (350 points)

1.1.7 Algorithm Design: ISI2015-CS-2a top

<https://gateoverflow.in/47271>


Selected Answer

- 1) Take an array of 26 elements where 'a' corresponds to index 0,...,z corresponds to index 25, initially initialized to all 0s
- 2) For the first string, increment the corresponding positions of the array by 1 corresponding to every character as the string is traversed.
- 3) For the second string, decrement the corresponding positions of the array by 1 corresponding to every character as the string is traversed.
- 4) If all the elements of the array are 0 finally, then output YES, else NO.

0 votes

-- Prasita Mukherjee (429 points)

1.1.8 Algorithm Design: TIFR2011-B-29 top

<https://gateoverflow.in/20576>


Selected Answer

I think its Tower of Hanoi problem.
Therefore, total number of function call $2^n - 1 = 1023$.
∴ Option B.

8 votes

-- Umang Raman (15.7k points)

1.2**Algorithm Design Techniques(5)** top**1.2.1 Algorithm Design Techniques: GATE1990-2-vii** top

Match the pairs in the following questions:

- | | |
|--|-------------------------|
| (a) Strassen's matrix multiplication algorithm | (p) Greedy method |
| (b) Kruskal's minimum spanning tree algorithm | (q) Dynamic programming |
| (c) Biconnected components algorithm | (r) Divide and Conquer |
| (d) Floyd's shortest path algorithm | (s) Depth first search |

gate1990 match-the-following algorithms algorithm-design-techniques

Answer

<https://gateoverflow.in/8399>

1.2.2 Algorithm Design Techniques: GATE1997-1.5 top <https://gateoverflow.in/2221>

The correct matching for the following pairs is

- | | |
|---------------------------------|------------------------|
| A. All pairs shortest path | 1. Greedy |
| B. Quick Sort | 2. Depth-First Search |
| C. Minimum weight spanning tree | 3. Dynamic Programming |
| D. Connected Components | 4. Divide and Conquer |
| A. A-2 B-4 C-1 D-3 | |
| B. A-3 B-4 C-1 D-2 | |
| C. A-3 B-4 C-2 D-1 | |
| D. A-4 B-1 C-2 D-3 | |

gate1997 algorithms normal algorithm-design-techniques

Answer

1.2.3 Algorithm Design Techniques: GATE2015-1-6 top <https://gateoverflow.in/8088>

Match the following:

- | | |
|---|--------------------------|
| P. Prim's algorithm for minimum spanning tree | i. Backtracking |
| Q. Floyd-Warshall algorithm for all pairs shortest path | ii. Greedy method |
| R. Merge sort | iii. Dynamic programming |
| S. Hamiltonian circuit | iv. Divide and conquer |
| A. P-iii, Q-ii, R-iv, S-i | |
| B. P-i, Q-ii, R-iv, S-iii | |
| C. P-ii, Q-iii, R-iv, S-i | |
| D. P-ii, Q-i, R-iii, S-iv | |

gate2015-1 algorithms normal algorithm-design-techniques

Answer

1.2.4 Algorithm Design Techniques: GATE2015-2-36 [top](#)

<https://gateoverflow.in/8161>

Given below are some algorithms, and some algorithm design paradigms.

1. Dijkstra's Shortest Path
2. Floyd-Warshall algorithm to compute all pairs shortest path
3. Binary search on a sorted array
4. Backtracking search on a graph

- i. Divide and Conquer
- ii. Dynamic Programming
- iii. Greedy design
- iv. Depth-first search
- v. Breadth-first search

Match the above algorithms on the left to the corresponding design paradigm they follow.

- A. 1-i, 2-iii, 3-i, 4-v
- B. 1-iii, 2-iii, 3-i, 4-v
- C. 1-iii, 2-ii, 3-i, 4-iv
- D. 1-iii, 2-ii, 3-i, 4-v

gate2015-2 algorithms easy algorithm-design-techniques

Answer

1.2.5 Algorithm Design Techniques: GATE2017-1-05 [top](#)

<https://gateoverflow.in/118707>

Consider the following table:

Algorithms	Design Paradigms
P. Kruskal	i. Divide and Conquer
Q. Quicksort	ii. Greedy
R. Floyd-Warshall	iii. Dynamic Programming

Match the algorithms to the design paradigms they are based on.

- A. $(P) \leftrightarrow (ii), (Q) \leftrightarrow (iii), (R) \leftrightarrow (i)$
- B. $(P) \leftrightarrow (iii), (Q) \leftrightarrow (i), (R) \leftrightarrow (ii)$
- C. $(P) \leftrightarrow (ii), (Q) \leftrightarrow (i), (R) \leftrightarrow (iii)$
- D. $(P) \leftrightarrow (i), (Q) \leftrightarrow (ii), (R) \leftrightarrow (iii)$

gate2017-1 algorithms algorithm-design-techniques

Answer

Answers: Algorithm Design Techniques

1.2.1 Algorithm Design Techniques: GATE1990-2-vii top



Selected Answer

<https://gateoverflow.in/83991>

- (a) Strassen's matrix multiplication algorithm - (r) Divide and Conquer
- (b) Kruskal's minimum spanning tree algorithm - (p) Greedy method
- (c) Biconnected components algorithm - (s) Depth first search
- (d) Floyd's shortest path algorithm - (q) Dynamic programming

 6 votes

-- vishwa ratna (2.5k points)

1.2.2 Algorithm Design Techniques: GATE1997-1.5 top <https://gateoverflow.in/2221>



Selected Answer

Answer : B) A-3 B-4 C-1 D-2

- | | |
|----------------------------------|-------------------------|
| (A) All pairs shortest path | (3) Dynamic Programming |
| (B) Quick Sort | (4) Divide and Conquer |
| (C) Minimum weight spanning tree | (1) Greedy |
| (D) Connected Components | (2) Depth-First Search |

Reference : Read the Intro/Algo Sub-Heading.

- https://en.wikipedia.org/wiki/Floyd%20Warshall_algorithm#History_and_naming
- <https://en.wikipedia.org/wiki/Quicksort#Algorithm>
- https://en.wikipedia.org/wiki/Minimum_spanning_tree#Algorithms
- [https://en.wikipedia.org/wiki/Connected_component_\(graph_theory\)#Algorithms](https://en.wikipedia.org/wiki/Connected_component_(graph_theory)#Algorithms)

 7 votes

-- Siddharth Mahapatra (1.3k points)

1.2.3 Algorithm Design Techniques: GATE2015-1-6 top <https://gateoverflow.in/8088>



Selected Answer

P. Prim - ii. Greedy

<http://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/>

Q. Floyd Warshall - iii. Dynamic

<http://www.geeksforgeeks.org/dynamic-programming-set-16-floyd-warshall-algorithm/>

R. Merge Sort - iv. Divide & Conquer

<http://www.geeksforgeeks.org/merge-sort/>

S. Hamiltonian Circuit - i. Backtracking

<http://www.geeksforgeeks.org/backtracking-set-7-hamiltonian-cycle/>

Option is C.

 9 votes

-- Pronomita Dey (2.1k points)

1.2.4 Algorithm Design Techniques: GATE2015-2-36 [top](#)



Selected Answer

<https://gateoverflow.in/8161>

Answer: C

17 votes

-- Rajarshi Sarkar (34.1k points)

1.2.5 Algorithm Design Techniques: GATE2017-1-05 [top](#)



Selected Answer

<https://gateoverflow.in/118707>

In **Kruskal**, in every iteration, an edge of the **most minimum weight (greediest)** possible is selected and added to MST construction. Hence, **greedy**.

In **Quick Sort**, we partition the problem into subproblems, solve them and then combine. Hence, it is **Divide & Conquer**.

Floyd-Warshall uses Dynamic programming.

Hence correct answer is : **OPTION (C)**.

17 votes

-- sriv_shubham (3.3k points)

1.3

Asymptotic Notations(22) [top](#)

1.3.1 Asymptotic Notations: GATE1994-1.23 [top](#)

<https://gateoverflow.in/2466>

Consider the following two functions:

$$g_1(n) = \begin{cases} n^3 & \text{for } 0 \leq n \leq 10,000 \\ n^2 & \text{for } n \geq 10,000 \end{cases}$$

$$g_2(n) = \begin{cases} n & \text{for } 0 \leq n \leq 100 \\ n^3 & \text{for } n > 100 \end{cases}$$

Which of the following is true?

- A. $g_1(n)$ is $O(g_2(n))$
- B. $g_1(n)$ is $O(n^3)$
- C. $g_2(n)$ is $O(g_1(n))$
- D. $g_2(n)$ is $O(n)$

[gate1994](#) [algorithms](#) [asymptotic-notations](#) [normal](#)

Answer

1.3.2 Asymptotic Notations: GATE1996-1.11 [top](#)

<https://gateoverflow.in/2715>

Which of the following is false?

- A. $100n \log n = O(\frac{n \log n}{100})$
- B. $\sqrt{\log n} = O(\log \log n)$
- C. If $0 < x < y$ then $n^x = O(n^y)$
- D. $2^n \neq O(nk)$

gate1996 algorithms asymptotic-notations normal

Answer

1.3.3 Asymptotic Notations: GATE1999-2.21 [top](#)

<https://gateoverflow.in/1498>

If $T_1 = O(1)$, give the correct matching for the following pairs:

- | | |
|--------------------------------|-------------------------|
| (M) $T_n = T_{n-1} + n$ | (U) $T_n = O(n)$ |
| (N) $T_n = T_{n/2} + n$ | (V) $T_n = O(n \log n)$ |
| (O) $T_n = T_{n/2} + n \log n$ | (W) $T = O(n^2)$ |
| (P) $T_n = T_{n-1} + \log n$ | (X) $T_n = O(\log^2 n)$ |

- A. M-W, N-V, O-U, P-X
 B. M-W, N-U, O-X, P-V
 C. M-V, N-W, O-X, P-U
 D. M-W, N-U, O-V, P-X

gate1999 algorithms recurrence asymptotic-notations normal

Answer

1.3.4 Asymptotic Notations: GATE2000-2.17 [top](#)

<https://gateoverflow.in/664>

Consider the following functions

- $f(n) = 3n^{\sqrt{n}}$
- $g(n) = 2^{\sqrt{n}} \log n$
- $h(n) = n!$

Which of the following is true?

- A. $h(n)$ is $O(f(n))$
 B. $h(n)$ is $O(g(n))$
 C. $g(n)$ is not $O(f(n))$
 D. $f(n)$ is $O(g(n))$

gate2000 algorithms asymptotic-notations normal

Answer

1.3.5 Asymptotic Notations: GATE2001-1.16 [top](#)

<https://gateoverflow.in/709>

Let $f(n) = n^2 \log n$ and $g(n) = n(\log n)^{10}$ be two positive functions of n . Which of the following statements is correct?

- A. $f(n) = O(g(n))$ and $g(n) \neq O(f(n))$
- B. $g(n) = O(f(n))$ and $f(n) \neq O(g(n))$
- C. $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$
- D. $f(n) = O(g(n))$ and $g(n) = O(f(n))$

[gate2001](#) [algorithms](#) [asymptotic-notations](#) [time-complexity](#) [normal](#)

[Answer](#)

1.3.6 Asymptotic Notations: GATE2003-20 [top](#)

<https://gateoverflow.in/910>

Consider the following three claims:

- I. $(n+k)^m = \Theta(n^m)$ where k and m are constants
- II. $2^{n+1} = O(2^n)$
- III. $2^{2n+1} = O(2^n)$

Which of the following claims are correct?

- A. I and II
- B. I and III
- C. II and III
- D. I, II, and III

[gate2003](#) [algorithms](#) [asymptotic-notations](#) [normal](#)

[Answer](#)

1.3.7 Asymptotic Notations: GATE2004-IT-55 [top](#)

<https://gateoverflow.in/3698>

Let $f(n)$, $g(n)$ and $h(n)$ be functions defined for positive integers such that $f(n) = O(g(n))$, $g(n) \neq O(f(n))$, $g(n) = O(h(n))$, and $h(n) = O(g(n))$.

Which one of the following statements is FALSE?

- A. $f(n) + g(n) = O(h(n) + h(n))$
- B. $f(n) = O(h(n))$
- C. $h(n) \neq O(f(n))$
- D. $f(n)h(n) \neq O(g(n)h(n))$

[gate2004-it](#) [algorithms](#) [asymptotic-notations](#) [normal](#)

[Answer](#)

1.3.8 Asymptotic Notations: GATE2005-37 [top](#)

<https://gateoverflow.in/1373>

Suppose $T(n) = 2T(\frac{n}{2}) + n$, $T(0) = T(1) = 1$

Which one of the following is FALSE?

- A. $T(n) = O(n^2)$

- B. $T(n) = \Theta(n \log n)$
 C. $T(n) = \Omega(n^2)$
 D. $T(n) = O(n \log n)$

gate2005 algorithms asymptotic-notations recurrence normal

Answer

1.3.9 Asymptotic Notations: GATE2008-39 top

<https://gateoverflow.in/450>

Consider the following functions:

- $f(n) = 2^n$
- $g(n) = n!$
- $h(n) = n^{\log n}$

Which of the following statements about the asymptotic behavior of $f(n)$, $g(n)$ and $h(n)$ is true?

- A. $f(n) = O(g(n))$; $g(n) = O(h(n))$
 B. $f(n) = \Omega(g(n))$; $g(n) = O(h(n))$
 C. $g(n) = O(f(n))$; $h(n) = O(f(n))$
 D. $h(n) = O(f(n))$; $g(n) = \Omega(f(n))$

gate2008 algorithms asymptotic-notations normal

Answer

1.3.10 Asymptotic Notations: GATE2008-IT-10 top

<https://gateoverflow.in/3270>

Arrange the following functions in increasing asymptotic order:

- A. $n^{1/3}$
 B. e^n
 C. $n^{7/4}$
 D. $n \log^9 n$
 E. 1.0000001^n

- A. a, d, c, e, b
 B. d, a, c, e, b
 C. a, c, d, e, b
 D. a, c, d, b, e

gate2008-it algorithms asymptotic-notations normal

Answer

1.3.11 Asymptotic Notations: GATE2011-37 top

<https://gateoverflow.in/2139>

Which of the given options provides the increasing order of asymptotic complexity of functions f_1, f_2, f_3 and f_4 ?

- $f_1(n) = 2^n$
- $f_2(n) = n^{3/2}$
- $f_3(n) = n \log_2 n$
- $f_4(n) = n^{\log_2 n}$

- A. f_3, f_2, f_4, f_1
- B. f_3, f_2, f_1, f_4
- C. f_2, f_3, f_1, f_4
- D. f_2, f_3, f_4, f_1

gate2011 algorithms asymptotic-notations normal

[Answer](#)

1.3.12 Asymptotic Notations: GATE2012-18 [top](#)

<https://gateoverflow.in/50>

Let $W(n)$ and $A(n)$ denote respectively, the worst case and average case running time of an algorithm executed on an input of size n . Which of the following is **ALWAYS TRUE**?

- A. $A(n) = \Omega(W(n))$
- B. $A(n) = \Theta(W(n))$
- C. $A(n) = O(W(n))$
- D. $A(n) = o(W(n))$

gate2012 algorithms easy asymptotic-notations

[Answer](#)

1.3.13 Asymptotic Notations: GATE2015-3-4 [top](#)

<https://gateoverflow.in/8398>

Consider the equality $\sum_{i=0}^n i^3 = X$ and the following choices for X :

- I. $\Theta(n^4)$
- II. $\Theta(n^5)$
- III. $O(n^5)$
- IV. $\Omega(n^3)$

The equality above remains correct if X is replaced by

- A. Only I
- B. Only II
- C. I or III or IV but not II
- D. II or III or IV but not I

gate2015-3 algorithms asymptotic-notations normal

[Answer](#)

1.3.14 Asymptotic Notations: GATE2015-3-42 [top](#)

<https://gateoverflow.in/8501>

Let $f(n) = n$ and $g(n) = n^{(1+\sin n)}$, where n is a positive integer. Which of the following statements is/are correct?

- I. $f(n) = O(g(n))$
- II. $f(n) = \Omega(g(n))$

- A. Only I
- B. Only II
- C. Both I and II
- D. Neither I nor II

[gate2015-3](#) [algorithms](#) [asymptotic-notations](#) [normal](#)
Answer

1.3.15 Asymptotic Notations: GATE2017-1-04 [top](#)

<https://gateoverflow.in/118703>

Consider the following functions from positive integers to real numbers:

$$10, \sqrt{n}, n, \log_2 n, \frac{100}{n}.$$

The CORRECT arrangement of the above functions in increasing order of asymptotic complexity is:

- A. $\log_2 n, \frac{100}{n}, 10, \sqrt{n}, n$
- B. $\frac{100}{n}, 10, \log_2 n, \sqrt{n}, n$
- C. $10, \frac{100}{n}, \sqrt{n}, \log_2 n, n$
- D. $\frac{100}{n}, \log_2 n, 10, \sqrt{n}, n$

[gate2017-1](#) [algorithms](#) [asymptotic-notations](#) [normal](#)
Answer

1.3.16 Asymptotic Notations: TIFR2011-B-27 [top](#)

<https://gateoverflow.in/20573>

Let n be a large integer. Which of the following statements is **TRUE**?

- A. $n^{1/\sqrt{\log_2 n}} < \sqrt{\log_2 n} < n^{1/100}$
- B. $n^{1/100} < n^{1/\sqrt{\log_2 n}} < \sqrt{\log_2 n}$
- C. $n^{1/\sqrt{\log_2 n}} < n^{1/100} < \sqrt{\log_2 n}$
- D. $\sqrt{\log_2 n} < n^{1/\sqrt{\log_2 n}} < n^{1/100}$
- E. $\sqrt{\log_2 n} < n^{1/100} < n^{1/\sqrt{\log_2 n}}$

[tifr2011](#) [asymptotic-notations](#)
Answer

1.3.17 Asymptotic Notations: TIFR2012-B-6 [top](#)

<https://gateoverflow.in/25106>

Let n be a large integer. Which of the following statements is **TRUE**?

- A. $2^{\sqrt{2 \log n}} < \frac{n}{\log n} < n^{1/3}$
- B. $\frac{n}{\log n} < n^{1/3} < 2^{\sqrt{2 \log n}}$
- C. $2^{\sqrt{2 \log n}} < n^{1/3} < \frac{n}{\log n}$
- D. $n^{1/3} < 2^{\sqrt{2 \log n}} < \frac{n}{\log n}$
- E. $\frac{n}{\log n} < 2^{\sqrt{2 \log n}} < n^{1/3}$

[tifr2012](#) [algorithms](#) [asymptotic-notations](#)
Answer

1.3.18 Asymptotic Notations: TIFR2014-B-8 [top](#)

<https://gateoverflow.in/27192>

Which of these functions grows fastest with n ?

- A. e^n/n .
- B. $e^{n-0.9} \log n$.
- C. 2^n .
- D. $(\log n)^{n-1}$.
- E. None of the above.

[tifr2014](#) [algorithms](#) [asymptotic-notations](#)

[Answer](#)

1.3.19 Asymptotic Notations: TIFR2016-B-7 [top](#)

<https://gateoverflow.in/30720>

Let $n = m!$. Which of the following is **TRUE**?

- A. $m = \Theta(\log n / \log \log n)$
- B. $m = \Omega(\log n / \log \log n)$ but not $m = O(\log n / \log \log n)$
- C. $m = \Theta(\log^2 n)$
- D. $m = \Omega(\log^2 n)$ but not $m = O((\log^2 n))$
- E. $m = \Theta(\log^{1.5} n)$

[tifr2016](#) [asymptotic-notations](#)

[Answer](#)

1.3.20 Asymptotic Notations: TIFR2017-A-4 [top](#)

<https://gateoverflow.in/94943>

Which of the following functions asymptotically grows the fastest as n goes to infinity?

- A. $(\log \log n)!$
- B. $(\log \log n)^{\log n}$
- C. $(\log \log n)^{\log \log \log n}$
- D. $(\log n)^{\log \log n}$
- E. $2^{\sqrt{\log \log n}}$

[tifr2017](#) [algorithms](#) [asymptotic-notations](#)

[Answer](#)

1.3.21 Asymptotic Notations: TIFR2018-A-3 [top](#)

<https://gateoverflow.in/179272>

Which of the following statements is TRUE for all sufficiently large integers n ?

- A. $2^{2^{\sqrt{\log \log n}}} < 2^{\sqrt{\log n}} < n$
- B. $2^{\sqrt{\log n}} < n < 2^{2^{\sqrt{\log \log n}}}$
- C. $n < 2^{\sqrt{\log n}} < 2^{2^{\sqrt{\log \log n}}}$
- D. $n < 2^{2^{\sqrt{\log \log n}}} < 2^{\sqrt{\log n}}$
- E. $2^{\sqrt{\log n}} < 2^{2^{\sqrt{\log \log n}}} < n$

[tifr2018](#) [asymptotic-notations](#)

Answer**1.3.22 Asymptotic Notations: TIFR2018-B-5** top<https://gateoverflow.in/179289>

Which of the following functions, given by there recurrence, grows the fastest asymptotically ?

- A. $T(n) = 4T\left(\frac{n}{2}\right) + 10n$
- B. $T(n) = 8T\left(\frac{n}{3}\right) + 24n^2$
- C. $T(n) = 16T\left(\frac{n}{4}\right) + 10n^2$
- D. $T(n) = 25T\left(\frac{n}{5}\right) + 20(n \log n)^{1.99}$
- E. They all are asymptotically the same

[tifr2018](#) [asymptotic-notations](#) [recurrence](#)**Answer****Answers: Asymptotic Notations****1.3.1 Asymptotic Notations: GATE1994-1.23** top<https://gateoverflow.in/2466>

Selected Answer

For asymptotic complexity, we assume sufficiently large n . So, $g_1(n) = n^2$ and $g_2(n) = n^3$. Growth rate of g_1 is less than that of g_2 , i.e., $g_1(n) = O(g_2(n))$.

Options A and B are TRUE here.

26 votes

-- Arjun Suresh (350k points)

1.3.2 Asymptotic Notations: GATE1996-1.11 top<https://gateoverflow.in/2715>

Selected Answer

- A. $100n \log n = O\left(\frac{n \log n}{100}\right)$: Big-O denotes the growth rate of functions and multiplication or division by a constant does not change the growth rate. So, this is TRUE and here O can even be replaced by Θ or Ω .
- B. $\sqrt{\log n} = O(\log \log n)$: FALSE. Take any long value like 256. LHS results in 16 but RHS results in 4 only. Generally we take log on the left side but that is wrong.
- C. $0 < x < y$ then $n^x = O(n^y)$: TRUE since y is always greater than x . So, RHS is always greater than LHS.
- D. $2^n \neq O(nk)$: TRUE since k is constant. So, for large values of n , LHS is much higher than RHS (exponential function always greater than linear).

Only
B is FALSE.

22 votes

-- Prashant Singh (59.8k points)

1.3.3 Asymptotic Notations: GATE1999-2.21 top<https://gateoverflow.in/1498>

Selected Answer

(M) $T(n) = \text{Sum of first } n \text{ natural numbers} = \frac{n(n+1)}{2} = O(n^2)$

(N) $T(n) = \Theta(n) = O(n)$, third case of Master theorem

$(f(n) = n = \Omega(n^{\log_b a + \epsilon}) = \Omega(n^{\log_2 1 + \epsilon}) = \Omega(n^{0+\epsilon}))$, satisfied for any positive $\epsilon \leq 1$. Also, $af\left(\frac{n}{b}\right) < cf(n) \implies f\left(\frac{n}{2}\right) < cf(n) \implies \frac{n}{2} < cn$, satisfied for any c between 0 and 0.5)

(O) $T(n) = \Theta(n \log n) = O(n \log n)$, third case of Master theorem

$(f(n) = n \log n = \Omega(n^{\log_b a + \epsilon}) = \Omega(n^{\log_2 1 + \epsilon}) = \Omega(n^{0.5+\epsilon}))$, satisfied for positive $\epsilon = 0.5$. Also, $af\left(\frac{n}{b}\right) < cf(n) \implies f\left(\frac{n}{2}\right) < cf(n) \implies \frac{n}{2} \log \frac{n}{2} < cn \log n$, satisfied for $c = 0.5$)

(P) Like in (M), here we are adding the log of the first n natural numbers. So,

$$T_n = \log 1 + \log 2 + \log 3 + \dots + \log n$$

$$= \log(1 \times 2 \times \dots \times n)$$

$$= \log(n!)$$

$= \Theta(n \log n)$ (Stirling's Approximation)

✍ 28 votes

-- Arjun Suresh (350k points)

1.3.4 Asymptotic Notations: GATE2000-2.17 top

<https://gateoverflow.in/664>



Selected Answer

	$n = 256$	$n = 65536$
$f(n) = 3n^{\sqrt{n}}$	3×256^{16} $= 3 \times 2^{128}$	3×65536^{256} $= 3 \times 2^{16 \times 256}$ $= 3 \times 2^{4096}$
$g(n) = 2^{\sqrt{n} \log_2 n}$	$2^{16 \times 8}$ $= 2^{128}$	$2^{256 \times 16}$ $= 2^{4096}$
$h(n) = n!$	256! $= O((2^8)^{256})$ $= O(2^{2048})$	65536! $= O((2^{16})^{65536})$ $= O(2^{1M})$

Case of $h(n)$ is given only by an upper bound but factorial has higher growth rate than exponential.

<http://math.stackexchange.com/questions/351815/do-factorials-really-grow-faster-than-exponential-functions>

$f(n)$ and $g(n)$ are having same order of growth as $f(n)$ is simply $3 \times g(n)$ (we can prove this by

taking log also). So, (d) is correct and all other choices are false.

36 votes

-- Arjun Suresh (350k points)

1.3.5 Asymptotic Notations: GATE2001-1.16 top

<https://gateoverflow.in/709>



A more formal approach:

$$f(n) = n^2 \log n$$

$$g(n) = n(\log n)^{10}$$

We can use the limit definition of O -notation

<http://cse.unl.edu/~choueiry/S06-235/files/Asymptotics-HandoutNoNotes.pdf>

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0, \implies f(n) = o(g(n))$$

small o implying f is strictly asymptotically lower than g . Also by definition, $o \implies O$ but $O \not\implies o$.

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c, c > 0 \implies f(n) = \Theta(g(n))$$

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty, \implies f(n) = \omega(g(n))$$

small ω implying f is strictly asymptotically higher than g . Also by definition, $\omega \implies \Omega$, but $\Omega \not\implies \omega$.

We can use this to prove the above question

For any $k > 0$

$$\lim_{n \rightarrow \infty} \frac{(\log n)^k}{n}$$

Applying L'Hôpital's rule,

$$= \lim_{n \rightarrow \infty} \frac{k * (\log n)^{k-1}}{n}$$

$$= \lim_{n \rightarrow \infty} \frac{k!}{n} = 0$$

So, $(\log n)^k = o(n)$

Now for large n , $n > (\log n)^9$

i.e $n^2 \log n > n(\log n)^{10}$

So $n(\log n)^{10} = o(n^2 \log n)$

Or

$$n(\log n)^{10} = O(n^2 \log n)$$

and

$$O(n^2 \log n) \neq n(\log n)^{10}$$

Option B.

Up 14 votes

-- janakyMurthy (1k points)

	$f(n)$	$g(n)$
$n = 2^{10}$	$10 * 2^{10} * 2^{10}$	$2^{10} * 10^{10}$
$n = 2^{256}$	$256 * 2^{256} * 2^{256}$	$2^{256} * 256^{10}$

So, as n is going larger, $f(n)$ is overtaking $g(n)$ and the growth rate of f is faster than that of g . So, $g(n) = O(f(n))$ and $f(n) \neq O(g(n))$.

B choice.

Up 39 votes

-- Arjun Suresh (350k points)

1.3.6 Asymptotic Notations: GATE2003-20 top

<https://gateoverflow.in/910>



Selected Answer

I. Rate of growth of $(n + k)^m$ is same as that of (n^m) as k and m are constants. (If either k or m is a variable then the equality does not hold), i.e., for sufficiently large values of n ,

$$(n + k)^m \leq an^m \text{ and}$$

$$n^m \leq b(n + k)^m$$

where a and b are positive constants. Here, a can be k^m and b can be 1.

So, **TRUE**.

II. $2^{n+1} = 2 \times (2^n) = \Theta(2^n)$ as 2 is a constant here.

As 2^{n+1} is both upper and lower bounded by 2^n we can say $2^{n+1} = O(2^n)$. (Θ implies both O as well as Ω)

So, **TRUE**.

III. 2^{2n+1} has same rate of growth as 2^{2n} .

$$2^{2n} = 2^{n2} = 2^n \times 2^n$$

2^n is upper bounded by $(2^n)^2$, not the other way round as 2^{2n} is increasing by a factor of 2^n which is not a constant.

So, **FALSE**.

Up 27 votes

-- Danish (3.8k points)

1.3.7 Asymptotic Notations: GATE2004-IT-55 top

<https://gateoverflow.in/3698>

Answer is
(D).

We can verify as : $f \leq g$ but $g \not\leq f$. Therefore,
 $f < g$.

Also,
 $g = h$, as $g = O(h)$ and $h = O(g)$.

👍 16 votes

-- Sandeep_Uniyal (7.5k points)

1.3.8 Asymptotic Notations: GATE2005-37 top

<https://gateoverflow.in/1373>



Selected Answer

Applying Masters theorem,

$$T(n) = \Theta(n \log n)$$

So, it cannot be $\Omega(n^2)$.

Hence, answer is
(C).

👍 25 votes

-- shreya ghosh (3.5k points)

1.3.9 Asymptotic Notations: GATE2008-39 top

<https://gateoverflow.in/450>



Selected Answer

$$g(n) = n!$$

On expanding the factorial we get $g(n) = O(n^n)$:

$$\begin{aligned} n^n &> n^{\log n} \\ n^n &> 2^n \end{aligned}$$

This condition is violated by options **A**, **B** and **C** by first statements of each. Hence, they cannot be said to be TRUE.

Second statement of option **D** says that $g(n)$ is asymptotically biggest of all.

Answer is option
(D).

👍 19 votes

-- Amar Vashishth (30.5k points)

1.3.10 Asymptotic Notations: GATE2008-IT-10 top

<https://gateoverflow.in/3270>



A < C and A < D

E < B

and

C, D < E as E is exponential function.

Now, we just need to see if *C* or *D* is larger.

In *C* we have a term $n^{3/4}$ and correspondingly in *D* we have $\log^9 n$ (after taking n out).

$n^{3/4}$ is asymptotically larger than $\log^9 n$ as when $n = 10^{100}$, $\log^9 n$ gives 100^9 , while $n^{3/4}$ gives $10^{75} > 100^{37}$ a much higher value and this is true for all higher values of n . So, *D < C*.

Thus, *A* is correct.

Upvote 28 votes

-- Arjun Suresh (350k points)

1.3.11 Asymptotic Notations: GATE2011-37 [top](#)

<https://gateoverflow.in/2139>



Answer is (*A*).

$n \log_2 n < n^{3/2}$ is quite straightforward as $n^{3/2} = n \times n^{1/2}$ and $\log n < n^{1/2}$ as logarithmic growth is smaller than exponential growth however small be the exponentiation factor.

Also, $n^{3/2} < n^{\log_2 n}$ and $n^{3/2} < 2^n$.

Now only $n^{\log_2 n}$ and 2^n need to be compared.

Taking log of both $(\log_2 n)^2$ and n ,

$$n > (\log_2 n)^2$$

Hence, $2^n > n^{\log_2 n}$.

NOTE: We cannot compare two functions for asymptotic growth by taking log if they are giving constants after log operation.

Upvote 28 votes

-- Ankit Rokde (9k points)

1.3.12 Asymptotic Notations: GATE2012-18 [top](#)

<https://gateoverflow.in/50>



Selected Answer

Worst case complexity can never be lower than the average case complexity, but it can be higher.
So, (*C*) is the answer.
 $A(n) = O(W(n))$.

Upvote 31 votes

-- Arjun Suresh (350k points)

1.3.13 Asymptotic Notations: GATE2015-3-4 top

<https://gateoverflow.in/8398>



Selected Answer

Sum of the cubes of the first n natural numbers is given by $(n(n+1)/2)^2$ which is $\Theta(n^4)$. So, I, III and IV are correct. II is wrong.

$\therefore (C)$ is correct.

1 38 votes

-- Arjun Suresh (350k points)

1.3.14 Asymptotic Notations: GATE2015-3-42 top

<https://gateoverflow.in/8501>



Selected Answer

The answer is option (D).

Since the value of $\sin(n)$ will always range from -1 to $+1$, hence $g(n)$ can take values $1, n, n^2$.

Hence, if $g(n) = 1$, then statement I is incorrect.

And, if $g(n) = n^2$, then statement II is incorrect.

1 39 votes

-- saurabhrk (1.3k points)

1.3.15 Asymptotic Notations: GATE2017-1-04 top

<https://gateoverflow.in/118703>



Selected Answer

10 is constant. \therefore Growth rate is 0.

\sqrt{n} grows slower than linear but faster than log. (Consider $\frac{\sqrt{n^2}}{\sqrt{n}} = \sqrt{n}$, whereas $\frac{\log(n^2)}{\log n} = 2$)

n : Growth rate is linear.

$\log_2 n$: Growth rate is logarithmic. For asymptotic growth, the base does not matter.

$\frac{100}{n}$: Growth rate decreases with n .

So, correct answer is (B).

NOTE: Please never substitute large values of n in such questions. If ever you do, at least do for 2 such values and take ratio to get the growth rate or plot a graph. Remember $1.01^n \neq O(n^{100})$.

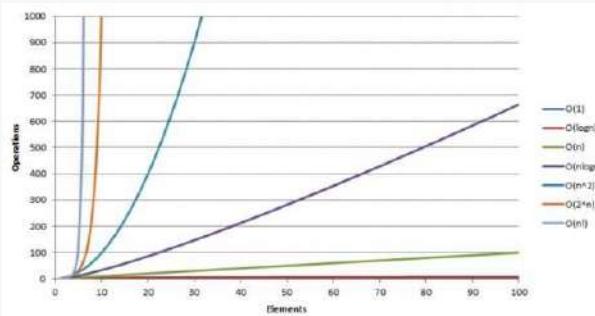
1 24 votes

-- Arjun Suresh (350k points)

- 10, Constant
- \sqrt{n} , Square root

- n , polynomial
- $\log_2 n$, Logarithmic
- $\frac{100}{n}$. Constant division by polynomial (**clearly less than constant for every value of n > 100**)

Now we know Constant division by polynomial < Constant < Logarithmic < Square root < polynomial



So correct order is $\frac{100}{n}, 10, \log_2 n, \sqrt{n}, n$

21 votes

-- Prashant Singh (59.8k points)

1.3.16 Asymptotic Notations: TIFR2011-B-27 top

<https://gateoverflow.in/20573>



Selected Answer

Let $n = 2^x$. Then, $\log_2 n = x$.

$$f(n) = n^{1/\sqrt{\log_2 n}} = (2^x)^{1/\sqrt{x}} = 2^{x/\sqrt{x}} = 2^{\sqrt{x}}$$

$$g(n) = \sqrt{\log_2 n} = \sqrt{\log_2 (2^x)} = \sqrt{x}$$

$$h(n) = n^{1/100} = (2^x)^{1/100} = 2^{x/100}$$

Since exponentials grow faster than polynomials, $h(n) > g(n)$ for large n .

Since linear functions grow faster than square roots, $\frac{x}{100} > \sqrt{x}$ for large x . Thus, $h(n) > f(n)$ for large n .

Since exponentials grow faster than polynomials, $2^{\sqrt{x}} > \sqrt{x}$ for large \sqrt{x} . Thus, $f(n) > g(n)$ for large n .

Hence, the relation is,

$$g(n) < f(n) < h(n)$$

Thus, option

(D) is correct.

23 votes

-- Pragy Agarwal (20.6k points)

1.3.17 Asymptotic Notations: TIFR2012-B-6 top

<https://gateoverflow.in/25106>



Selected Answer
Answer will be (C).

Take $n = 2^{1024}$

Now, $2^{\sqrt{2 \log n}} \approx 2^{45}$

$$n^{\frac{1}{3}} \approx 2^{341}$$

$$n/\log n = 2^{1024}/1024 \approx 2^{1014}$$

Just one value is not enough to confirm growth rate. So, take $n = 1024$.

Now, $2^{\sqrt{2 \log n}} \approx 2^4$

$$n^{\frac{1}{3}} \approx 2^3$$

$$n/\log n = 2^{10}/10 \approx 2^7$$

So, as n increases, the gap between second and third function increases and also the second function overtakes the first. So, $f1 < f2 < f3$.

13 votes

-- srestha (87k points)

1.3.18 Asymptotic Notations: TIFR2014-B-8 top

<https://gateoverflow.in/27192>



Selected Answer
Assuming that the base of the log in the question is e .

Let us try to rewrite each of these functions in the form $e^{\text{something}}$, to make the comparison easier.

a.	e^n / n	$= \frac{e^n}{e^{\ln n}}$	$= e^{(n - \ln n)}$
b.	$e^{n-0.9 \ln n}$		$= e^{(n - 0.9 \ln n)}$
c.	2^n	$= (e^{\ln 2})^n$	$= e^{(n \ln 2)}$
d.	$(\ln n)^{n-1}$	$= (e^{\ln \ln n})^{n-1}$	$= e^{(n \ln \ln n - \ln \ln n)}$

Now, if we just compare the exponents of all, we can clearly see that $(n \ln \ln n - \ln \ln n)$ grows

faster than the rest. Note that in option (C), the multiplicative $\ln 2$ is a constant, and hence grows slower than the multiplicative $\ln \ln n$ from option (D).

This implies that $e^{(n \ln \ln n - \ln \ln n)}$ grows the fastest, and hence, $(\ln n)^{n-1}$ grows the fastest.

Thus, option (D) is the correct answer.

Upvote 25 votes

-- Pragy Agarwal (20.6k points)

1.3.19 Asymptotic Notations: TIFR2016-B-7 [top](#)

<https://gateoverflow.in/30720>

m	$n = m!$	$\log n / \log \log n$
4	24	$4/2 = 2$
6	720	$9/3 = 3$
8	$720 * 56$	$15/3 = 5$
10	$720 * 56 * 90$	$21/4 = 5$

If we see, m is growing at same rate as $\log n / \log \log n$.

Upvote 9 votes

-- Arjun Suresh (350k points)

1.3.20 Asymptotic Notations: TIFR2017-A-4 [top](#)

<https://gateoverflow.in/94943>



Selected Answer

Let $N = 2^{256}$

- A. $(\log \log N)! = 8!$
- B. $(\log \log N)^{\log N} = (8)^{256} = 2^{768}$
- C. $(\log \log N)^{\log \log \log N} = 8^3 = 512$
- D. $(\log N)^{\log \log N} = (256)^8 = 2^{64}$
- E. $2^{\sqrt{\log \log N}} = 2^{2\sqrt{2}}$

Let $N = 2^{16}$

- A. $(\log \log N)! = 4!$
- B. $(\log \log N)^{\log N} = (4)^{16} = 2^{32}$
- C. $(\log \log N)^{\log \log \log N} = 4^2 = 16$
- D. $(\log N)^{\log \log N} = (16)^4 = 2^{16}$
- E. $2^{\sqrt{\log \log N}} = 2^2 = 4$

Taking ratio for both N values,

- A. $(\log \log N)! \rightarrow 8!/4!$
- B. $(\log \log N)^{\log N} \rightarrow 2^{736}$
- C. $(\log \log N)^{\log \log \log N} \rightarrow 32$
- D. $(\log N)^{\log \log N} \rightarrow 2^{48}$
- E. $2^{\sqrt{\log \log N}} \rightarrow \approx 2$

Option $B = (\log \log N)^{\log N}$ asymptotically grows the fastest, as for the same change in N , the value increased the most (growth) for option B and this growth is monotonic (continuous).

Upvote 23 votes

-- Kapil Phulwani (50k points)

1.3.21 Asymptotic Notations: TIFR2018-A-3 top

<https://gateoverflow.in/179272>

Take $n = 2^{2^k}$

Now, option (A) gives $n = 2^{2^k}$

Option (B) gives $2^{\sqrt{\log 2^{2^k}}} = 2^{\sqrt{2^k}}$

Option (C) gives $2^{2^{\sqrt{\log(\log 2^{2^k})}}} = 2^{2^{\sqrt{k}}}$

Now, check power only for (B) and (C).

Take log of power of both functions, i.e.,

$$(\log(2^{\sqrt{k}})) = O(\log(\sqrt{2^k}))$$

$$\sqrt{k} = O(\frac{k}{2})$$

Clearly, (C) < (B) < (A).

Option (A) is answer.

Upvote 3 votes

-- Anu007 (17k points)

1.3.22 Asymptotic Notations: TIFR2018-B-5 top

<https://gateoverflow.in/179289>



Selected Answer

By applying master theorem,

(a) $T(n) = \Theta(n^2)$

(b) $T(n) = \Theta(n^2)$

(c) $T(n) = \Theta(n^2 * \log n)$

(d) $T(n) = \Theta(n^2)$

C is growing fastest.

Upvote 5 votes

-- Hemant Parihar (14.8k points)

1.4

Dynamic Programming(12) top

1.4.1 Dynamic Programming: GATE2008-80 top

<https://gateoverflow.in/498>

The subset-sum problem is defined as follows. Given a set of n positive integers, $S = \{a_1, a_2, a_3, \dots, a_n\}$, and positive integer W , is there a subset of S whose elements sum to W ? A dynamic program for solving this problem uses a 2-dimensional Boolean array, X , with n rows and $W + 1$ columns. $X[i, j], 1 \leq i \leq n, 0 \leq j \leq W$, is TRUE, if and only if there is a subset of $\{a_1, a_2, \dots, a_i\}$ whose elements sum to j .

Which of the following is valid for $2 \leq i \leq n$, and $a_i \leq j \leq W$?

- A. $X[i, j] = X[i - 1, j] \vee X[i, j - a_i]$
- B. $X[i, j] = X[i - 1, j] \vee X[i - 1, j - a_i]$
- C. $X[i, j] = X[i - 1, j] \wedge X[i, j - a_i]$
- D. $X[i, j] = X[i - 1, j] \wedge X[i - 1, j - a_i]$

gate2008 algorithms normal dynamic-programming

Answer

1.4.2 Dynamic Programming: GATE2008-81 [top](#)

<https://gateoverflow.in/43484>

The subset-sum problem is defined as follows. Given a set of n positive integers, $S = \{a_1, a_2, a_3, \dots, a_n\}$, and positive integer W , is there a subset of S whose elements sum to W ? A dynamic program for solving this problem uses a 2-dimensional Boolean array, X , with n rows and $W + 1$ columns. $X[i, j], 1 \leq i \leq n, 0 \leq j \leq W$, is TRUE, if and only if there is a subset of $\{a_1, a_2, \dots, a_i\}$ whose elements sum to j .

Which entry of the array X , if TRUE, implies that there is a subset whose elements sum to W ?

- A. $X[1, W]$
- B. $X[n, 0]$
- C. $X[n, W]$
- D. $X[n - 1, n]$

gate2008 algorithms normal dynamic-programming

Answer

1.4.3 Dynamic Programming: GATE2009-53 [top](#)

<https://gateoverflow.in/1338>

A sub-sequence of a given sequence is just the given sequence with some elements (possibly none or all) left out. We are given two sequences $X[m]$ and $Y[n]$ of lengths m and n , respectively with indexes of X and Y starting from 0.

We wish to find the length of the longest common sub-sequence (LCS) of $X[m]$ and $Y[n]$ as $l(m, n)$, where an incomplete recursive definition for the function $I(i, j)$ to compute the length of the LCS of $X[m]$ and $Y[n]$ is given below:

```

l(i, j) = 0, if either i = 0 or j = 0
        = expr1, if i, j > 0 and X[i-1] = Y[j-1]
        = expr2, if i, j > 0 and X[i-1] ≠ Y[j-1]
    
```

Which one of the following options is correct?

- A. $expr1 = l(i - 1, j) + 1$
- B. $expr1 = l(i, j - 1)$
- C. $expr2 = \max(l(i - 1, j), l(i, j - 1))$

- D. $\text{expr2} = \max(l(i-1, j-1), l(i, j))$

gate2009 algorithms normal dynamic-programming recursion

Answer

1.4.4 Dynamic Programming: GATE2009-54 [top](#)

<https://gateoverflow.in/43476>

A sub-sequence of a given sequence is just the given sequence with some elements (possibly none or all) left out. We are given two sequences $X[m]$ and $Y[n]$ of lengths m and n , respectively with indexes of X and Y starting from 0.

We wish to find the length of the longest common sub-sequence (LCS) of $X[m]$ and $Y[n]$ as $l(m, n)$, where an incomplete recursive definition for the function $I(i, j)$ to compute the length of the LCS of $X[m]$ and $Y[n]$ is given below:

$$\begin{aligned} l(i, j) &= 0, \text{ if either } i = 0 \text{ or } j = 0 \\ &= \text{expr1, if } i, j > 0 \text{ and } X[i-1] = Y[j-1] \\ &= \text{expr2, if } i, j > 0 \text{ and } X[i-1] \neq Y[j-1] \end{aligned}$$

The value of $l(i, j)$ could be obtained by dynamic programming based on the correct recursive definition of $l(i, j)$ of the form given above, using an array $L[M, N]$, where $M = m + 1$ and $N = n + 1$, such that $L[i, j] = l(i, j)$.

Which one of the following statements would be TRUE regarding the dynamic programming solution for the recursive definition of $l(i, j)$?

- A. All elements of L should be initialized to 0 for the values of $l(i, j)$ to be properly computed.
- B. The values of $l(i, j)$ may be computed in a row major order or column major order of $L[M, N]$.
- C. The values of $l(i, j)$ cannot be computed in either row major order or column major order of $L[M, N]$.
- D. $L[p, q]$ needs to be computed before $L[r, s]$ if either $p < r$ or $q < s$.

gate2009 normal algorithms dynamic-programming recursion

Answer

1.4.5 Dynamic Programming: GATE2010-34 [top](#)

<https://gateoverflow.in/2208>

The weight of a sequence a_0, a_1, \dots, a_{n-1} of real numbers is defined as $a_0 + a_1/2 + \dots + a_{n-1}/2^{n-1}$. A subsequence of a sequence is obtained by deleting some elements from the sequence, keeping the order of the remaining elements the same. Let X denote the maximum possible weight of a subsequence of a_0, a_1, \dots, a_{n-1} and Y the maximum possible weight of a subsequence of a_1, a_2, \dots, a_{n-1} . Then X is equal to

- A. $\max(Y, a_0 + Y)$
- B. $\max(Y, a_0 + Y/2)$
- C. $\max(Y, a_0 + 2Y)$
- D. $a_0 + Y/2$

gate2010 algorithms dynamic-programming normal

Answer

1.4.6 Dynamic Programming: GATE2011-25 [top](#)

<https://gateoverflow.in/2127>

An algorithm to find the length of the longest monotonically increasing sequence of numbers in an

array $A[0 : n - 1]$ is given below.

Let L_i , denote the length of the longest monotonically increasing sequence starting at index i in the array.

Initialize $L_{n-1} = 1$.

For all i such that $0 \leq i \leq n - 2$

$$L_i = \begin{cases} 1 + L_{i+1} & \text{if } A[i] < A[i+1] \\ 1 & \text{Otherwise} \end{cases}$$

Finally, the length of the longest monotonically increasing sequence is $\max(L_0, L_1, \dots, L_{n-1})$.

Which of the following statements is **TRUE**?

- A. The algorithm uses dynamic programming paradigm
- B. The algorithm has a linear complexity and uses branch and bound paradigm
- C. The algorithm has a non-linear polynomial complexity and uses branch and bound paradigm
- D. The algorithm uses divide and conquer paradigm

[gate2011](#) [algorithms](#) [easy](#) [dynamic-programming](#)

Answer

1.4.7 Dynamic Programming: GATE2011-38 [top](#)

<https://gateoverflow.in/2140>

Four Matrices M_1, M_2, M_3 and M_4 of dimensions $p \times q$, $q \times r$, $r \times s$ and $s \times t$ respectively can be multiplied in several ways with different number of total scalar multiplications. For example when multiplied as $((M_1 \times M_2) \times (M_3 \times M_4))$, the total number of scalar multiplications is $pqr + rst + prt$. When multiplied as $((M_1 \times M_2) \times M_3) \times M_4$, the total number of scalar multiplications is $pqr + prs + pst$.

If $p = 10, q = 100, r = 20, s = 5$ and $t = 80$, then the minimum number of scalar multiplications needed is

- A. 248000
- B. 44000
- C. 19000
- D. 25000

[gate2011](#) [algorithms](#) [dynamic-programming](#) [normal](#)

Answer

1.4.8 Dynamic Programming: GATE2014-2-37 [top](#)

<https://gateoverflow.in/1996>

Consider two strings $A = "qpqrr"$ and $B = "pqprqrp"$. Let x be the length of the longest common subsequence (not necessarily contiguous) between A and B and let y be the number of such longest common subsequences between A and B . Then $x + 10y = \underline{\hspace{2cm}}$.

[gate2014-2](#) [algorithms](#) [normal](#) [numerical-answers](#) [dynamic-programming](#)

Answer

1.4.9 Dynamic Programming: GATE2014-3-37 [top](#)

<https://gateoverflow.in/2071>

Suppose you want to move from 0 to 100 on the number line. In each step, you either move right

by a unit distance or you take a *shortcut*. A shortcut is simply a pre-specified pair of integers i, j with $i < j$. Given a shortcut (i, j) , if you are at position i on the number line, you may directly move to j . Suppose $T(k)$ denotes the smallest number of steps needed to move from k to 100. Suppose further that there is at most 1 shortcut involving any number, and in particular, from 9 there is a shortcut to 15. Let y and z be such that $T(9) = 1 + \min(T(y), T(z))$. Then the value of the product yz is _____.

gate2014-3 | algorithms | normal | numerical-answers | dynamic-programming

Answer

1.4.10 Dynamic Programming: GATE2016-2-14 [top](#)

<https://gateoverflow.in/39570>

The Floyd-Warshall algorithm for all-pair shortest paths computation is based on

- A. Greedy paradigm.
- B. Divide-and-conquer paradigm.
- C. Dynamic Programming paradigm.
- D. Neither Greedy nor Divide-and-Conquer nor Dynamic Programming paradigm.

gate2016-2 | algorithms | dynamic-programming | easy

Answer

1.4.11 Dynamic Programming: GATE2016-2-38 [top](#)

<https://gateoverflow.in/39587>

Let A_1, A_2, A_3 and A_4 be four matrices of dimensions $10 \times 5, 5 \times 20, 20 \times 10$ and 10×5 , respectively. The minimum number of scalar multiplications required to find the product $A_1 A_2 A_3 A_4$ using the basic matrix multiplication method is _____.

gate2016-2 | dynamic-programming | algorithms | normal | numerical-answers

Answer

1.4.12 Dynamic Programming: GATE2018-31 [top](#)

<https://gateoverflow.in/204105>

Assume that multiplying a matrix G_1 of dimension $p \times q$ with another matrix G_2 of dimension $q \times r$ requires pqr scalar multiplications. Computing the product of n matrices $G_1 G_2 G_3 \dots G_n$ can be done by parenthesizing in different ways. Define $G_i G_{i+1}$ as an **explicitly computed pair** for a given parenthesization if they are directly multiplied. For example, in the matrix multiplication chain $G_1 G_2 G_3 G_4 G_5 G_6$ using parenthesization $(G_1(G_2 G_3))(G_4(G_5 G_6))$, $G_2 G_3$ and $G_5 G_6$ are only explicitly computed pairs.

Consider a matrix multiplication chain $F_1 F_2 F_3 F_4 F_5$, where matrices F_1, F_2, F_3, F_4 and F_5 are of dimensions $2 \times 25, 25 \times 3, 3 \times 16, 16 \times 1$ and 1×1000 , respectively. In the parenthesization of $F_1 F_2 F_3 F_4 F_5$ that minimizes the total number of scalar multiplications, the explicitly computed pairs is/are

- A. $F_1 F_2$ and $F_3 F_4$ only
- B. $F_2 F_3$ only
- C. $F_3 F_4$ only
- D. $F_2 F_2$ and $F_4 F_5$ only

gate2018 | algorithms | dynamic-programming

Answer

Answers: Dynamic Programming

1.4.1 Dynamic Programming: GATE2008-80 [top](#)

<https://gateoverflow.in/498>



Selected Answer

This is analogous to the dynamic programming solution to 0/1 knapsack problem.

Consider the capacity of the knapsack, i.e., W to be analogous to J (the total sum here).

The solution **exploits the optimal substructure** of the problem.

At each stage we can have 2 options:

Case (1): Either we take an item(in this question either we consider the element A_i) along with the total solution to previous sub-problem(total solution here means the total sum obtained till previous sub-problem)

in which case we choose $A[i - 1][j - a_i]$

$A[i - 1]$ indicates we are considering solution to previous subproblem and

$A[j - a_i]$ means we have considered element a_i and now remaining sum is $J - a_i$ which has to be further considered.

Case (2): Or we do not consider the item(in this case the element a_i) in which case we only consider the solution to previous subproblem, which is,

$A[i - 1][J]$

Since the whole solution to this subset-sum problem is Logical OR(+) of cases 1 and 2, we eliminate options C and D because both are considering the Logical AND of the two parts of the solution.

Now, since here in the given question we are given a boolean array $X[n][W + 1]$

So, an entry $X[i][j]$ is true only if sum j is possible with array elements from 0 to i .

So, for Each element of array, a_i , we consider two possibilites:

(1) EIther we can ignore it and still get our possible sum, which is,

$X[i - 1][j]$

OR

(2) We could include element a_i and then get our required sum, which is,

$X[i - 1][j - a_i]$

And finally, to get $X[i][j]$, we take logical or of the above two cases.

Hence, **answer is option B.**

Reference :

Video:

By using the analogy of the problem and solution between subset-sum problem and 0/1 knapsack problem, the above video clearly explains the how the solution to the problem is structured .

Video:

8 votes

-- Ayush Upadhyaya (9k points)

1.4.2 Dynamic Programming: GATE2008-81 [top](#)

<https://gateoverflow.in/43484>

Selected Answer

ANSWER is C.

If LAST ROW and LAST COLUMN entry is 1, then there exists a subset whose elements sum to W .

11 votes

-- Shivam Bhardwaj (289 points)

1.4.3 Dynamic Programming: GATE2009-53 [top](#)

<https://gateoverflow.in/1338>

Selected Answer

Answer is **C**. When the currently compared elements doesn't match, we have two possibilities for the LCS, one including $X[i]$ but not $Y[j]$ and other including $Y[j]$ but not $X[i]$.

```
/* Returns length of LCS for X[0..m-1], Y[0..n-1] */
int lcs( char *X, char *Y, int m, int n )
{
    if (m == 0 || n == 0)
        return 0;
    if (X[m-1] == Y[n-1])
        return 1 + lcs(X, Y, m-1, n-1);
    else
        return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));
```

20 votes

-- Sona Praneeth Akula (4.2k points)

1.4.4 Dynamic Programming: GATE2009-54 [top](#)

<https://gateoverflow.in/43476>

Selected Answer

$$\text{expr2} = \max(l(i-1, j), l(i, j-1))$$

When the currently compared elements doesn't match, we have two possibilities for the LCS, one including $X[i]$ but not $Y[j]$ and other including $Y[j]$ but not $X[i]$.

```
/* Returns length of LCS for X[0..m-1], Y[0..n-1] */
int lcs( char *X, char *Y, int m, int n )
{
    if (m == 0 || n == 0)
        return 0;
    if (X[m-1] == Y[n-1])
        return 1 + lcs(X, Y, m-1, n-1);
    else
        return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));
```

Answer is **B**. Dynamic programming is used to save the previously found LCS. So, for any index $[p,q]$ all smaller ones should have been computed earlier. Option D is not correct as the condition given requires even $L[3,2]$ to be computed before $L[2,4]$ which is not a necessity if we follow row-major order.

```

int lcs( char *X, char *Y, int m, int n )
{
    int L[m+1][n+1];
    int i, j;

    /* Following steps build L[m+1][n+1] in bottom up fashion. Note
       that L[i][j] contains length of LCS of X[0..i-1] and Y[0..j-1] */
    for (i=0; i<=m; i++)
    {
        for (j=0; j<=n; j++)
        {
            if (i == 0 || j == 0)
                L[i][j] = 0;

            else if (X[i-1] == Y[j-1])
                L[i][j] = L[i-1][j-1] + 1;

            else
                L[i][j] = max(L[i-1][j], L[i][j-1]);
        }
    }

    /* L[m][n] contains length of LCS for X[0..n-1] and Y[0..m-1] */
    return L[m][n];
}

```

26 votes

-- Arjun Suresh (350k points)

1.4.5 Dynamic Programming: GATE2010-34 top

<https://gateoverflow.in/2208>



Selected Answer

$$\begin{aligned} S &= \langle a_0, S_1 \rangle \\ S_1 &= \langle a_1, a_2, a_3 \dots a_{n-1} \rangle \end{aligned}$$

Two possible cases arise:

1. **a_0 is included in the max weight subsequence of S :**

In this case, $X = \text{weight}\left(\left(\langle a_0, S_1 \rangle\right)\right) = a_0 + \frac{Y}{2}$

2. **a_0 is not included in the max weight subsequence of S :**

In this case, $X = \text{weight}(S_1) = Y$

Since the value of a_0 can be anything (negative or $< \frac{Y}{2}$ in general) $\{\because a_i \in \mathbb{R}\}$, it is possible that $Y > a_0 + \frac{Y}{2}$.

The maximum possible weight of a subsequence of S is given by:

$$X = \max\left(Y, a_0 + \frac{Y}{2}\right)$$

Thus, option B is correct.

50 votes

-- Pragy Agarwal (20.6k points)

1.4.6 Dynamic Programming: GATE2011-25 top

<https://gateoverflow.in/2127>

Selected Answer

(A) is the answer.

The algorithm is storing the optimal solutions to subproblems at each point (for each i), and then using it to derive the optimal solution of a bigger problem. And that is dynamic programming approach. And the program has linear time complexity.

<http://stackoverflow.com/questions/1065433/what-is-dynamic-programming>

Now, branch and bound comes when we explore all possible solutions (branch) and we backtrack as soon as we realise we won't get a solution (in classical backtracking we will retreat only when we won't find the solution). In backtracking : In each step, you check if this step satisfies all the conditions.

If it does : you continue generating subsequent solutions

If not : you go one step backward to check for another path

So, backtracking gives all possible solutions while branch and bound will give only the optimal one. <http://www.cs.cornell.edu/~wdtseng/icpc/notes/bt2.pdf>

The given algorithm here is neither backtracking nor branch and bound. Because we are not branching anywhere in the solution space.

And the algorithm is not divide and conquer as we are not dividing the problem and then merging the solution as in the case of merge sort (where merge is the conquer step).

https://en.wikipedia.org/wiki/Divide_and_conquer_algorithms

38 votes

-- Arjun Suresh (350k points)

1.4.7 Dynamic Programming: GATE2011-38 top

<https://gateoverflow.in/2140>

Selected Answer

Answer is C.

Ordering: First Multiply $M_2 \times M_3$. This requires $100*20*5$ multiplications.
 Then Multiply $M_1 \times (M_2 \times M_3)$. This requires $10*100*5$ multiplications.
 Then Multiply $(M_1 \times (M_2 \times M_3)) \times M_4$. This requires $10*5*8$ multiplications.
 Total 19000 Multiplications.

Brute Force approach - anyone can do.

No. of possible ordering for 4 matrices is C_3 where C_3 is the 3rd [Catalan number](#) and given by $n = 3$ in $\frac{1}{n+1}^{2n} C_n = 5$.

So, here we have

1. $(M_1 \times M_2) \times (M_3 \times M_4)$
2. $(M_1 \times (M_2 \times M_3)) \times M_4$
3. $((M_1 \times M_2) \times M_3) \times M_4$

4. $M_1 \times (M_2 \times (M_3 \times M_4))$
5. $M_1 \times ((M_2 \times M_3) \times M_4))$

Each of these would give no. of multiplications required as

1. $pqr + rst + prt$
2. $qrs + pqs + pst$
3. $pqr + prs + pst$
4. $rst + qrt + pqt$
5. $qrs + qst + pst$

The last 2 are having qt terms which are the highest terms by far and hence we can avoid them from consideration $qt = 8000$ multiplied by one other term would be larger than any value in choice. So, just find the value of first 3 terms.

1. $pqr + rst + prt = 20000 + 8000 + 16000 = 44000$
2. $qrs + pqs + pst = 10000 + 5000 + 4000 = 19000$ - smallest value in choice, we can stop here.
3. $pqr + prs + pst$

Dynamic Programming Solution (should know [Matrix Chain Ordering algorithm](#))

Here we have a chain of length 4.

Dynamic programming solution of Matrix chain ordering has the solution

$$m[i, j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \leq k < j} m[i, k] + m[k+1, j] + p_{i-1} p_j p_k & \text{if } i < j \end{cases}$$

So, we can fill the following table starting with the diagonals and moving upward diagonally. Here $k < j$ but $\geq i$.

	j=1	j=2	j=3	
i=1	0	$p_0 p_1 p_2 = 20000$	$\min(10000 + p_0 p_1 p_3, 20000 + p_0 + p_2 p_3) = 15000$	$\min(18000 + p_0 p_1 p_2, 20000 + p_0 + p_2 p_3) = 19000$
i=2	0	$p_1 p_2 p_3 = 10000$		$\min(10000 + p_2 p_3 p_4, 15000 + p_1 p_2 p_3) = 15000$
i=3		0		$p_2 p_3 p_4 = 8000$
i=4				0

Our required answer is given by $m[1, 4] = 19000$.

拇指图标 25 votes

-- Sona Praneeth Akula (4.2k points)

1.4.8 Dynamic Programming: GATE2014-2-37 top

<https://gateoverflow.in/1996>



Selected Answer

Answer is 34.

In first string, if we want to get 4 as maximum length then LCS should end with either "rr" or "qr". Only 4 combinations are possible for LCS with length 4:

"qpqr"

"qqrr"

"pqrr"

"qprr"

Now, check for matching sequences in second string, except for "qgrr" all are possible.

16 votes

-- Anurag Semwal (8k points)

1.4.9 Dynamic Programming: GATE2014-3-37 top

<https://gateoverflow.in/2071>



Selected Answer

$T(k)$ is the smallest number of steps needed to move from k to 100.

Now, it is given that y and z are two numbers such that,

$$T(9) = 1 + \min(T(y), T(z)), \text{ i.e.,}$$

$T(9) = 1 + \min(\text{Steps from } y \text{ to 100}, \text{Steps from } z \text{ to 100})$, where y and z are two possible values that can be reached from 9.

One number that can be reached from 9 is 10, which is the number obtained if we simply move one position right on the number line. Another number is 15, the shortcut path from 9, as given in the question. So, we have two paths from 9, one is 10 and the other is 15.

Therefore, the value of y and z is 10 and 15 (either variable may take either of the values).

Thus, $yz = 150$.

17 votes

-- Divya Bharti (8.1k points)

$$T(9) = \text{Distance from 9 to 100}$$

$$T(9) = 1 + \min(T(y), T(z)) = 1 + \min(\text{Distance from } y \text{ to 100}, \text{Distance from } z \text{ to 100})$$

There are only two such values where we can reach from 9, one is simple step to right on number line, i.e 10 and another is 15 (given shortcut)

Hence, $y = 10, z = 15$

$$yz = 10 \times 15 = 150$$

31 votes

-- Srinath Jayachandran (3.7k points)

1.4.10 Dynamic Programming: GATE2016-2-14 top

<https://gateoverflow.in/39570>



Selected Answer

In Floyd Warshall's, we calculate all possibilities and select best one so its neither Divide & Conquer nor Greedy but based on Dynamic Programming Paradigm.

25 votes

-- Anurag Semwal (8k points)

1.4.11 Dynamic Programming: GATE2016-2-38 top

<https://gateoverflow.in/39587>



Selected Answer

Answer is 1500.

atrix Parenthesizing : $A_1((A_2A_3)A_4)$

Check my solution below, using dynamic programming

$$\begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \\ 10 \times 5 & 5 \times 20 & 20 & 10 \times 5 \\ & & \times 10 & \end{array}$$

$$A_{12} = 10 \times 5 \times 20 = 1000$$

$$A_{23} = 5 \times 20 \times 10 = 1000$$

$$A_{34} = 20 \times 10 \times 5 = 1000$$

$$A_{13} = \min \begin{cases} A_{12} + A_{33} + 5 \times 20 \times 10 = 2000 \\ A_{11} + A_{23} + 10 \times 5 \times 10 = 1500 \end{cases}$$

$$A_{24} = \min \begin{cases} A_{23} + A_{44} + 5 \times 10 \times 5 = 1250 \\ A_{22} + A_{34} + 5 \times 20 \times 5 = 1500 \end{cases}$$

$$A_{14} = \min \begin{cases} A_{11} + A_{24} + 10 \times 5 \times 5 = 1500 \\ A_{12} + A_{34} + 10 \times 20 \times 5 \geq 2000 \\ A_{13} + A_{44} + 10 \times 20 \times 5 = 2000 \end{cases}$$

Answer is 1500.

31 votes

-- Akash Kanase (42.5k points)

1.4.12 Dynamic Programming: GATE2018-31 [top](#)

<https://gateoverflow.in/204105>



If we multiply anything with F_5 we will get much greater multiplication cost because F_5 is 1×1000 matrix so 1000 will play vital role in cost. So we will multiply F_5 at very last step.

So, here is the sequence giving minimal cost:

$$(((F_1(F_2(F_3F_4))(F_5)) = 48 + 75 + 50 + 2000 = 2173$$

Explicitly computed pairs is (F_3F_4)

12 votes

-- Digvijay (54.9k points)

1.5

Graph Algorithms(48) [top](#)

1.5.1 Graph Algorithms: GATE1994-1.22 [top](#)

<https://gateoverflow.in/2465>

Which of the following statements is false?

- A. Optimal binary search tree construction can be performed efficiently using dynamic programming
- B. Breadth-first search cannot be used to find connected components of a graph

- C. Given the prefix and postfix walks over a binary tree, the binary tree cannot be uniquely constructed.
- D. Depth-first search can be used to find connected components of a graph

gate1994 algorithms normal graph-algorithms

Answer

1.5.2 Graph Algorithms: GATE1994-24 top

<https://gateoverflow.in/2520>

An independent set in a graph is a subset of vertices such that no two vertices in the subset are connected by an edge. An incomplete scheme for a greedy algorithm to find a maximum independent set in a tree is given below:

```
V: Set of all vertices in the tree;
I :=  $\emptyset$ 
while V  $\neq \emptyset$  do
begin
    select a vertex u  $\in$  V such that
    _____;
    V := V - {u};
    if u is such that
        _____ then I := I  $\cup$  {u}
end;
Output(I);
```

- Complete the algorithm by specifying the property of vertex u in each case.
- What is the time complexity of the algorithm?

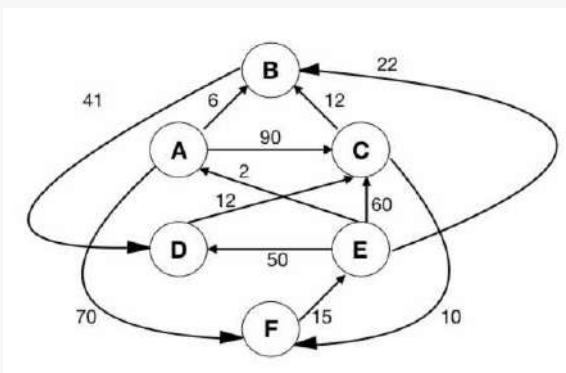
gate1994 algorithms graph-algorithms normal

Answer

1.5.3 Graph Algorithms: GATE1996-17 top

<https://gateoverflow.in/2769>

Let G be the directed, weighted graph shown in below figure



We are interested in the shortest paths from A .

- Output the sequence of vertices identified by the Dijkstra's algorithm for single source shortest path when the algorithm is started at node A
- Write down sequence of vertices in the shortest path from A to E
- What is the cost of the shortest path from A to E ?

gate1996 algorithms graph-algorithms normal

Answer

1.5.4 Graph Algorithms: GATE1998-1.21, ISRO2008-16 [top](#)

<https://gateoverflow.in/1658>

Which one of the following algorithm design techniques is used in finding all pairs of shortest distances in a graph?

- A. Dynamic programming
- B. Backtracking
- C. Greedy
- D. Divide and Conquer

gate1998 algorithms graph-algorithms easy isro2008

Answer

1.5.5 Graph Algorithms: GATE2000-1.13 [top](#)

<https://gateoverflow.in/636>

The most appropriate matching for the following pairs

X: depth first search	1: heap
Y: breadth-first search	2: queue
Z: sorting	3: stack

is:

- A. X - 1, Y - 2, Z - 3
- B. X - 3, Y - 1, Z - 2
- C. X - 3, Y - 2, Z - 1
- D. X - 2, Y - 3, Z - 1

gate2000 algorithms easy graph-algorithms

Answer

1.5.6 Graph Algorithms: GATE2001-2.14 [top](#)

<https://gateoverflow.in/732>

Consider an undirected, unweighted graph G . Let a breadth-first traversal of G be done starting from a node r . Let $d(r, u)$ and $d(r, v)$ be the lengths of the shortest paths from r to u and v respectively in G . If u is visited before v during the breadth-first traversal, which of the following statements is correct?

- A. $d(r, u) < d(r, v)$
- B. $d(r, u) > d(r, v)$
- C. $d(r, u) \leq d(r, v)$
- D. None of the above

gate2001 algorithms graph-algorithms normal

Answer

1.5.7 Graph Algorithms: GATE2002-12 [top](#)

<https://gateoverflow.in/865>

Fill in the blanks in the following template of an algorithm to compute all pairs shortest path lengths in a directed graph G with $n * n$ adjacency matrix A . $A[i, j]$ equals 1 if there is an edge in G from i to j , and 0 otherwise. Your aim in filling in the blanks is to ensure that the algorithm is correct.

```
INITIALIZATION: For i = 1 ... n
    {For j = 1 ... n
        { if a[i,j] = 0 then P[i,j] = _____ else P[i,j] = _____; }
    }

ALGORITHM: For i = 1 ... n
    {For j = 1 ... n
        {For k = 1 ... n
            {P[_____, _____] = min{_____, _____}; }
        }
    }
}
```

- Copy the complete line containing the blanks in the Initialization step and fill in the blanks.
- Copy the complete line containing the blanks in the Algorithm step and fill in the blanks.
- Fill in the blank: The running time of the Algorithm is $O(\underline{\hspace{2cm}})$.

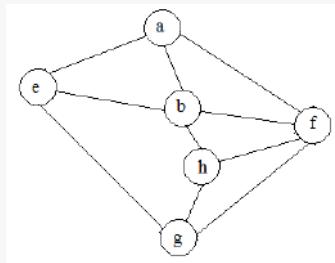
gate2002 algorithms graph-algorithms time-complexity normal descriptive

Answer

1.5.8 Graph Algorithms: GATE2003-21 [top](#)

<https://gateoverflow.in/911>

Consider the following graph:



Among the following sequences:

- I. abeghf
- II. abfehg
- III. abfhge
- IV. aghbe

Which are the depth-first traversals of the above graph?

- A. I, II and IV only
- B. I and IV only
- C. II, III and IV only
- D. I, III and IV only

gate2003 algorithms graph-algorithms normal

Answer

1.5.9 Graph Algorithms: GATE2003-67 [top](#)

<https://gateoverflow.in/954>

Let $G = (V, E)$ be an undirected graph with a subgraph $G_1 = (V_1, E_1)$. Weights are assigned to edges of G as follows.

$$w(e) = \begin{cases} 0, & \text{if } e \in E_1 \\ 1, & \text{otherwise} \end{cases}$$

A single-source shortest path algorithm is executed on the weighted graph (V, E, w) with an arbitrary vertex v_1 of V_1 as the source. Which of the following can always be inferred from the path costs computed?

- A. The number of edges in the shortest paths from v_1 to all vertices of G
- B. G_1 is connected
- C. V_1 forms a clique in G
- D. G_1 is a tree

gate2003 algorithms graph-algorithms normal

Answer

1.5.10 Graph Algorithms: GATE2003-70 [top](#)

<https://gateoverflow.in/957>

Let $G = (V, E)$ be a directed graph with n vertices. A path from v_i to v_j in G is a sequence of vertices $(v_i, v_{i+1}, \dots, v_j)$ such that $(v_k, v_{k+1}) \in E$ for all k in i through $j - 1$. A simple path is a path in which no vertex appears more than once.

Let A be an $n \times n$ array initialized as follows:

$$A[j, k] = \begin{cases} 1, & \text{if } (j, k) \in E \\ 0, & \text{otherwise} \end{cases}$$

Consider the following algorithm:

```
for i=1 to n
    for j=1 to n
        for k=1 to n
            A[j,k] = max(A[j,k], A[j,i] + A[i,k]);
```

Which of the following statements is necessarily true for all j and k after termination of the above algorithm?

- A. $A[j, k] \leq n$
- B. If $A[j, j] \geq n - 1$ then G has a Hamiltonian cycle
- C. If there exists a path from j to k , $A[j, k]$ contains the longest path length from j to k
- D. If there exists a path from j to k , every simple path from j to k contains at most $A[j, k]$ edges

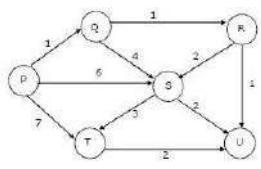
gate2003 algorithms graph-algorithms normal

Answer

1.5.11 Graph Algorithms: GATE2004-44 [top](#)

<https://gateoverflow.in/1041>

Suppose we run Dijkstra's single source shortest-path algorithm on the following edge-weighted directed graph with vertex P as the source.



In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized?

- A. P, Q, R, S, T, U
- B. P, Q, R, U, S, T
- C. P, Q, R, U, T, S
- D. P, Q, T, R, U, S

[gate2004](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.12 Graph Algorithms: GATE2004-81 [top](#)

<https://gateoverflow.in/1075>

Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be connected graphs on the same vertex set V with more than two vertices. If $G_1 \cap G_2 = (V, E_1 \cap E_2)$ is not a connected graph, then the graph $G_1 \cup G_2 = (V, E_1 \cup E_2)$

- A. cannot have a cut vertex
- B. must have a cycle
- C. must have a cut-edge (bridge)
- D. has chromatic number strictly greater than those of G_1 and G_2

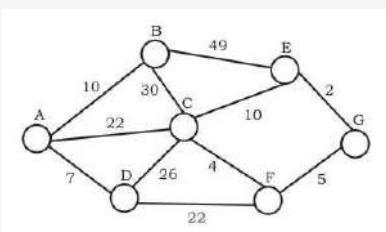
[gate2004](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.13 Graph Algorithms: GATE2004-IT-56 [top](#)

<https://gateoverflow.in/3699>

Consider the undirected graph below:



Using Prim's algorithm to construct a minimum spanning tree starting with node A, which one of the following sequences of edges represents a possible order in which the edges would be added to construct the minimum spanning tree?

- A. $(E, G), (C, F), (F, G), (A, D), (A, B), (A, C)$
- B. $(A, D), (A, B), (A, C), (C, F), (G, E), (F, G)$
- C. $(A, B), (A, D), (D, F), (F, G), (G, E), (F, C)$
- D. $(A, D), (A, B), (D, F), (F, C), (F, G), (G, E)$

[gate2004-it](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.14 Graph Algorithms: GATE2005-38 [top](#)

<https://gateoverflow.in/1374>

Let $G(V, E)$ be an undirected graph with positive edge weights. Dijkstra's single source shortest path algorithm can be implemented using the binary heap data structure with time complexity:

- A. $O(|V|^2)$
- B. $O(|E| + |V|\log |V|)$
- C. $O(|V|\log |V|)$
- D. $O((|E| + |V|)\log |V|)$

[gate2005](#) [algorithms](#) [graph-algorithms](#) [normal](#)

[Answer](#)

1.5.15 Graph Algorithms: GATE2005-82a [top](#)

<https://gateoverflow.in/1404>

Let s and t be two vertices in a undirected graph $G = (V, E)$ having distinct positive edge weights. Let $[X, Y]$ be a partition of V such that $s \in X$ and $t \in Y$. Consider the edge e having the minimum weight amongst all those edges that have one vertex in X and one vertex in Y .

The edge e must definitely belong to:

- A. the minimum weighted spanning tree of G
- B. the weighted shortest path from s to t
- C. each path from s to t
- D. the weighted longest path from s to t

[gate2005](#) [algorithms](#) [graph-algorithms](#) [normal](#)

[Answer](#)

1.5.16 Graph Algorithms: GATE2005-82b [top](#)

<https://gateoverflow.in/82129>

Let s and t be two vertices in a undirected graph $G = (V, E)$ having distinct positive edge weights. Let $[X, Y]$ be a partition of V such that $s \in X$ and $t \in Y$. Consider the edge e having the minimum weight amongst all those edges that have one vertex in X and one vertex in Y .

Let the weight of an edge e denote the congestion on that edge. The congestion on a path is defined to be the maximum of the congestions on the edges of the path. We wish to find the path from s to t having minimum congestion. Which of the following paths is always such a path of minimum congestion?

- A. a path from s to t in the minimum weighted spanning tree
- B. a weighted shortest path from s to t
- C. an Euler walk from s to t
- D. a Hamiltonian path from s to t

[gate2005](#) [algorithms](#) [graph-algorithms](#) [normal](#)

[Answer](#)

1.5.17 Graph Algorithms: GATE2005-IT-14 [top](#)

<https://gateoverflow.in/3759>

In a depth-first traversal of a graph G with n vertices, k edges are marked as tree edges. The number of connected components in G is

- A. k
- B. $k + 1$
- C. $n - k - 1$
- D. $n - k$

gate2005-it algorithms graph-algorithms normal

Answer

1.5.18 Graph Algorithms: GATE2005-IT-15 [top](#)

<https://gateoverflow.in/3760>

In the following table, the left column contains the names of standard graph algorithms and the right column contains the time complexities of the algorithms. Match each algorithm with its time complexity.

	A : $O(m \log n)$
1. Bellman-Ford algorithm	B : $O(n^3)$
2. Kruskal's algorithm	C : $O(nm)$
3. Floyd-Warshall algorithm	D : $O(n + m)$
4. Topological sorting	

- A. $1 \rightarrow C, 2 \rightarrow A, 3 \rightarrow B, 4 \rightarrow D$
- B. $1 \rightarrow B, 2 \rightarrow D, 3 \rightarrow C, 4 \rightarrow A$
- C. $1 \rightarrow C, 2 \rightarrow D, 3 \rightarrow A, 4 \rightarrow B$
- D. $1 \rightarrow B, 2 \rightarrow A, 3 \rightarrow C, 4 \rightarrow D$

gate2005-it algorithms graph-algorithms normal

Answer

1.5.19 Graph Algorithms: GATE2005-IT-84a [top](#)

<https://gateoverflow.in/3856>

A sink in a directed graph is a vertex i such that there is an edge from every vertex $j \neq i$ to i and there is no edge from i to any other vertex. A directed graph G with n vertices is represented by its adjacency matrix A , where $A[i][j] = 1$ if there is an edge directed from vertex i to j and 0 otherwise. The following algorithm determines whether there is a sink in the graph G .

```
i = 0;
do {
    j = i + 1;
    while ((j < n) && E1) j++;
    if (j < n) E2;
} while (j < n);
flag = 1;
for (j = 0; j < n; j++)
    if ((j != i) && E3) flag = 0;
if (flag) printf("Sink exists");
else printf ("Sink does not exist");
```

Choose the correct expressions for E_1 and E_2

- A. $E_1 : A[i][j]$ and $E_2 : i = j$;
- B. $E_1 : !A[i][j]$ and $E_2 : i = j + 1$;
- C. $E_1 : !A[i][j]$ and $E_2 : i = j$;
- D. $E_1 : A[i][j]$ and $E_2 : i = j + 1$;

gate2005-it algorithms graph-algorithms normal

Answer

1.5.20 Graph Algorithms: GATE2005-IT-84b [top](#)

<https://gateoverflow.in/3857>

A sink in a directed graph is a vertex i such that there is an edge from every vertex $j \neq i$ to i and there is no edge from i to any other vertex. A directed graph G with n vertices is represented by its adjacency matrix A , where $A[i][j] = 1$ if there is an edge directed from vertex i to j and 0 otherwise. The following algorithm determines whether there is a sink in the graph G .

```
i = 0;
do {
    j = i + 1;
    while ((j < n) && E1) j++;
    if (j < n) E2;
} while (j < n);
flag = 1;
for (j = 0; j < n; j++)
    if ((j != i) && E3) flag = 0;
if (flag) printf("Sink exists");
else printf ("Sink does not exist");
```

Choose the correct expression for E_3

- A. $(A[i][j] \&& !A[j][i])$
- B. $(!A[i][j] \&& A[j][i])$
- C. $(!A[i][j] || A[j][i])$
- D. $(A[i][j] || !A[j][i])$

gate2005-it algorithms graph-algorithms normal

Answer

1.5.21 Graph Algorithms: GATE2006-12 [top](#)

<https://gateoverflow.in/891>

To implement Dijkstra's shortest path algorithm on unweighted graphs so that it runs in linear time, the data structure to be used is:

- A. Queue
- B. Stack
- C. Heap
- D. B-Tree

gate2006 algorithms graph-algorithms easy

Answer

1.5.22 Graph Algorithms: GATE2006-48 [top](#)

<https://gateoverflow.in/1824>

Let T be a depth first search tree in an undirected graph G . Vertices u and v are leaves of this tree T . The degrees of both u and v in G are at least 2. which one of the following statements is true?

- A. There must exist a vertex w adjacent to both u and v in G
 B. There must exist a vertex w whose removal disconnects u and v in G
 C. There must exist a cycle in G containing u and v
 D. There must exist a cycle in G containing u and all its neighbours in G

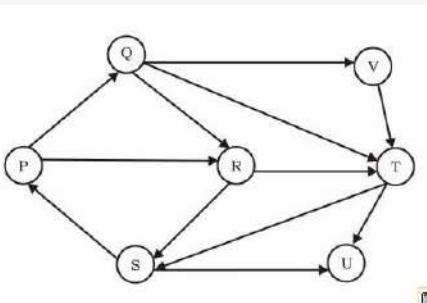
gate2006 algorithms graph-algorithms normal

Answer

1.5.23 Graph Algorithms: GATE2006-IT-46 top

<https://gateoverflow.in/3589>

Which of the following is the correct decomposition of the directed graph given below into its strongly connected components?



- A. $\{P, Q, R, S\}, \{T\}, \{U\}, \{V\}$
 B. $\{P, Q, R, S, T, V\}, \{U\}$
 C. $\{P, Q, S, T, V\}, \{R\}, \{U\}$
 D. $\{P, Q, R, S, T, U, V\}$

gate2006-it algorithms graph-algorithms normal

Answer

1.5.24 Graph Algorithms: GATE2006-IT-47 top

<https://gateoverflow.in/3590>

Consider the depth-first-search of an undirected graph with 3 vertices P , Q , and R . Let discovery time $d(u)$ represent the time instant when the vertex u is first visited, and finish time $f(u)$ represent the time instant when the vertex u is last visited. Given that

$$\begin{array}{ll}
 d(P) & f(P) \\
 = 5 & = 12 \\
 \text{units} & \text{units} \\
 \\
 d(Q) & f(Q) \\
 = 6 & = 10 \\
 \text{units} & \text{units} \\
 \\
 d(R) & f(R) \\
 = 14 & = 18 \\
 \text{unit} & \text{units}
 \end{array}$$

Which one of the following statements is TRUE about the graph?

- A. There is only one connected component
 B. There are two connected components, and P and R are connected
 C. There are two connected components, and Q and R are connected
 D. There are two connected components, and P and Q are connected

gate2006-it algorithms graph-algorithms normal

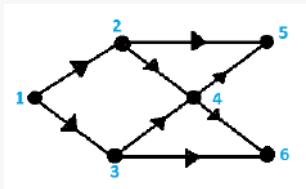
Answer**1.5.25 Graph Algorithms: GATE2007-41** top<https://gateoverflow.in/1239>

In an unweighted, undirected connected graph, the shortest path from a node S to every other node is computed most efficiently, in terms of *time complexity*, by

- A. Dijkstra's algorithm starting from S .
- B. Warshall's algorithm.
- C. Performing a DFS starting from S .
- D. Performing a BFS starting from S .

[gate2007](#) [algorithms](#) [graph-algorithms](#) [easy](#)**Answer****1.5.26 Graph Algorithms: GATE2007-5** top<https://gateoverflow.in/31821>

Consider the DAG with $V = \{1, 2, 3, 4, 5, 6\}$ shown below.



Which of the following is not a topological ordering?

- A. 1 2 3 4 5 6
- B. 1 3 2 4 5 6
- C. 1 3 2 4 6 5
- D. 3 2 4 1 6 5

[gate2007](#) [algorithms](#) [graph-algorithms](#)**Answer****1.5.27 Graph Algorithms: GATE2007-IT-24** top<https://gateoverflow.in/3457>

A depth-first search is performed on a directed acyclic graph. Let $d[u]$ denote the time at which vertex u is visited for the first time and $f[u]$ the time at which the DFS call to the vertex u terminates. Which of the following statements is always TRUE for all edges (u, v) in the graph ?

- A. $d[u] < d[v]$
- B. $d[u] < f[v]$
- C. $f[u] < f[v]$
- D. $f[u] > f[v]$

[gate2007-it](#) [algorithms](#) [graph-algorithms](#) [normal](#)**Answer**

1.5.28 Graph Algorithms: GATE2007-IT-3, UGCNET-June2012-III-34 [top](#)

<https://gateoverflow.in/3434>

Consider a weighted, undirected graph with positive edge weights and let uv be an edge in the graph. It is known that the shortest path from the source vertex s to u has weight 53 and the shortest path from s to v has weight 65. Which one of the following statements is always TRUE?

- A. Weight $(u, v) \leq 12$
- B. Weight $(u, v) = 12$
- C. Weight $(u, v) \geq 12$
- D. Weight $(u, v) > 12$

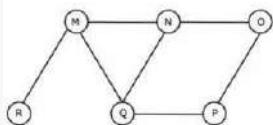
[gate2007-it](#) [algorithms](#) [graph-algorithms](#) [normal](#) [ugcnetjune2012iii](#)

Answer

1.5.29 Graph Algorithms: GATE2008-19 [top](#)

<https://gateoverflow.in/417>

The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is:



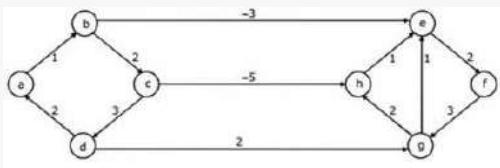
- A. $MNOPQR$
- B. $NQMPOR$
- C. $QMNPOR$
- D. $QMNPOR$

[gate2008](#) [normal](#) [algorithms](#) [graph-algorithms](#)

Answer

1.5.30 Graph Algorithms: GATE2008-45 [top](#)

<https://gateoverflow.in/457>



Dijkstra's single source shortest path algorithm when run from vertex a in the above graph, computes the correct shortest path distance to

- A. only vertex a
- B. only vertices a, e, f, g, h
- C. only vertices a, b, c, d
- D. all the vertices

[gate2008](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.31 Graph Algorithms: GATE2008-7 [top](#)

<https://gateoverflow.in/405>

The most efficient algorithm for finding the number of connected components in an undirected graph on n vertices and m edges has time complexity

- A. $\Theta(n)$
- B. $\Theta(m)$
- C. $\Theta(m + n)$
- D. $\Theta(mn)$

[gate2008](#) [algorithms](#) [graph-algorithms](#) [time-complexity](#) [normal](#)

Answer

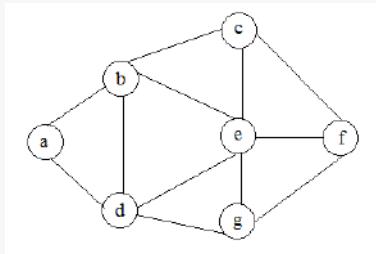
1.5.32 Graph Algorithms: GATE2008-IT-47 [top](#)

<https://gateoverflow.in/3357>

Consider the following sequence of nodes for the undirected graph given below:

1. $a b e f d g c$
2. $a b e f c g d$
3. $a d g e b c f$
4. $a d b c g e f$

A Depth First Search (DFS) is started at node a . The nodes are listed in the order they are first visited. Which of the above is/are possible output(s)?



- A. 1 and 3 only
- B. 2 and 3 only
- C. 2, 3 and 4 only
- D. 1, 2 and 3 only

[gate2008-it](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.33 Graph Algorithms: GATE2009-13 [top](#)

<https://gateoverflow.in/1305>

Which of the following statement(s) is/are correct regarding Bellman-Ford shortest path algorithm?

P: Always finds a negative weighted cycle, if one exists.

Q: Finds whether any negative weighted cycle is reachable from the source.

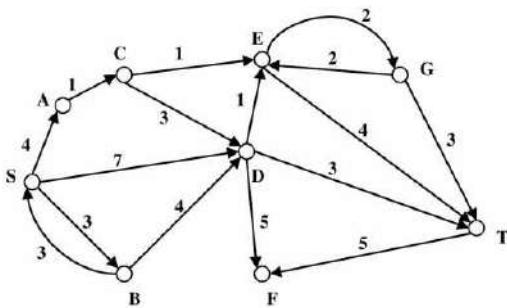
- A. P only
- B. Q only
- C. Both P and Q
- D. Neither P nor Q

[gate2009](#) [algorithms](#) [graph-algorithms](#) [normal](#)
Answer

1.5.34 Graph Algorithms: GATE2012-40 [top](#)

<https://gateoverflow.in/1765>

Consider the directed graph shown in the figure below. There are multiple shortest paths between vertices S and T . Which one will be reported by Dijkstra's shortest path algorithm? Assume that, in any iteration, the shortest path to a vertex v is updated only when a strictly shorter path to v is discovered.



- A. SDT
- B. SBDT
- C. SACDT
- D. SACET

[gate2012](#) [algorithms](#) [graph-algorithms](#) [normal](#)
Answer

1.5.35 Graph Algorithms: GATE2013-19 [top](#)

<https://gateoverflow.in/1441>

What is the time complexity of Bellman-Ford single-source shortest path algorithm on a complete graph of n vertices?

- A. $\theta(n^2)$
- B. $\theta(n^2 \log n)$
- C. $\theta(n^3)$
- D. $\theta(n^3 \log n)$

[gate2013](#) [algorithms](#) [graph-algorithms](#) [normal](#)
Answer

1.5.36 Graph Algorithms: GATE2014-1-11 [top](#)

<https://gateoverflow.in/1771>

Let G be a graph with n vertices and m edges. What is the tightest upper bound on the running time of Depth First Search on G , when G is represented as an adjacency matrix?

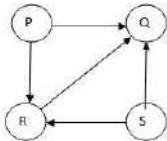
- A. $\Theta(n)$
- B. $\Theta(n + m)$
- C. $\Theta(n^2)$
- D. $\Theta(m^2)$

[gate2014-1](#) [algorithms](#) [graph-algorithms](#) [normal](#)
Answer

1.5.37 Graph Algorithms: GATE2014-1-13 top

<https://gateoverflow.in/1779>

Consider the directed graph below given.



Which one of the following is **TRUE**?

- A. The graph does not have any topological ordering.
- B. Both PQRS and SRQP are topological orderings.
- C. Both PSRQ and SPRQ are topological orderings.
- D. PSRQ is the only topological ordering.

[gate2014-1](#) [graph-algorithms](#) [easy](#)

Answer

1.5.38 Graph Algorithms: GATE2014-2-14 top

<https://gateoverflow.in/1969>

Consider the tree arcs of a BFS traversal from a source node W in an unweighted, connected, undirected graph. The tree T formed by the tree arcs is a data structure for computing

- A. the shortest path between every pair of vertices.
- B. the shortest path from W to every vertex in the graph.
- C. the shortest paths from W to only those nodes that are leaves of T .
- D. the longest path in the graph.

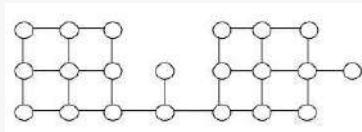
[gate2014-2](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.39 Graph Algorithms: GATE2014-3-13 top

<https://gateoverflow.in/2047>

Suppose depth first search is executed on the graph below starting at some unknown vertex. Assume that a recursive call to visit a vertex is made only after first checking that the vertex has not been visited earlier. Then the maximum possible recursion depth (including the initial call) is _____.



[gate2014-3](#) [algorithms](#) [graph-algorithms](#) [numerical-answers](#) [normal](#)

Answer

1.5.40 Graph Algorithms: GATE2015-1-45 top

<https://gateoverflow.in/8321>

Let $G = (V, E)$ be a simple undirected graph, and s be a particular vertex in it called the source. For $x \in V$, let $d(x)$ denote the shortest distance in G from s to x . A breadth first search (BFS) is

performed starting at s . Let T be the resultant BFS tree. If (u, v) is an edge of G that is not in T , then which one of the following CANNOT be the value of $d(u) - d(v)$?

- A. -1
- B. 0
- C. 1
- D. 2

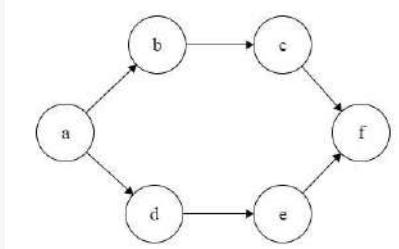
[gate2015-1](#) [algorithms](#) [graph-algorithms](#) [normal](#)

[Answer](#)

1.5.41 Graph Algorithms: GATE2016-1-11 [top](#)

<https://gateoverflow.in/39669>

Consider the following directed graph:



The number of different topological orderings of the vertices of the graph is _____.

[gate2016-1](#) [algorithms](#) [graph-algorithms](#) [normal](#) [numerical-answers](#)

[Answer](#)

1.5.42 Graph Algorithms: GATE2016-2-11 [top](#)

<https://gateoverflow.in/39563>

Breadth First Search (BFS) is started on a binary tree beginning from the root vertex. There is a vertex t at a distance four from the root. If t is the n^{th} vertex in this BFS traversal, then the maximum possible value of n is _____

[gate2016-2](#) [algorithms](#) [graph-algorithms](#) [normal](#) [numerical-answers](#)

[Answer](#)

1.5.43 Graph Algorithms: GATE2016-2-41 [top](#)

<https://gateoverflow.in/39620>

In an adjacency list representation of an undirected simple graph $G = (V, E)$, each edge (u, v) has two adjacency list entries: $[v]$ in the adjacency list of u , and $[u]$ in the adjacency list of v . These are called twins of each other. A twin pointer is a pointer from an adjacency list entry to its twin. If $|E| = m$ and $|V| = n$, and the memory size is not a constraint, what is the time complexity of the most efficient algorithm to set the twin pointer in each entry in each adjacency list?

- A. $\Theta(n^2)$
- B. $\Theta(n + m)$
- C. $\Theta(m^2)$
- D. $\Theta(n^4)$

[gate2016-2](#) [algorithms](#) [graph-algorithms](#) [normal](#)

[Answer](#)

1.5.44 Graph Algorithms: GATE2017-1-26 [top](#)

<https://gateoverflow.in/118306>

Let $G = (V, E)$ be *any* connected, undirected, edge-weighted graph. The weights of the edges in E are positive and distinct. Consider the following statements:

- Minimum Spanning Tree of G is always unique.
- Shortest path between any two vertices of G is always unique.

Which of the above statements is/are necessarily true?

- A. I only
- B. II only
- C. both I and II
- D. neither I nor II

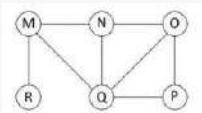
gate2017-1 algorithms graph-algorithms normal

Answer

1.5.45 Graph Algorithms: GATE2017-2-15 [top](#)

<https://gateoverflow.in/118196>

The Breadth First Search (BFS) algorithm has been implemented using the queue data structure. Which one of the following is a possible order of visiting the nodes in the graph below?



- A. MNOPQR
- B. NQMPOR
- C. QMNROP
- D. POQNMR

gate2017-2 algorithms graph-algorithms

Answer

1.5.46 Graph Algorithms: Gate2000-2.19 [top](#)

<https://gateoverflow.in/4208>

Let G be an undirected graph. Consider a depth-first traversal of G , and let T be the resulting depth-first search tree. Let u be a vertex in G and let v be the first new (unvisited) vertex visited after visiting u in the traversal. Which of the following statement is always true?

- A. $\{u, v\}$ must be an edge in G , and u is a descendant of v in T
- B. $\{u, v\}$ must be an edge in G , and v is a descendant of u in T
- C. If $\{u, v\}$ is not an edge in G then u is a leaf in T
- D. If $\{u, v\}$ is not an edge in G then u and v must have the same parent in T

gate2000 algorithms graph-algorithms normal

Answer

1.5.47 Graph Algorithms: TIFR2013-B-5 [top](#)

<https://gateoverflow.in/25666>

Given a weighted directed graph with n vertices where edge weights are integers (positive, zero, or negative), determining whether there are paths of arbitrarily large weight can be performed in time

- A. $O(n)$

- B. $O(n \cdot \log(n))$ but not $O(n)$
- C. $O(n^{1.5})$ but not $O(n \log n)$
- D. $O(n^3)$ but not $O(n^{1.5})$
- E. $O(2^n)$ but not $O(n^3)$

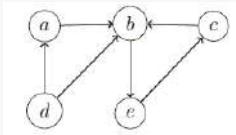
tifr2013 | algorithms | graph-algorithms

[Answer](#)

1.5.48 Graph Algorithms: TIFR2014-B-3 [top](#)

<https://gateoverflow.in/27137>

Consider the following directed graph.



Suppose a depth-first traversal of this graph is performed, assuming that whenever there is a choice, the vertex earlier in the alphabetical order is to be chosen. Suppose the number of tree edges is T , the number of back edges is B and the number of cross edges is C . Then

- a. $B = 1, C = 1$, and $T = 4$.
- b. $B = 0, C = 2$, and $T = 4$.
- c. $B = 2, C = 1$, and $T = 3$.
- d. $B = 1, C = 2$, and $T = 3$.
- e. $B = 2, C = 2$, and $T = 1$.

tifr2014 | algorithms | graph-algorithms

[Answer](#)

Answers: Graph Algorithms

1.5.1 Graph Algorithms: GATE1994-1.22 [top](#)

<https://gateoverflow.in/2465>



Selected Answer

Answer is **B**.

- A. True.
- B. False.
- C. True.
- D. True.

11 votes

-- Rajarshi Sarkar (34.1k points)

1.5.2 Graph Algorithms: GATE1994-24 [top](#)

<https://gateoverflow.in/2520>



Selected Answer

- a. While adding vertex u to I it should not have an edge with any node in I .
- b. The algorithm runs till V is empty (in $O(n)$ time) and is checking u with each vertex v in set I (in $O(n)$ time). So, overall complexity $O(n^2)$.

15 votes

-- Rajarshi Sarkar (34.1k points)

1.5.3 Graph Algorithms: GATE1996-17 top

<https://gateoverflow.in/2769>

Selected Answer

DIJKSTRA(G, w, s)

- 1 **INITIALIZE-SINGLE-SOURCE(G, s)**
- 2 $S = \emptyset$
- 3 $Q = G.V$
- 4 **while** $Q \neq \emptyset$
- 5 $u = \text{EXTRACT-MIN}(Q)$
- 6 $S = S \cup \{u\}$
- 7 **for each vertex** $v \in G.Adj[u]$
- 8 **RELAX**(u, v, w)

Correct Solutions:

(A).

$$Q = \left\{ \begin{matrix} [0] & [\infty] & [\infty] & [\infty] & [\infty] & [\infty] \\ A, & B, & C, & D, & E, & F \end{matrix} \right\}$$

- First we visit A as it is the one with smallest distance 0
- Relax operation updates distances to B,C,F as 6, 90, 70 respectively

$$Q = \left\{ \begin{matrix} [6] & [90] & [\infty] & [\infty] & [70] \\ B, & C, & D, & E, & F \end{matrix} \right\}$$

- B is next visited as its distance is now 6
- Relax operation updates distance to D as $41 + 6 = 47$

$$Q = \left\{ \begin{matrix} [90] & [47] & [\infty] & [70] \\ C, & D, & E, & F \end{matrix} \right\}$$

- D is next visited as its distance is now 47
- Relax operation updates distance to C as $47 + 12 = 59$

$$Q = \left\{ \begin{matrix} [59] & [\infty] & [70] \\ C, & E, & F \end{matrix} \right\}$$

- C is next visited as its distance is now 59
- Relax operation updates distance to F as $59 + 10 = 69$

$$Q = \left\{ \begin{matrix} [\infty] & [69] \\ E, & F \end{matrix} \right\}$$

- F is next visited as its distance is now 69
- Relax operation updates distance to E as $69 + 15 = 84$

$$Q = \left\{ \begin{matrix} [84] \\ E \end{matrix} \right\}$$

- Finally E is visited.

So, the sequence of node visits are A, B, D, C, F, E

(B). Sequence of vertices in the shortest path from A to E: A – B – D – C – F – E

(C). Cost of the shortest path from A to E = 84.

3 votes

-- Manu Thakur (39.6k points)

1.5.4 Graph Algorithms: GATE1998-1.21, ISRO2008-16 [top](#)



Selected Answer

<https://gateoverflow.in/1658>

Answer is (A) because Floyd Warshall algorithm is used to find all shortest paths which is a dynamic programming approach.

15 votes

-- shashi shekhar (569 points)

1.5.5 Graph Algorithms: GATE2000-1.13 [top](#)

<https://gateoverflow.in/636>



Selected Answer

Answer is C.

X - 3 **DFS** uses **stack** implicitly
Y - 2 **BFS** uses **queue** explicitly in Algo
Z - 1 **Heap-Heapsort**

16 votes

-- Akash Kanase (42.5k points)

1.5.6 Graph Algorithms: GATE2001-2.14 [top](#)

<https://gateoverflow.in/732>



Selected Answer

Answer is (C).

BFS is used to count shortest path from source (If all path costs are 1)

Now, if u is visited before v it means 2 things:

1. Either u is closer to v , or,
2. If u & v are same distance from r , then our BFS algo chose to visit u before v .

24 votes

-- Akash Kanase (42.5k points)

1.5.7 Graph Algorithms: GATE2002-12 [top](#)

<https://gateoverflow.in/865>

```

INITIALIZATION: For i = 1 ... n
    {For j = 1 ... n
        { if a[i,j] = 0 then P[i,j] = infinite
        // i.e. if there is no direct path then put infinite
        else P[i,j] = a[i,j];
        }
    }
ALGORITHM:
For i = 1 ... n
{For j = 1 ... n
    {For k = 1 ... n
        {
            P[i, j] = min( p[i,j] , p[i,k] + p[k,j])
        };
    }
}

```

Time complexity $O(n^3)$

This algorithm is for weighted graph but it will work for unweighted graph too because if $p[i,j] = 1, p[i,k] = 1$ and $p[k,j] = 1$ then according to the algorithm $p[i,j] = \min(p[i,j], p[i,k] + p[k,j]) = \min(1, 2) = 1$

And all the other cases are also satisfied.(like as if $p[i,j]$ was 0 in last iteration *nd* there exist a path via k)

12 votes

-- Saurav Kumar Gupta (2.2k points)

1.5.8 Graph Algorithms: GATE2003-21 top

<https://gateoverflow.in/911>



Selected Answer

For GATE purpose, without actually applying DFS, we can answer by just seeing options. In DFS, we go in depth first i.e., one node to another in depth first order.

Here, $abfehg$ is not possible as we can not go from f to e directly.
Thus, option (D) is correct.

In all the other options we can reach directly from the node to the next node.

So, just visualize and do.

12 votes

-- Monanshi Jain (9.3k points)

1.5.9 Graph Algorithms: GATE2003-67 top

<https://gateoverflow.in/954>



Selected Answer

After applying the shortest path algorithm, check cost of vertex from source to every vertex in G_1 . If G_1 is connected all these costs must be 0 as edge weights of subgraph G_1 is 0 and that should be the shortest path. If cost is not 0, to at least one vertex in G_1 (not necessarily G), then G_1 is disconnected.

Answer is B.

38 votes

-- Anurag Semwal (8k points)

1.5.10 Graph Algorithms: GATE2003-70 top

<https://gateoverflow.in/957>



D is correct.

Consider a graph with 2 nodes and one edge from V_1 to V_2 ,

Running the above algorithm will result in A being

A	1	2
1	1	2
2	1	2

Clearly options B and C are wrong. Since

1. $A[1][1]$ and $A[2][2] > n - 1$ and there exists no Hamiltonian cycle. Hence invalid.
2. The longest path between V_1 and V_2 is 1, but $A[1][2]$ is 2, which is invalid. And no path between V_2 and V_1 yet $A[2][1] = 1$ // **it should be max cost path between j and k, not path length.**

Hence A or D could be valid.

Now consider a graph with 2 nodes and two edges, one from V_1 and V_2 and other form V_2 and V_1 . Running the above algorithm will result in A being

A	1	2
1	2	3
2	3	4

Hence option A is invalid, as $A[i][j]$ can be $> n$

D is correct

25 votes

-- ryan sequeira (3.4k points)

1.5.11 Graph Algorithms: GATE2004-44 top

<https://gateoverflow.in/1041>



Answer is (B). In Dijkstra's algorithm at each point we choose the smallest weight edge which starts from any one of the vertices in the shortest path found so far and add it to the shortest path.

18 votes

-- gate_asp (749 points)

1.5.12 Graph Algorithms: GATE2004-81 top

<https://gateoverflow.in/1075>



There are two connected graphs G_1 and G_2 , with same vertices. In least case, both graphs will have $n - 1$ edges with n vertices as both the given graphs are connected.

When we UNION both the graphs as $G = G_1 \cup G_2$ then G will be having at most $(2n - 2)$ edges in the best case when both the graphs don't have any common edges between them (or) at least more than $n - 1$ edges if they have few common edges between them as the intersection of these two graphs is not connected.

Suppose if both the graphs G_1 and G_2 have exactly the same edges then their intersection will be a connected graph.

A graph with

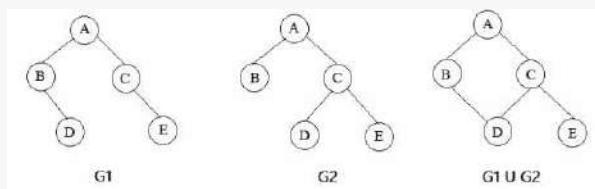
n vertices and more than $n - 1$ edges will definitely have a cycle.

Hence (B) is the correct option.

Upvote 23 votes

-- Vijay Thakur (17.1k points)

Take a tree for example



- A. False. Every vertex of tree(other than leaves) is a cut vertex.
- B. True.
- C. False. Without E in G_1 and G_2 , $G_1 \cup G_2$ has no bridge.
- D. False. $G_1 \cup G_2$, G_1 , G_2 three graphs have same chromatic number of 2.

Upvote 26 votes

-- srestha (87k points)

1.5.13 Graph Algorithms: GATE2004-IT-56 top

<https://gateoverflow.in/3699>



Answer is **D**.

A and *B* produce disconnected components with the GIVEN order in options which is NEVER allowed by prim's algorithm.

C produces connected component every instant a new edge is added BUT when first vertex is chosen(first vertex is chosen randomly) first edge must be the minimum weight edge that is chosen . Therefore, (A, D) MUST be chosen BEFORE (A, B) . Therefore, *C* is FALSE.

Upvote 16 votes

-- Sandeep_Uniyal (7.5k points)

1.5.14 Graph Algorithms: GATE2005-38 top

<https://gateoverflow.in/1374>



Option (D) : Binary heap. $|E|$ decrease key operations and each taking $O(\log |V|)$ time + $|V|$

extract-min operations each taking $O(\log |V|)$.

Option (B) : Fibonacci heap. $|E|$ decrease key operations and each taking $O(1)$ time + $|V|$ extract-min operations each taking $O(\log |V|)$.

Option (A) : Array. Finding min-vertex in each iteration takes $O(V)$ and this needs to be done $|V|$ times.

Binomial Heap is same as Binary heap here, as the critical operations are decrease key and extract-min.

38 votes

-- Gate Keeda (19.6k points)

1.5.15 Graph Algorithms: GATE2005-82a top

<https://gateoverflow.in/1404>

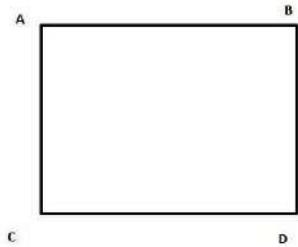


Selected Answer

For 82a: The answer should be Option A because edge e is the lightest safe edge connecting X and Y so the minimum spanning tree of G must contain e (Greedy and optimal choice).

While option (B) might seem correct but it is not always true. One such case is when G is not connected therefore there might not be any path between s and t .

Since the question is about definitely TRUE, (B) is incorrect and (A) is the only correct option.



Lets say $AC = 1$, $CD = 2$, $BD = 3$ and $AB = 4$

Then if $s = A$ and $t = B$ then AC is the lightest edge crossing X and Y where $X = A$ and $Y = C, B, D$

But clearly AC is not on the shortest path from A to B . The shortest path is $AB = 4$.

20 votes

-- chandan1223 (217 points)

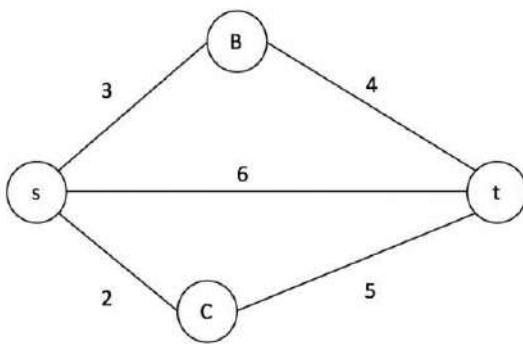
1.5.16 Graph Algorithms: GATE2005-82b top

<https://gateoverflow.in/82129>



Selected Answer

Here answer should be A.



Here shortest path will give 6.

Spanning tree contains edges of weights 2,3,4 so congestion in this case is $\max(2, 3, 4)$, that is, 4. For path s to t , overall congestion is $\max(3, 4) = 4$ but total weight is 7.

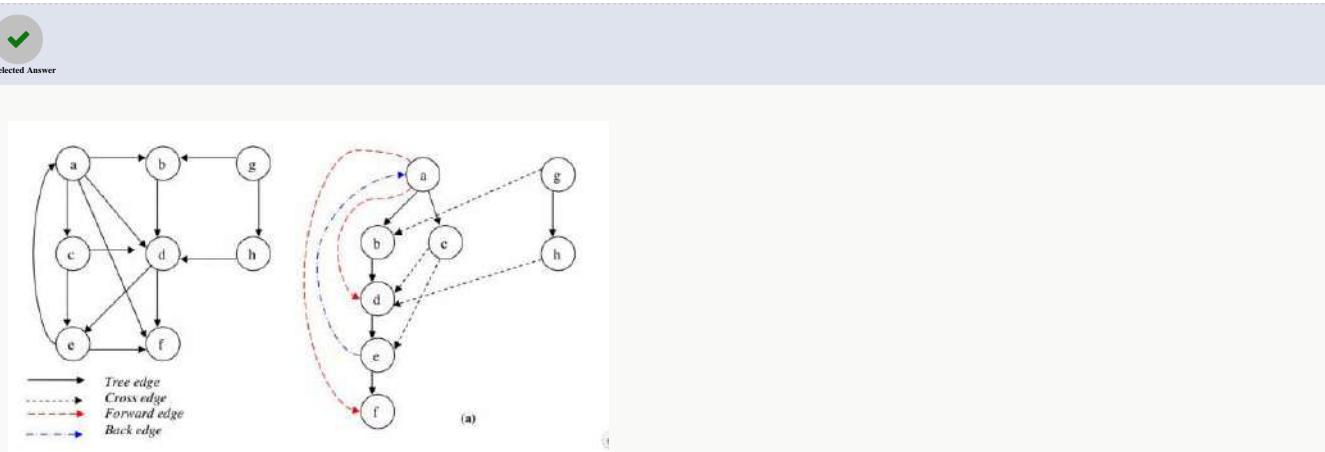
Option C and D are I think not related to this question.

 13 votes

-- **papesh** (25.7k points)

1.5.17 Graph Algorithms: GATE2005-IT-14

<https://gateoverflow.in/3759>



Tree edges are those edges which appear in the final DFS forest. For example in case of connected graph (i.e. resulting DFS forest containing only one tree), if we run DFS over it, edges belonging to the resulting DFS tree are called tree edges.

Let us assume the graph has x number of connected (or strongly connected in case of a directed graph) components. And assume 1st component has K_1 tree edges, 2nd component has K_2 tree edges and x th component has K_x tree edges.

Such that $K_1 + K_2 + K_3 + \dots + K_x = K$ (= total)

Or in other way we can imagine like, the final DFS forest has x trees and those trees are having $K_1, K_2, K_3, \dots, K_x$ edges respectively.

Now we know that a tree having K_x edges contains $K_x + 1$ nodes. Similarly a tree having K_1 edges contains $K_1 + 1$ nodes, etc. and so on.

So, Summation of nodes in each tree = n

$$(K_1 + 1) + (K_2 + 1) + (K_3 + 1) + \dots + (K_x + 1) = n \implies (K_1 + K_2 + K_3 + \dots + K_x) + x = n$$

 36 votes

-- Debashish Deka (56.7k points)

1.5.18 Graph Algorithms: GATE2005-IT-15 [top](#)

<https://gateoverflow.in/3760>



Selected Answer

1. Bellman-Ford algorithm \Rightarrow option(C), $O(nm)$. Assuming n as edges, m as vertices, for every vertex we relax all edges. $m * n$, $O(mn)$.
2. Kruskal's algorithm \Rightarrow Remaining Option (A) : $O(m \log n)$.
3. Floyd-Warshall algorithm \Rightarrow option (B), Dynamic Programming Algo, $O(N^3)$.
4. Topological sorting \Rightarrow option(D), boils down to DFS, $O(V + E)$.

Answer

(A).

15 votes

-- Akash Kanase (42.5k points)

1.5.19 Graph Algorithms: GATE2005-IT-84a [top](#)

<https://gateoverflow.in/3856>



Selected Answer

If there is a sink in the graph, the adjacency matrix will contain all 1's (except diagonal) in one column and all 0's (except diagonal) in the corresponding row of that vertex. The given algorithm is a smart way of doing this as it finds the sink in $O(n)$ time complexity.

The first part of the code, is finding if there is any vertex which doesn't have any outgoing edge to any vertex coming after it in adjacency matrix. The smart part of the code is E_2 , which makes rows skip when there is no edge from i to it, making it impossible for them to form a sink. This is done through

$$E_1 : A[i][j]$$

and

$$E_2 : i = j;$$

E_1 makes sure that there is no edge from i to j and i is a potential sink till $A[i][j]$ becomes 1. If $A[i][j]$ becomes 1, i can no longer be a sink, similarly all previous j can also not be a sink (as there was no edge from i to them and a sink requires an edge from all other vertices). Now, the next potential candidate for sink is j . So, in E_2 , we must make $i = j$.

So, answer is (C)

For E_3 , https://gateoverflow.in/3857/gate2005-it_84b

28 votes

-- Arjun Suresh (350k points)

1.5.20 Graph Algorithms: GATE2005-IT-84b [top](#)

<https://gateoverflow.in/3857>



Selected Answer

If there is a sink in the graph, the adjacency matrix will contain all 1s (except diagonal) in one column and all 0s (except diagonal) in the corresponding row of that vertex. The given algorithm is a smart way of doing this as it finds the sink in $O(n)$ time complexity.

The first part of the code, is finding if there is any vertex which does not have any outgoing edge to any vertex coming after it in adjacency matrix. The smart part of the code is E_2 , which makes

rows skip when there is no edge from i to it, making it impossible them to form a sink. This is done through

$$E_1 : !A[i][j]$$

and

$$E_2 : i = j;$$

E_1 makes sure that there is no edge from i to j and i is a potential sink till $A[i][j]$ becomes 1. If $A[i][j]$ becomes 1, i can no longer be a sink, similarly all previous j can also not be a sink (as there was no edge from i to them and a sink requires an edge from all other vertices). Now, the next potential candidate for sink is j . So, in E_2 , we must make $i = j$.

Now, the loop breaks when we found a potential sink- that is a vertex which does not have any outgoing edge to any coming after it in adjacency matrix. So, if the column in which this vertex comes is all 1s and the row is all 0s (except diagonal), this is the sink. Otherwise there is no sink in the graph. So, E_3 is checking this condition.

But in the code flag is used for storing the state that sink is present or not. And as per the usage of flag in code, by default sink is considered present. So, the condition in E_3 must make flag = 0, if the found i is not a sink. So, the condition should be:

$$A[i][j] \ || \ !A[j][i]$$

So, **(D) is the answer.**

42 votes

-- Arjun Suresh (350k points)

1.5.21 Graph Algorithms: GATE2006-12 top

<https://gateoverflow.in/891>



Answer is **A: Queue**

We can find single source shortest path in [unweighted graph](#) by using Breadth First Search (BFS) algorithm by using "Queue" data structure , in time $O(m + n)$ (i.e. linear with respect to the number of vertices and edges.)

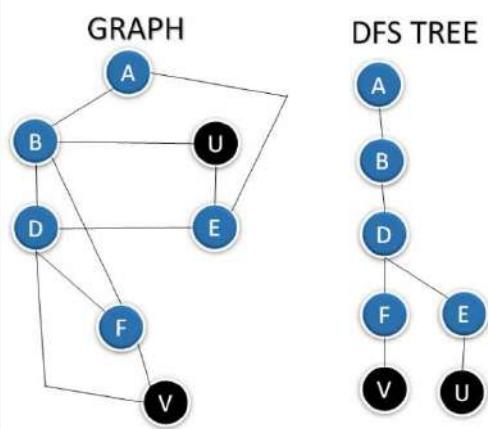
42 votes

-- Mithlesh Upadhyay (5.8k points)

1.5.22 Graph Algorithms: GATE2006-48 top

<https://gateoverflow.in/1824>





One diagram, which is eliminating option A, B, C.
Hence **D** is the answer.

29 votes

-- Ahwan Mishra (10.5k points)

1.5.23 Graph Algorithms: GATE2006-IT-46 top

<https://gateoverflow.in/3589>



Here the answer is **B**.

A graph is said to be **strongly connected** if every vertex is reachable from every other vertex.

The strongly connected component is always maximal that is if x is strongly connected component there should not exist another strongly connected component which contains x .

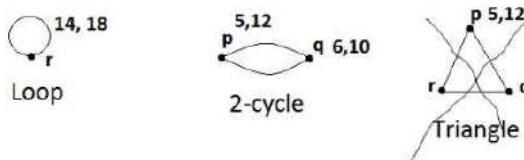
If we take R as a strongly connected component but which is part of $PQRS$ and $PQRS$ is part of $PQRSVT$.

22 votes

-- papesh (25.7k points)

1.5.24 Graph Algorithms: GATE2006-IT-47 top

<https://gateoverflow.in/3590>



As seen in question, after 10 we have to go for p again and since p is finished and then r is started so r must be disconnected because if there is edges from q to r then r must be visited before q and p end.

D is answer.

15 votes

-- Prashant Singh (59.8k points)

1.5.25 Graph Algorithms: GATE2007-41 top

<https://gateoverflow.in/1239>



Dijkstra and Warshall's algorithm used only for weighted graph.

Both *DFS* and *BFS* can be used for finding path between 2 vertices in undirected and unweighted graph but *BFS* can only give the shortest path as concerned in given question. So, **BFS is answer**.

Note : Finding only path(*DFS*) and finding shortest path(*BFS*) matters a lot.

Just Read:

<https://www.quora.com/What-are-the-advantages-of-using-BFS-over-DFS-or-using-DFS-over-BFS-What-are-the-applications-and-downsides-of-each>

19 votes

-- Rajesh Pradhan (22.6k points)

1.5.26 Graph Algorithms: GATE2007-5 top

<https://gateoverflow.in/31821>



Go with vertex with indegree 0. Remove the vertex with all edges going from it. Follow that procedure.

We see 3 cannot come at first because indegree is not 0. **So, D is answer here.**

All other options are in Topological order.

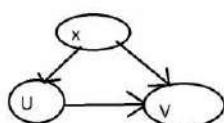
Only 1 and 4 order matters for this question.

17 votes

-- Prashant Singh (59.8k points)

1.5.27 Graph Algorithms: GATE2007-IT-24 top

<https://gateoverflow.in/3457>



I'm gonna disprove all wrong options here:

- $d[u] < d[v]$, Counter Example \implies Well if we directly start DFS on V first, then I call DFS on X which visits U .
- $d[u] < f[v]$, Counter example \implies Same as A
- $f[u] < f[v]$, Counter example \implies Same as A again

So, answer is D.

20 votes

-- Akash Kanase (42.5k points)

1.5.28 Graph Algorithms: GATE2007-IT-3, UGCNET-June2012-III-34 [top](#)

<https://gateoverflow.in/3434>



C. Weight $(u, v) \geq 12$

If weight $(u, v) < 12$, then the min. weight of $(s, v) = \text{weight of } (s, u) + \text{weight of } (u, v) = 53 + (< 12)$ will be less than 65.

27 votes

-- Arjun Suresh (350k points)

1.5.29 Graph Algorithms: GATE2008-19 [top](#)

<https://gateoverflow.in/417>



- A. MNOPQR: If you try to run BFS, after M, you must traverse NQR (In some order). Here, P is traversed before Q, which is wrong.
- B. NQMPOR: This is also not BFS. P is traversed before O.
- C. QMNPOR: Correct.
- D. QMNPOR: Incorrect. Because R needs to be traversed before O.(Because M is ahead of N in queue).

Answer :- C

17 votes

-- Akash Kanase (42.5k points)

1.5.30 Graph Algorithms: GATE2008-45 [top](#)

<https://gateoverflow.in/457>



D. all the vertices. Just simulate the Dijkstra's algorithm on it. Dijkstra's algorithm is not meant for graphs with negative-edge-weight-cycle, but here it does give the correct shortest path.

24 votes

-- Arjun Suresh (350k points)

1.5.31 Graph Algorithms: GATE2008-7 [top](#)

<https://gateoverflow.in/405>



Run DFS to find connected components. Its time complexity is $\Theta(m + n)$, hence **(C)** is the answer.

19 votes

-- Happy Mittal (11.2k points)

1.5.32 Graph Algorithms: GATE2008-IT-47 [top](#)

<https://gateoverflow.in/3357>



Answer: B

1. After f is visited, c or g should be visited next. So, the traversal is incorrect.
4. After c is visited, e or f should be visited next. So, the traversal is incorrect.
- 2 and 3 are correct.

13 votes

-- Rajarshi Sarkar (34.1k points)

1.5.33 Graph Algorithms: GATE2009-13 top

<https://gateoverflow.in/1305>



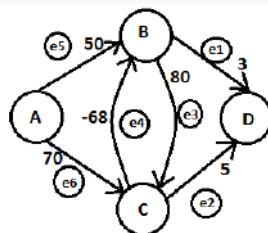
Bellman-ford Algorithm

Single source shortest Path $O(VE)$

Relax every edge once in each iteration

$$\underbrace{E \times (V - 1)}_{\text{at max } (V - 1) \text{ edges can be there}} = E \cdot V - E = O(V \cdot E)$$

at max $(V - 1)$ edges can be there



Verification →

	A	B	C	D
0 null	0 null	∞ null	∞ null	∞ null
i = 0	0 X	50 A	70 A	∞ X
i = 1	0 X	2 C	70 A	53 B
i = 2	0 X	2 C	70 A	5 B
i = 3	0 X	2 C	70 A	5 B

(E * 1) times
Decrease Key
(E * 1) times
Decrease Key

$(V-1)E = V \cdot E$

As we can see that the last step is the verification step. In that step, values remained unchanged. If there was a negative edge weight cycle reachable from source, then at verification step also, those values will be different from the values above.

In case the cycle is not reachable from source then we can see that they will be at ∞ distance(or cost) from the source from the beginning till the last step. As take anything away from the ∞ it will still be infinite.

But it can also be the case that there are some points which are not forming a cycle and are still unreachable from source, those also will be at ∞ distance from the source from the beginning till end.

Hence, we won't be able to make a distinction among the cycle and such vertices. Thus, we say that this algorithm can detect negative edge weight cycles only if they are reachable from the source.

Answer is **option B**

20 votes

-- Amar Vashishth (30.5k points)

1.5.34 Graph Algorithms: GATE2012-40 [top](#)

<https://gateoverflow.in/1765>



Selected Answer

Relaxation at every vertex is as follows

Note the next vertex is taken out here:

	A	B	C	D	E	F	
S	4	3(by s)	∞	7	∞	∞	∞
B	4(by s)		∞	7 $\because (4 + 3 \text{ also } = 7)(S \rightarrow d)$	∞	∞	∞
A			5 ($S \rightarrow B \rightarrow A$)	7	∞	∞	∞
C				7	6 ($S \rightarrow B \rightarrow C$)	∞	∞
E				7 ($S \rightarrow D$)		∞	8 ($S \rightarrow A$ -)
D						12($S \rightarrow B \rightarrow D$)	8
E						12	
T						12	

Now We see for S to T its ($S \rightarrow A \rightarrow C \rightarrow E \rightarrow T$)

which is **Option : D**

29 votes

-- Kalpish Singhal (2.1k points)

1.5.35 Graph Algorithms: GATE2013-19 [top](#)

<https://gateoverflow.in/1441>



Selected Answer

Time complexity of Bellman-Ford algorithm is $\Theta(|V||E|)$ where $|V|$ is number of vertices and $|E|$ is number of edges. If the graph is complete, the value of $|E|$ becomes $\Theta(|V|^2)$. So overall time complexity becomes $\Theta(|V|^3)$. And given here is n vertices. So, the answer ends up to be $\Theta(n^3)$.

35 votes

-- Gate Keeda (19.6k points)

1.5.36 Graph Algorithms: GATE2014-1-11 [top](#)

<https://gateoverflow.in/1771>



[Depth First Search of a graph](#) takes $O(m + n)$ time when the graph is represented using adjacency list. In adjacency matrix representation, graph is represented as an $n * n$ matrix. To do DFS, for every vertex, we traverse the row corresponding to that vertex to find all adjacent vertices (In adjacency list representation we traverse only the adjacent vertices of the vertex). Therefore time complexity becomes $O(n^2)$.

1 like 7 votes

-- Divya Bharti (8.1k points)

1.5.37 Graph Algorithms: GATE2014-1-13 top

<https://gateoverflow.in/1779>



The C option has been copied wrongly
C. Both PSRQ and SPRQ are topological orderings

- i. Apply DFS by choosing P or S as starting vertices
- ii. As the vertex gets a finishing time assign it to the head of a linked list
- iii. The linked list is your required topological ordering

1 like 21 votes

-- Akshay Jindal (457 points)

1.5.38 Graph Algorithms: GATE2014-2-14 top

<https://gateoverflow.in/1969>



BFS always has a starting node. It does not calculate shortest path between every pair but it computes shortest path between W and any other vertex.

1 like 26 votes

-- Digvijay (54.9k points)

1.5.39 Graph Algorithms: GATE2014-3-13 top

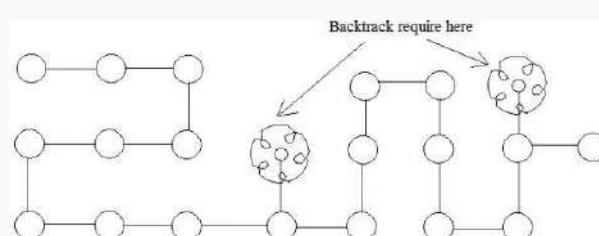
<https://gateoverflow.in/2047>



Total 21 nodes are there. 2 nodes require back track here in this question.

So, max recursion depth is $21 - 2 = 19$

(Do DFS from extreme ends such that max recursion depth will occur i.e. take leftmost top node as initial node for DFS as shown in below image)



Note:- Backtrack means it reduces recursion depth in stack.

29 votes

-- Rajesh Pradhan (22.6k points)

1.5.40 Graph Algorithms: GATE2015-1-45 top

<https://gateoverflow.in/8321>



Selected Answer
2 is the answer.

$d(u) - d(v) = 0$ is possible when both u and v have an edge from t and t is in the shortest path from s to u or v .

$d(u) - d(v) = 1$ is possible when v and t are in the shortest path from s to u and both t and v are siblings- same distance from s to both t and v causing $t - u$ edge to be in BFS tree and not $v - u$.

$d(u) - d(v) = -1$ is possible as explained above by interchanging u and v .

$d(u) - d(v) = 2$ is not possible. This is because on BFS traversal we either visit u first or v . Let's take u first. Now, we put all neighbors of u on queue. Since v is a neighbour and v is not visited before as assumed, $d(v)$ will become $d(u) + 1$. Similarly, for v being visited first.

30 votes

-- Arjun Suresh (350k points)

1.5.41 Graph Algorithms: GATE2016-1-11 top

<https://gateoverflow.in/39669>



Selected Answer
Here, start with
 a and end with
 f .

a — — —
 f

Blank spaces are to be filled with

$b,$
 $c,$
 $d,$
 e such that
 b comes before
 $c,$ and
 d comes before
 $e.$

Number of ways to arrange

$b,$
 $c,$
 $d,$
 e such that
 b comes before
 c and

d comes before
e, will be =
 $4!/(2! * 2!) = 6$

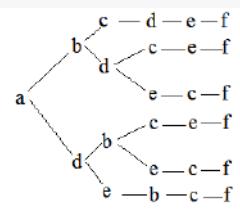
64 votes

-- Abhilash Panicker (9.5k points)

In topological sorting all nodes are like tasks and edges show the dependency among the tasks.

Node *i* to *j* an edge is there means task *i* must complete before task *j*. (in the mean time some other task may get complete after task *i* and before task *j*..but task *i* and *j* sequence need to be maintained)

Here in following 6 ways all the 6 tasks can get completed.



36 votes

-- Rajesh Pradhan (22.6k points)

1.5.42 Graph Algorithms: GATE2016-2-11 top

<https://gateoverflow.in/39563>



No of nodes at level 0(root) of tree $\Rightarrow 1$

No of nodes at level 1 of tree $\Rightarrow 2$

No of nodes at level 2 of tree $\Rightarrow 4$

No of nodes at level 3 of tree $\Rightarrow 8$

No of nodes at level 4 of tree $\Rightarrow 16$

Last node in level 4th is the node we are looking for $\Rightarrow 1 + 2 + 4 + 8 + 16 \Rightarrow 31$

28 votes

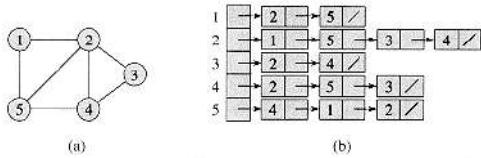
-- Akash Kanase (42.5k points)

1.5.43 Graph Algorithms: GATE2016-2-41 top

<https://gateoverflow.in/39620>



Applying BFS on undirected graph gives you twin pointer. Visit every vertex level-wise. For every vertex, fill adjacent vertex in the adjacency list. BFS takes $\Theta(n + m)$ time.



Take extra field for storing number of linked lists for particular vertex. Take extra $m + n$ time(m vertex and n edges).

So, option **B** is the answer.

 29 votes

-- Prashant Singh (59.8k points)

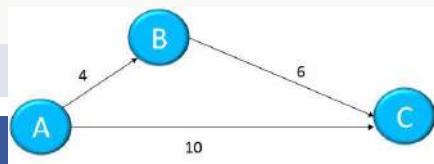
1.5.44 Graph Algorithms: GATE2017-1-26 top

<https://gateoverflow.in/118306>



Answer is **A.**

- MST is not unique only when edges are not distinct. Here the edges are distinct. Be careful for the keyword DISTINCT.
 - Shortest Path can be different even if the edges are distinct. Example is shown below. Shortest path from A to C is not unique here.



 24 votes

-- Ahwan Mishra (10.5k points)

1.5.45 Graph Algorithms: GATE2017-2-15

<https://gateoverflow.in/118196>



In BFS, starting from a node, we traverse all node adjacent to it at first then repeat same for next nodes.

**Here you can see that only
(D) is following BFS sequence properly.**

- As per BFS, if we start from M then RQN (immediate neighbors of M) have to come after it in any order but in A here, O comes in between. So, it is not BFS.
 - As per BFS, if we start from N then QMO has to come after it in any order but in B here, P comes. So, it is not BFS.
 - As per BFS, if we start from Q then $MNOP$ has to come after it in any order but in C here, R comes. So, it is not BFS.

But D is following the sequences.

So, **D** is the correct answer.

 12 votes

-- **Aboveallplayer** (18.4k points)

1.5.46 Graph Algorithms: Gate2000-2.19 top

<https://gateoverflow.in/4208>

Selected Answer

Let this be the DFS order of the tree, then,

$u = D$ and $v = F$

So, we conclude that

1. It is not necessary that there is an edge between them.
2. If there is no edge then u must be leaf i.e. D is leaf here.
3. It is not always possible that u and v have same parent. But they have same ancestor.

34 votes -- Prashant Singh (59.8k points)

1.5.47 Graph Algorithms: TIFR2013-B-5 top

<https://gateoverflow.in/25666>

Selected Answer

I think arbitrary large weights means having positive weight cycle.

So, Bellman Ford algorithm can be used.

$O(VE)$

Changing sign of weights of edges.

5 votes -- papesh pathare (599 points)

1.5.48 Graph Algorithms: TIFR2014-B-3 top

<https://gateoverflow.in/27137>

Selected Answer

Since they said that whenever there is a choice we will have to select the node which is alphabetically earlier, therefore we choose the starting node as A .

The tree then becomes $A - B - E - C$. Therefore number of tree edges is 3, that is, ($T = 3$)

Now, there is one cycle $B - E - C$, so, we will get a back edge from C to B while performing DFS. Hence $B = 1$.

Now, D becomes disconnected node and it can only contribute in forming cross edge. There are 2 cross edges $D - A$, $D - B$. Therefore $C = 2$.

Answer is Option D.

Correct me if am going wrong.

19 votes

-- Riya Roy(Arayana) (7.2k points)

1.6

Graph Connectivity(1) [top](#)1.6.1 Graph Connectivity: GATE2018-43 [top](#)<https://gateoverflow.in/204117>

Let G be a graph with $100!$ vertices!, with each vertex labelled by a distinct permutation od the numbers $1, 2, \dots, 100$. There is an edge between vertices u and v if and only if the label of u can be obtained by swapping two adjacent numbers in the label of v . Let y denote the degree of a vertex in G , and z denote the number of connected components in G . Then $y + 10z = \underline{\hspace{2cm}}$

[gate2018](#) [algorithms](#) [graph-algorithms](#) [graph-connectivity](#) [numerical-answers](#)
[Answer](#)

Answers: Graph Connectivity

1.6.1 Graph Connectivity: GATE2018-43 [top](#)<https://gateoverflow.in/204117>

Selected Answer

Answer: 109**Explanation:**

We have to find 2 things here, the degree of every vertex(which will be same for all vertices) and number of connected components.

Instead of 100, let's solve this by taking lesser value, say 4.

With 4! vertices, each vertex is a permutation of $\{1, 2, 3, 4\}$. So, we have vertices like $\{1, 2, 3, 4\}$, $\{1, 3, 2, 4\}$, $\{4, 1, 3, 2\}$, ... etc.

Here $\{1, 2, 3, 4\}$ will be connected with

$\{2, 1, 3, 4\}$

$\{1, 3, 2, 4\}$

$\{1, 2, 4, 3\}$

To get this list, just take 2 adjacent numbers and swap them. eg. $\{1, 2, 3, 4\}$ swap 1 and 2 to get $\{2, 1, 3, 4\}$.

The given 3 are the only permutations we can get by swapping only 2 adjacent numbers from $\{1, 2, 3, 4\}$. So, the degree of vertex $\{1, 2, 3, 4\}$ will be 3. Similarly for any vertex it's degree will be 3.

Here we got "3" because we can chose any 3 pairs of adjacenet numbers. So, with n , we have $n - 1$ adjacent pairs to swap. So, degree will be $n-1$.

In our question, degree will be $100 - 1 = 99$

Now let's see how many connected components we have.

It will be 1. *Why?*

If one can reach from one vertex to **any** other vertex, then that means that the graph is

connected.

Now if we start with a vertex say $\{1, 2, 3, 4\}$ we can reach to other vertex, say $\{4, 3, 2, 1\}$ by the following path:

$\{1234\} \rightarrow \{1243\} \rightarrow \{1423\} \rightarrow \{4123\} \rightarrow \{4132\} \rightarrow \{4312\} \rightarrow \{4321\}$

Just take two adjacent numbers and swap them. With this operation you can create any permutation, from any given initial permutation.

This way you can show that from any given vertex we can reach any other vertex. This shows that the graph is connected and the number of connected components is **1**.

y = 99 and z = 1

y + 10z = 99 + 10*1 = 109

26 votes

-- Rishabh Gupta (13.9k points)

1.7

Greedy Algorithm(8) top

1.7.1 Greedy Algorithm: CMI2015-B-04 top

<https://gateoverflow.in/47063>

You are given n positive integers, $d_1, d_2 \dots d_n$, each greater than 0. Design a greedy algorithm to test whether these integers correspond to the degrees of some n -vertex simple undirected graph $G = (V, E)$. [A simple graph has no self-loops and at most one edge between any pair of vertices].

cmi2015 descriptive algorithms greedy-algorithm

Answer

1.7.2 Greedy Algorithm: GATE1999-2.20 top

<https://gateoverflow.in/466>

The minimum number of record movements required to merge five files A (with 10 records), B (with 20 records), C (with 15 records), D (with 5 records) and E (with 25 records) is:

- A. 165
- B. 90
- C. 75
- D. 65

gate1999 algorithms normal greedy-algorithm

Answer

1.7.3 Greedy Algorithm: GATE2003-69 top

<https://gateoverflow.in/956>

The following are the starting and ending times of activities A, B, C, D, E, F, G and H respectively in chronological order: " $a_s b_s c_s a_e d_s c_e e_s f_s b_e d_e g_s e_e f_e h_s g_e h_e$ ". Here, x_s denotes the starting time and x_e denotes the ending time of activity X. We need to schedule the activities in a set of rooms available to us. An activity can be scheduled in a room only if the room is reserved for the activity for its entire duration. What is the minimum number of rooms required?

- A. 3
- B. 4
- C. 5
- D. 6

gate2003 algorithms normal greedy-algorithm

Answer**1.7.4 Greedy Algorithm: GATE2005-84a** [top](#)<https://gateoverflow.in/1406>

We are given 9 tasks T_1, T_2, \dots, T_9 . The execution of each task requires one unit of time. We can execute one task at a time. Each task T_i has a profit P_i and a deadline d_i . Profit P_i is earned if the task is completed before the end of the d_i^{th} unit of time.

Task	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Profit	15	20	30	18	18	10	23	16	25
Deadline	7	2	5	3	4	5	2	7	3

Are all tasks completed in the schedule that gives maximum profit?

- A. All tasks are completed
- B. T_1 and T_6 are left out
- C. T_1 and T_8 are left out
- D. T_4 and T_6 are left out

[gate2005](#) [algorithms](#) [greedy-algorithm](#) [process-schedule](#) [normal](#)
Answer**1.7.5 Greedy Algorithm: GATE2005-84b** [top](#)<https://gateoverflow.in/82514>

We are given 9 tasks T_1, T_2, \dots, T_9 . The execution of each task requires one unit of time. We can execute one task at a time. Each task T_i has a profit P_i and a deadline d_i . Profit P_i is earned if the task is completed before the end of the d_i^{th} unit of time.

Task	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Profit	15	20	30	18	18	10	23	16	25
Deadline	7	2	5	3	4	5	2	7	3

What is the maximum profit earned?

- A. 147
- B. 165
- C. 167
- D. 175

[gate2005](#) [algorithms](#) [greedy-algorithm](#) [process-schedule](#) [normal](#)
Answer**1.7.6 Greedy Algorithm: GATE2006-IT-48** [top](#)<https://gateoverflow.in/3591>

The characters a to h have the set of frequencies based on the first 8 Fibonacci numbers as follows

$a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21$

A Huffman code is used to represent the characters. What is the sequence of characters corresponding to the following code?

110111100111010

- A. *fdheg*
 B. *ecgdf*
 C. *dchfg*
 D. *fehdg*

gate2006-it algorithms greedy-algorithm normal

Answer

1.7.7 Greedy Algorithm: GATE2007-76 [top](#)

<https://gateoverflow.in/1271>

Suppose the letters a, b, c, d, e, f have probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32}$, respectively.

Which of the following is the Huffman code for the letter a, b, c, d, e, f ?

- A. 0, 10, 110, 1110, 11110, 11111
 B. 11, 10, 011, 010, 001, 000
 C. 11, 10, 01, 001, 0001, 0000
 D. 110, 100, 010, 000, 001, 111

gate2007 algorithms greedy-algorithm normal

Answer

1.7.8 Greedy Algorithm: GATE2018-48 [top](#)

<https://gateoverflow.in/204123>

Consider the weights and values of items listed below. Note that there is only one unit of each item.

Item number	Weight (in Kgs)	Value (in rupees)
1	10	60
2	7	28
3	4	20
4	2	24

The task is to pick a subset of these items such that their total weight is no more than 11 Kgs and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by V_{opt} . A greedy algorithm sorts the items by their value-to-weight ratios in descending order and packs them greedily, starting from the first item in the ordered list. The total value of items picked by the greedy algorithm is denoted by V_{greedy} .

The value of $V_{opt} - V_{greedy}$ is _____

gate2018 algorithms greedy-algorithm numerical-answers

Answer

Answers: Greedy Algorithm

1.7.1 Greedy Algorithm: CMI2015-B-04 [top](#)

<https://gateoverflow.in/47063>



Selected Answer

1. Sort the degrees in non-increasing order.
2. Pick up the highest degree (let us say it a), remove it from the list of degrees and subtract 1 from next a degrees in list.
3. Repeat Step 2 until :

- If we get all 0 entries in list \Rightarrow Simple Graph exists
- If we get a negative entry or not enough entries to subtract 1 in step 2 \Rightarrow Simple Graph does not exist

Read More : <http://goo.gl/4u3nfh>

Let's take a example : 3, 2, 1, 2

Step 1 : Sort the degree sequence : 3, 2, 2, 1

Step 2: Pick 3, Remove 3 from list and from next 3 elements subtract 1, Result : (1, 1, 0)

Again repeat step 2 : select 1, Remove 1 and from next 1 subtract 1, Result : (0, 0, 0)

Thus, a simple graph exists for the following degree-sequence.

15 votes

-- Manish Joshi (27.9k points)

1.7.2 Greedy Algorithm: GATE1999-2.20 top

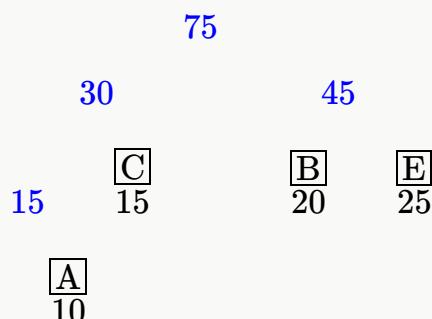
<https://gateoverflow.in/466>



Selected Answer

Arrange files in increasing order of records:

5	10	15	20	25
D	A	C	B	E



No. of movements = $15 + 30 + 45 + 75 = 165$.

23 votes

-- Pooja Palod (31.3k points)

1.7.3 Greedy Algorithm: GATE2003-69 top

<https://gateoverflow.in/956>

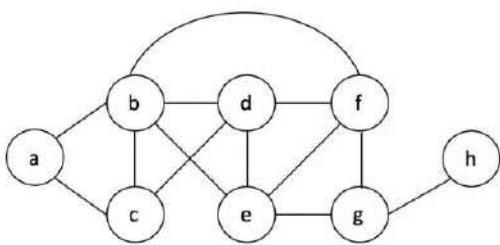


Selected Answer

Solution: B

The problem can be modeled as a graph coloring problem. Construct a graph with one node corresponding to each activity A, B, C, D, E, F, G and H. Connect the activities that occur

between the start and end time of an activity. Now, the chromatic number of the graph is the number of rooms required.



28 votes

-- Gowthaman Arumugam (1.4k points)

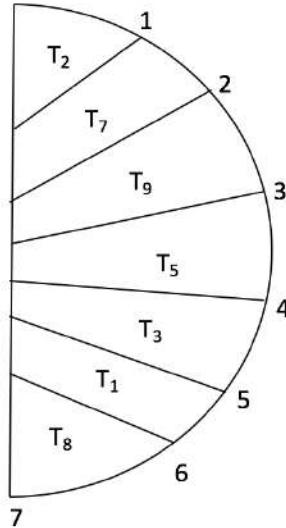
1.7.4 Greedy Algorithm: GATE2005-84a top

<https://gateoverflow.in/1406>



Selected Answer

Task	Profit	Deadline	
T_3	30	5	✓
T_9	25	3	✓
T_7	23	2	✓
T_2	20	2	✓
T_5	18	4	✓
T_4	18	3	✗
T_8	16	7	✓
T_1	15	7	✓
T_6	10	2	✗



Step -1 Sort the tasks in decreasing order of profit and if any conflict arises between two or more tasks, resolve them by sorting them on basis of having greater deadline first (Because we have more time to complete the task with greater deadline and same profit).

Step 2- Since Maximum deadline given is 7, so we consider we have 7 time slots ranging from 0 – 7 where a task T_i having deadline say 2 can be filled in slots either 0 – 1 or 1 – 2 and not beyond 2 because this task has deadline of 2 time units, so this task has to be completed by atmost time $T = 2$.

Now according to question, since Each task completes in Unit time, so a single tasks takes only one slot as shown.

Now Take the first task in the list i.e. T_3 which has a deadline of 5, so it can be completed in maximum 5 time units, so place it in slot 4 – 5 which is the maximum deadline by which this task can be completed.

Task T_9 with deadline 3 is similarly placed in slot 2 – 3.

Task T_7 with deadline 2 is placed in slot 1 – 2.

Now for task T_2 having deadline 2 can be placed in either 0 – 1 or 1 – 2 (Occupied by T_7). So T_2 will occupy slot 0 – 1.

Task T_5 with deadline 4 is placed in slot 3 – 4.

Now comes task T_4 which has deadline 3 can be put in slots 0 – 1 or 1 – 2 or 2 – 3 and not beyond that. Unfortunately, all such slots are occupied so
 T_4 will be left out.

Task T_8 with deadline 7 goes in slot 6 – 7.

Task T_1 with deadline 7 can be placed in slot 5 – 6.

Now all time slots are **full**.

So, Task
 T_6 will be left out.

So, option (d) is the answer.

15 votes

-- Ayush Upadhyaya (9k points)

1.7.5 Greedy Algorithm: GATE2005-84b top

<https://gateoverflow.in/82514>



The most important statement in question is

each task requires one unit of time

This shows that we can greedily choose the better task and that should give us the optimal solution. The best task would be the one with maximum profit. Thus we can sort the tasks based on deadline and then profit as follows:

Task	T_7	T_2	T_9	T_4	T_5	T_3	T_6	T_8	T_1
Deadline	2	2	3	3	4	5	5	7	7

0 -- T_7 -- 1 -- T_2 -- 2 -- T_9 -- 3 -- T_5 -- 4 -- T_3 -- 5 -- T_8 -- 6 -- T_1 -- 7

so we know that T_4 and T_6 are left

so profit will not include T_4 and T_6 = $15 + 20 + 30 + 18 + 16 + 23 + 25 = 147$

A is answer

20 votes

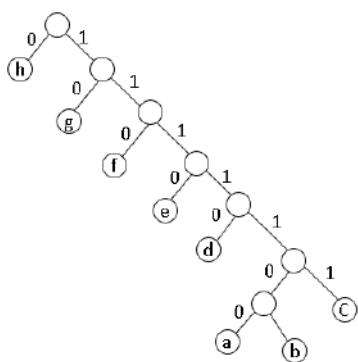
-- Prashant Singh (59.8k points)

1.7.6 Greedy Algorithm: GATE2006-IT-48 top

<https://gateoverflow.in/3591>



Answer is **A**. Huffman's tree is as follows. The two least frequent characters are taken as the children of a newly made node and the frequency of the newly made node is made equal to the sum of those two child nodes. Then the same procedure is repeated till all nodes are finished.



$110111100111010 = 110111100111010 = fdheg$

1 31 votes

-- Arjun Suresh (350k points)

1.7.7 Greedy Algorithm: GATE2007-76 top

<https://gateoverflow.in/1271>

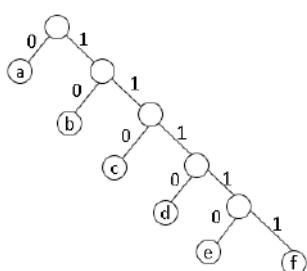


Selected Answer

Based on the probabilities, we can say the probable frequency of the letters will be

16, 8, 4, 2, 1, 1

Now, the Huffman tree can be constructed as follows:



So, A is the answer for 76.

<https://www.siggraph.org/education/materials/HyperGraph/video/mpeg/mpegfaq/huffmanTutorial.htm>

1 22 votes

-- Arjun Suresh (350k points)

1.7.8 Greedy Algorithm: GATE2018-48 top

<https://gateoverflow.in/204123>



Selected Answer

Voptimal is clearly 60. You can go for brute force or by normal intuition you can get it.

Now solving for **Vgreedy**.

Item name	Weight	Value	Value/Weight
1	10	60	6
2	7	28	4
3	4	20	5
4	2	24	12

Sort them in descending order of Value/Weight as per the question.

Item name	Weight	Value	Value/Weight
4	2	24	12
1	10	60	6
3	4	20	5
2	7	28	4

Now start picking items.(Note: You cannot take a fraction of the given weight as per the question). Max weight size is given as 11(Inclusive).

Item 4 is picked. Weight remaining = $11 - 2 = 9\text{kg}$.

Item 1 cannot be picked as $10\text{kg} > 9\text{kg}$.

Item 3 can be picked as $4\text{kg} < 9\text{kg}$. Weight Remaining = $9 - 4 = 5\text{kg}$

Item 2 cannot be picked as $7\text{kg} > 5\text{kg}$.

So, item 4 and Item 3 are picked. Their values are 24 and 20 respectively.

So, Vgreedy = $24 + 20 = 44$

Voptimal - Vgreedy= $60 - 44 = 16$ answer.

12 votes

-- Ruturaj Mohanty (337 points)

1.8

Huffman Code(2) top

1.8.1 Huffman Code: GATE2007-77 top

<https://gateoverflow.in/43513>

Suppose the letters a, b, c, d, e, f have probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32}$, respectively.

What is the average length of the Huffman code for the letters a, b, c, d, e, f ?

- A. 3
- B. 2.1875
- C. 2.25
- D. 1.9375

[gate2007](#) [algorithms](#) [greedy-algorithm](#) [normal](#) [huffman-code](#)

Answer

1.8.2 Huffman Code: GATE2017-2-50 top

<https://gateoverflow.in/118395>

A message is made up entirely of characters from the set $X = \{P, Q, R, S, T\}$. The table of probabilities for each of the characters is shown below:

Character	Probability
P	0.22
Q	0.34
R	0.17
S	0.19
T	0.08
Total	1.00

If a message of 100 characters over X is encoded using Huffman coding, then the expected length of the encoded message in bits is _____.

gate2017-2 huffman-code numerical-answers algorithms

Answer

Answers: Huffman Code

1.8.1 Huffman Code: GATE2007-77 top

<https://gateoverflow.in/43513>

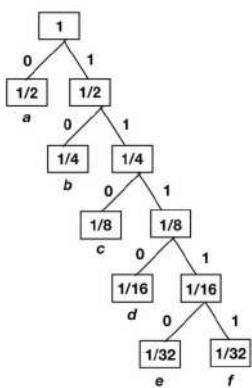


Selected Answer

Answer should be D.

Letter Probability

a	1/2
b	1/4
c	1/8
d	1/16
e	1/32
f	1/32



Prefix Code	Code Length
a=0	1
b=10	2
c=110	3
d=1110	4
e=11110	5

f=11111

5

Avg length

$$= \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 + \frac{1}{32} \times 5 + \frac{1}{32} \times 5 = \frac{16+16+12+8+5+5}{32} = 1.9375$$

1 23 votes

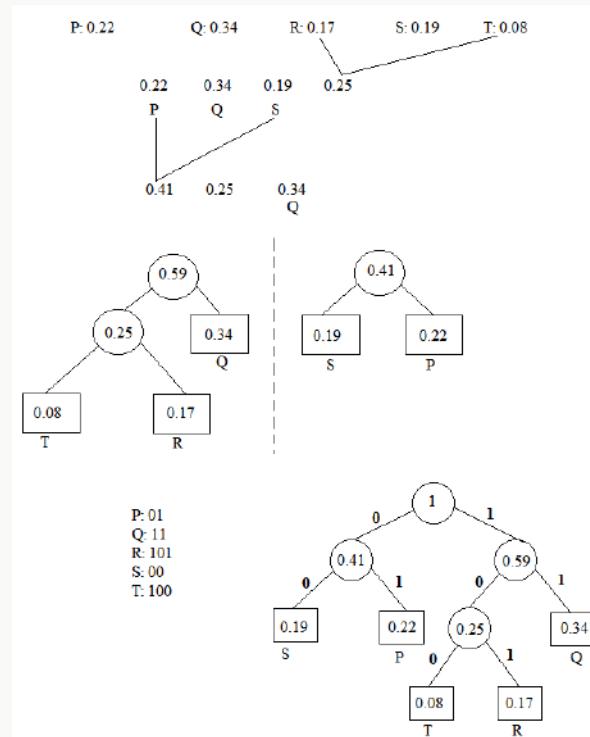
-- sonam vyas (14.8k points)

1.8.2 Huffman Code: GATE2017-2-50 top

<https://gateoverflow.in/118395>

Selected Answer

$$X = \{P, Q, R, S, T\}$$



∴ Expected length of encoded message

$$= (22 \times 2) + (34 \times 2) + (17 \times 3) + (19 \times 2) + (8 \times 3) \text{ bits} = 44 + 68 + 51 + 38 + 24 \text{ bits} = 205 \text{ bits}$$

1 6 votes

-- Akash Dinkar (23.8k points)

1.9

Identify Function(37) top

1.9.1 Identify Function: CMI2010-B-07b top

<https://gateoverflow.in/47075>

A finite sequence of bits is represented as a list with values from the set $\{0, 1\}$. For example, $[0, 1, 0], [1, 0, 1, 1], \dots$.

$[]$ denotes the empty list, and $[b]$ is the list consisting of one bit b . The function $length(l)$ returns the length (number of bits) in the list l . For a nonempty list l , $head(l)$ returns the first element of l ,

and $\text{tail}(l)$ returns the list obtained by removing the first element from l . The operator $++$ denotes list concatenation.

For example:

- $\text{head}([0, 1, 0]) = 0, \text{tail}([0, 1, 0]) = [1, 0]$
- $\text{head}([1]) = 1, \text{tail}([1]) = [],$ and
- $[0, 1, 0] ++ [1] = [0, 1, 0, 1].$

Consider the following functions:

op takes as input two bits and returns a bit.

```
op(a, b)
if (a = b) return(0)
else return(1)
endif
```

$mystery1$ takes as input two lists and returns a list.

```
mystery1(s, t)
if (length(s) != length(t)) then return(t)
else if (length(s) = 0) then return(s)
else return([op(head(s), head(t))] ++ mystery1(tail(s), tail(t)))
endif
endif
```

$mystery2$ takes as input two lists and outputs a list.

```
mystery2(s, t)
if (length(t) = 0) then return(s)
else return(mystery1(mystery2(s, tail(t)), mystery2(s, tail(t))))
```

- Suppose $s = t = 110100100$. What are the first two bits of $mystery2(s, t)$?

descriptive | cmi2010 | algorithms | identify-function

Answer

1.9.2 Identify Function: CMI2013-A-09 top

<https://gateoverflow.in/46599>

The below question is based on the following program.

```
procedure mystery (A : array [1..100] of int)
    int i, j, position, tmp;
begin
    for j := 1 to 100 do
        position := j;
        for i := j to 100 do
            if (A[i] > A[position]) then
                position := i;
            endfor
            tmp := A[j];
            A[j] := A[position];
            A[position] := tmp;
        endfor
    end
```

When the procedure terminates, the array A has been:

- Reversed
- Sorted in descending order
- Left unaltered
- Sorted in ascending order

cmi2013 | algorithms | identify-function

Answer**1.9.3 Identify Function: GATE1990-11b** [top](#)<https://gateoverflow.in/85987>

The following program computes values of a mathematical function $f(x)$. Determine the form of $f(x)$.

```
main ()
{
    int m, n; float x, y, t;
    scanf ("%f%d", &x, &n);
    t = 1; y = 0; m = 1;
    do
    {
        t *= (-x/m);
        y += t;
    } while (m++ < n);
    printf ("The value of y is %f", y);
}
```

[gate1990](#) [descriptive](#) [algorithms](#) [identify-function](#)**Answer****1.9.4 Identify Function: GATE1991-03-viii** [top](#)<https://gateoverflow.in/523>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Consider the following Pascal function:

```
Function X(M:integer):integer;
Var i:integer;
Begin
    i := 0;
    while i*i < M
    do i:= i+1
    X := i
end
```

The function call $X(N)$, if N is positive, will return

- A. $\lfloor \sqrt{N} \rfloor$
- B. $\lfloor \sqrt{N} \rfloor + 1$
- C. $\lceil \sqrt{N} \rceil$
- D. $\lceil \sqrt{N} \rceil + 1$
- E. None of the above

[gate1991](#) [algorithms](#) [easy](#) [identify-function](#)**Answer****1.9.5 Identify Function: GATE1993-7.4** [top](#)<https://gateoverflow.in/2292>

What does the following code do?

```
var a, b: integer;
begin
    a:=a+b;
```

```
b:=a-b;
a:=a-b;
end;
```

- A. exchanges a and b
- B. doubles a and stores in b
- C. doubles b and stores in a
- D. leaves a and b unchanged
- E. none of the above

gate1993 algorithms identify-function easy

Answer

1.9.6 Identify Function: GATE1994-6 [top](#)

<https://gateoverflow.in/2502>

What function of x, n is computed by this program?

```
Function what(x, n:integer): integer;
Var
  value : integer
begin
  value := 1
  if n > 0 then
    begin
      if n mod 2 =1 then
        value := value * x;
      value := value * what(x*x, n div 2);
    end;
  what := value;
end;
```

gate1994 algorithms identify-function normal

Answer

1.9.7 Identify Function: GATE1995-1.4 [top](#)

<https://gateoverflow.in/2591>

In the following Pascal program segment, what is the value of X after the execution of the program segment?

```
X := -10; Y := 20;
If X > Y then if X < 0 then X := abs(X) else X := 2*X;
```

- A. 10
- B. -20
- C. -10
- D. None

gate1995 algorithms identify-function easy

Answer

1.9.8 Identify Function: GATE1995-2.3 [top](#)

<https://gateoverflow.in/2615>

Assume that X and Y are non-zero positive integers. What does the following Pascal program segment do?

```
while X <> Y do
```

```

if  X > Y then
  X := X - Y
else
  Y := Y - X;
write (X);

```

- A. Computes the LCM of two numbers
- B. Divides the larger number by the smaller number
- C. Computes the GCD of two numbers
- D. None of the above

gate1995 algorithms identify-function normal

Answer

1.9.9 Identify Function: GATE1995-4 [top](#)

<https://gateoverflow.in/2640>

- a. Consider the following Pascal function where A and B are non-zero positive integers. What is the value of $GET(3, 2)$?

```

function GET(A,B:integer): integer;
begin
  if B=0 then
    GET:= 1
  else if A < B then
    GET:= 0
  else
    GET:= GET(A-1, B) + GET(A-1, B-1)
end;

```

- b. The Pascal procedure given for computing the transpose of an $N \times N$, ($N > 1$) matrix A of integers has an error. Find the error and correct it. Assume that the following declaration are made in the main program

```

const
  MAXSIZE=20;
type
  INTARR=array [1..MAXSIZE,1..MAXSIZE] of integer;
Procedure TRANSPOSE (var A: INTARR; N : integer);
var
  I, J, TMP: integer;
begin
  for I:=1 to N - 1 do
    for J:=1 to N do
      begin
        TMP:= A[I, J];
        A[I, J]:= A[J, I];
        A[J, I]:= TMP
      end
end;

```

gate1995 algorithms identify-function normal

Answer

1.9.10 Identify Function: GATE1998-2.12 [top](#)

<https://gateoverflow.in/1684>

What value would the following function return for the input $x = 95$?

```

Function fun (x:integer):integer;
Begin
    If x > 100 then fun = x - 10
    Else fun = fun(fun (x+11))
End;

```

- A. 89
- B. 90
- C. 91
- D. 92

gate1998 algorithms recursion identify-function normal

Answer

1.9.11 Identify Function: GATE1999-2.24 [top](#)

<https://gateoverflow.in/1501>

Consider the following C function definition

```

int Trial (int a, int b, int c)
{
    if ((a>=b) && (c<b)) return b;
    else if (a>=b) return Trial(a, c, b);
    else return Trial(b, a, c);
}

```

The functional Trial:

- A. Finds the maximum of a , b , and c
- B. Finds the minimum of a , b , and c
- C. Finds the middle number of a , b , c
- D. None of the above

gate1998 algorithms recursion identify-function normal

Answer

1.9.12 Identify Function: GATE2000-2.15 [top](#)

<https://gateoverflow.in/662>

Suppose you are given an array $s[1....n]$ and a procedure reverse (s, i, j) which reverses the order of elements in s between positions i and j (both inclusive). What does the following sequence do, where $1 \leq k \leq n$:

```

reverse (s, 1, k);
reverse (s, k+1, n);
reverse (s, 1, n);

```

- A. Rotates s left by k positions
- B. Leaves s unchanged
- C. Reverses all elements of s
- D. None of the above

gate2000 algorithms normal identify-function

Answer

1.9.13 Identify Function: GATE2003-1 [top](#)

<https://gateoverflow.in/892>

Consider the following C function.

For large values of y , the return value of the function f best approximates

```
float f, (float x, int y) {
    float p, s; int i;
    for (s=1, p=1, i=1; i<y; i++) {
        p *= x/i;
        s += p;
    }
    return s;
}
```

- A. x^y
- B. e^x
- C. $\ln(1 + x)$
- D. x^x

[gate2003](#) [algorithms](#) [identify-function](#) [normal](#)

Answer

1.9.14 Identify Function: GATE2003-88 [top](#)

<https://gateoverflow.in/971>

In the following C program fragment, j , k , n and TwoLog_n are integer variables, and A is an array of integers. The variable n is initialized to an integer ≥ 3 , and TwoLog_n is initialized to the value of $2^{\lfloor \log_2(n) \rfloor}$

```
for (k = 3; k <= n; k++)
    A[k] = 0;
for (k = 2; k <= TwoLog_n; k++)
    for (j = k+1; j <= n; j++)
        A[j] = A[j] || (j%k);
for (j = 3; j <= n; j++)
    if (!A[j]) printf("%d", j);
```

The set of numbers printed by this program fragment is

- A. $\{m \mid m \leq n, (\exists i) [m = i!]\}$
- B. $\{m \mid m \leq n, (\exists i) [m = i^2]\}$
- C. $\{m \mid m \leq n, m \text{ is prime}\}$
- D. $\{ \}$

[gate2003](#) [algorithms](#) [identify-function](#) [normal](#)

Answer

1.9.15 Identify Function: GATE2004-41 [top](#)

<https://gateoverflow.in/1038>

Consider the following C program

```
main()
{
    int x, y, m, n;
    scanf("%d %d", &x, &y);
    /* Assume x>0 and y>0*/
    m = x; n = y;
    while(m != n)
```

```

    {
        if  (m > n)
            m = m-n;
        else
            n = n-m;
    }
    printf("%d", n);
}

```

The program computes

- A. $x + y$ using repeated subtraction
- B. $x \bmod y$ using repeated subtraction
- C. the greatest common divisor of x and y
- D. the least common multiple of x and y

[gate2004](#) [algorithms](#) [normal](#) [identify-function](#)

Answer

1.9.16 Identify Function: GATE2004-42 [top](#)

<https://gateoverflow.in/1039>

What does the following algorithm approximate? (Assume $m > 1, \epsilon > 0$).

```

x = m;
y = 1;
While (x-y > ε)
{
    x = (x+y)/2;
    y = m/x;
}
print(x);

```

- A. $\log m$
- B. m^2
- C. $m^{\frac{1}{2}}$
- D. $m^{\frac{1}{3}}$

[gate2004](#) [algorithms](#) [identify-function](#) [normal](#)

Answer

1.9.17 Identify Function: GATE2005-31 [top](#)

<https://gateoverflow.in/1367>

Consider the following C-program:

```

void foo (int n, int sum) {
    int k = 0, j = 0;
    if (n == 0) return;
    k = n % 10; j = n/10;
    sum = sum + k;
    foo (j, sum);
    printf ("%d", k);
}

int main() {
    int a = 2048, sum = 0;
    foo(a, sum);
}

```

```

    printf("%d\n", sum);
}

```

What does the above program print?

- A. 8, 4, 0, 2, 14
- B. 8, 4, 0, 2, 0
- C. 2, 0, 4, 8, 14
- D. 2, 0, 4, 8, 0

[gate2005](#) [algorithms](#) [identify-function](#) [recursion](#) [normal](#)

Answer

1.9.18 Identify Function: GATE2005-IT-57 [top](#)

<https://gateoverflow.in/3818>

What is the output printed by the following program?

```

#include <stdio.h>

int f(int n, int k) {
    if (n == 0) return 0;
    else if (n % 2) return f(n/2, 2*k) + k;
    else return f(n/2, 2*k) - k;
}

int main () {
    printf("%d", f(20, 1));
    return 0;
}

```

- A. 5
- B. 8
- C. 9
- D. 20

[gate2005-it](#) [algorithms](#) [identify-function](#) [normal](#)

Answer

1.9.19 Identify Function: GATE2006-50 [top](#)

<https://gateoverflow.in/1828>

A set X can be represented by an array $x[n]$ as follows:

$$x[i] = \begin{cases} 1 & \text{if } i \in X \\ 0 & \text{otherwise} \end{cases}$$

Consider the following algorithm in which x , y , and z are Boolean arrays of size n :

```

algorithm zzz(x[], y[], z[]) {
    int i;
    for(i=0; i<n; ++i)
        z[i] = (x[i] ^ ~y[i]) || (~x[i] ^ y[i]);
}

```

The set Z computed by the algorithm is:

- A. $(X \cup Y)$
- B. $(X \cap Y)$

- C. $(X - Y) \cap (Y - X)$
 D. $(X - Y) \cup (Y - X)$

gate2006 | algorithms | identify-function | normal

Answer

1.9.20 Identify Function: GATE2006-53 top

<https://gateoverflow.in/1831>

Consider the following C-function in which $a[n]$ and $b[m]$ are two sorted integer arrays and $c[n + m]$ be another integer array,

```
void xyz(int a[], int b[], int c[]) {
    int i, j, k;
    i=j=k=0;
    while ((i < n) && (j < m))
        if (a[i] < b[j]) c[k++] = a[i++];
        else c[k++] = b[j++];
}
```

Which of the following condition(s) hold(s) after the termination of the while loop?

- i. $j < m, k = n + j - 1$ and $a[n - 1] < b[j]$ if $i = n$
 ii. $i < n, k = m + i - 1$ and $b[m - 1] \leq a[i]$ if $j = m$

- A. only (i)
 B. only (ii)
 C. either (i) or (ii) but not both
 D. neither (i) nor (ii)

gate2006 | algorithms | identify-function | normal

Answer

1.9.21 Identify Function: GATE2006-IT-52 top

<https://gateoverflow.in/3595>

The following function computes the value of $\binom{m}{n}$ correctly for all legal values m and n ($m \geq 1, n \geq 0$ and $m > n$)

```
int func(int m, int n)
{
    if (E) return 1;
    else return(func(m - 1, n) + func(m - 1, n - 1));
}
```

In the above function, which of the following is the correct expression for E?

- A. $(n == 0) \mid\mid (m == 1)$
 B. $(n == 0) \&\& (m == 1)$
 C. $(n == 0) \mid\mid (m == n)$
 D. $(n == 0) \&\& (m == n)$

gate2006-it | algorithms | identify-function | normal

Answer

1.9.22 Identify Function: GATE2008-IT-82 top

<https://gateoverflow.in/3406>

Consider the code fragment written in C below :

```
void f (int n)
{
    if (n <=1) {
        printf ("%d", n);
    }
    else {
        f (n/2);
        printf ("%d", n%2);
    }
}
```

What does $f(173)$ print?

- A. 010110101
- B. 010101101
- C. 10110101
- D. 10101101

gate2008-it algorithms recursion identify-function normal

Answer

1.9.23 Identify Function: GATE2008-IT-83 [top](#)

<https://gateoverflow.in/3407>

Consider the code fragment written in C below :

```
void f (int n)
{
    if (n <= 1) {
        printf ("%d", n);
    }
    else {
        f (n/2);
        printf ("%d", n%2);
    }
}
```

Which of the following implementations will produce the same output for $f(173)$ as the above code?

P1

```
void f (int n)
{
    if (n/2) {
        f(n/2);
    }
    printf ("%d", n%2);
}
```

P2

```
void f (int n)
{
    if (n <=1) {
        printf ("%d", n);
    }
    else {
        printf ("%d", n%2);
        f (n/2);
    }
}
```

- A. Both $P1$ and $P2$
- B. $P2$ only
- C. $P1$ only
- D. Neither $P1$ nor $P2$

[gate2008-it](#)
[algorithms](#)
[recursion](#)
[identify-function](#)
[normal](#)
Answer

1.9.24 Identify Function: GATE2009-18 [top](#)

<https://gateoverflow.in/1310>

Consider the program below:

```
#include <stdio.h>
int fun(int n, int *f_p) {
    int t, f;
    if (n <= 1) {
        *f_p = 1;
        return 1;
    }
    t = fun(n-1, f_p);
    f = t + *f_p;
    *f_p = t;
    return f;
}

int main() {
    int x = 15;
    printf("%d\n", fun(5, &x));
    return 0;
}
```

The value printed is:

- A. 6
- B. 8
- C. 14
- D. 15

[gate2009](#)
[algorithms](#)
[recursion](#)
[identify-function](#)
[normal](#)
Answer

1.9.25 Identify Function: GATE2010-35 [top](#)

<https://gateoverflow.in/2336>

What is the value printed by the following C program?

```
#include<stdio.h>

int f(int *a, int n)
{
    if (n <= 0) return 0;
    else if (*a % 2 == 0) return *a+f(a+1, n-1);
    else return *a - f(a+1, n-1);
}

int main()
{
    int a[] = {12, 7, 13, 4, 11, 6};
    printf("%d", f(a, 6));
    return 0;
}
```

- A. -9
- B. 5
- C. 15
- D. 19

gate2010 algorithms recursion identify-function normal

Answer

1.9.26 Identify Function: GATE2011-48 [top](#)

<https://gateoverflow.in/2154>

Consider the following recursive C function that takes two arguments.

```
unsigned int foo(unsigned int n, unsigned int r) {
    if (n>0) return ((n%r) + foo(n/r, r));
    else return 0;
}
```

What is the return value of the function foo when it is called as foo(345, 10)?

- A. 345
- B. 12
- C. 5
- D. 3

gate2011 algorithms recursion identify-function normal

Answer

1.9.27 Identify Function: GATE2011-49 [top](#)

<https://gateoverflow.in/43324>

Consider the following recursive C function that takes two arguments.

```
unsigned int foo(unsigned int n, unsigned int r) {
    if (n>0) return ((n%r) + foo(n/r, r));
    else return 0;
}
```

What is the return value of the function foo when it is called as foo(513, 2)?

- A. 9
- B. 8
- C. 5
- D. 2

gate2011 algorithms recursion identify-function normal

Answer

1.9.28 Identify Function: GATE2013-31 [top](#)

<https://gateoverflow.in/1542>

Consider the following function:

```
int unknown(int n) {
    int i, j, k=0;
    for (i=n/2; i<=n; i++)
        for (j=2; j<=n; j=j*2)
            k = k + n/2;
    return (k);
}
```

The return value of the function is

- A. $\Theta(n^2)$

- B. $\Theta(n^2 \log n)$
- C. $\Theta(n^3)$
- D. $\Theta(n^3 \log n)$

gate2013 algorithms identify-function normal

Answer

1.9.29 Identify Function: GATE2014-1-41 [top](#)

<https://gateoverflow.in/1919>

Consider the following C function in which **size** is the number of elements in the array **E**:

```
int MyX(int *E, unsigned int size)
{
    int Y = 0;
    int Z;
    int i, j, k;

    for(i = 0; i < size; i++)
        Y = Y + E[i];

    for(i=0; i < size; i++)
        for(j = i; j < size; j++)
        {
            Z = 0;
            for(k = i; k <= j; k++)
                Z = Z + E[k];
            if(Z > Y)
                Y = Z;
        }
    return Y;
}
```

The value returned by the function **MyX** is the

- A. maximum possible sum of elements in any sub-array of array **E**.
- B. maximum element in any sub-array of array **E**.
- C. sum of the maximum elements in all possible sub-arrays of array **E**.
- D. the sum of all the elements in the array **E**.

gate2014-1 algorithms identify-function normal

Answer

1.9.30 Identify Function: GATE2014-2-10 [top](#)

<https://gateoverflow.in/1964>

Consider the function **func** shown below:

```
int func(int num) {
    int count = 0;
    while (num) {
        count++;
        num>>= 1;
    }
    return (count);
}
```

The value returned by **func(435)** is _____

gate2014-2 algorithms identify-function numerical-answers easy

Answer**1.9.31 Identify Function: GATE2014-3-10** top<https://gateoverflow.in/2044>

Let A be the square matrix of size $n \times n$. Consider the following pseudocode. What is the expected output?

```
C=100;
for i=1 to n do
    for j=1 to n do
    {
        Temp = A[i][j]+C;
        A[i][j] = A[j][i];
        A[j][i] = Temp -C;
    }
for i=1 to n do
    for j=1 to n do
        output (A[i][j]);
```

- A. The matrix A itself
- B. Transpose of the matrix A
- C. Adding 100 to the upper diagonal elements and subtracting 100 from lower diagonal elements of A
- D. None of the above

[gate2014-3](#) [algorithms](#) [identify-function](#) [easy](#)**Answer****1.9.32 Identify Function: GATE2015-1-31** top<https://gateoverflow.in/8263>

Consider the following C function.

```
int fun1 (int n) {
    int i, j, k, p, q = 0;
    for (i = 1; i < n; ++i)
    {
        p = 0;
        for (j = n; j > 1; j = j/2)
            ++p;
        for (k = 1; k < p; k = k * 2)
            ++q;
    }
    return q;
}
```

Which one of the following most closely approximates the return value of the function fun1?

- A. n^3
- B. $n(\log n)^2$
- C. $n \log n$
- D. $n \log(\log n)$

[gate2015-1](#) [algorithms](#) [normal](#) [identify-function](#)**Answer****1.9.33 Identify Function: GATE2015-2-11** top<https://gateoverflow.in/8060>

Consider the following C function.

```

int fun(int n) {
    int x=1, k;
    if (n==1) return x;
    for (k=1; k<n; ++k)
        x = x + fun(k) * fun (n-k);
    return x;
}

```

The return value of $fun(5)$ is _____.

[gate2015-2](#) [algorithms](#) [identify-function](#) [recurrence](#) [normal](#) [numerical-answers](#)

Answer

1.9.34 Identify Function: GATE2015-3-49 [top](#)

<https://gateoverflow.in/8558>

Suppose $c = \langle c[0], \dots, c[k-1] \rangle$ is an array of length k , where all the entries are from the set $\{0, 1\}$. For any positive integers a and n , consider the following pseudocode.

DOSOMETHING (c, a, n)

```

z ← 1
for i ← 0 to k – 1
    do z ←  $z^2 \bmod n$ 
    if  $c[i]=1$ 
        then z ←  $(z \times a) \bmod n$ 
return z

```

If $k = 4, c = \langle 1, 0, 1, 1 \rangle, a = 2$, and $n = 8$, then the output of DOSOMETHING(c, a, n) is _____.

[gate2015-3](#) [algorithms](#) [identify-function](#) [normal](#) [numerical-answers](#)

Answer

1.9.35 Identify Function: TIFR2010-B-24 [top](#)

<https://gateoverflow.in/18742>

Consider the following program operating on four variables u, v, x, y , and two constants X and Y .

```

x, y, u, v := X, Y, Y, X;
While (x ≠ y)
do
    if (x > y) then x, v := x - y, v + u;
    else if (y > x) then y, u := y - x, u + v;
od;
print ((x + y) / 2); print ((u + v) / 2);

```

Given $X > 0 \wedge Y > 0$, pick the true statement out of the following:

- A. The program prints $\gcd(X, Y)$ and the first prime larger than both X and Y .
- B. The program prints $\gcd(X, Y)$ followed by $\text{lcm}(X, Y)$.
- C. The program prints $\gcd(X, Y)$ followed by $\frac{1}{2} \times \text{lcm}(X, Y)$.
- D. The program prints $\frac{1}{2} \times \gcd(X, Y)$ followed by $\frac{1}{2} \times \text{lcm}(X, Y)$.
- E. The program does none of the above.

[tifr2010](#) [algorithms](#) [identify-function](#)

Answer

1.9.36 Identify Function: TIFR2014-B-2 [top](#)

<https://gateoverflow.in/27136>

Consider the following code.

```
def brian(n):
    count = 0

    while (n != 0):
        n = n & (n-1)
        count = count + 1

    return count
```

Here n is meant to be an unsigned integer. The operator $\&$ considers its arguments in binary and computes their bit wise *AND*. For example, $22 \& 15$ gives 6, because the binary (say 8-bit) representation of 22 is 00010110 and the binary representation of 15 is 00001111, and the bit-wise *AND* of these binary strings is 00000110, which is the binary representation of 6. What does the function *brian* return?

- a. The highest power of 2 dividing n , but zero if n is zero.
- b. The number obtained by complementing the binary representation of n .
- c. The number of ones in the binary representation of n .
- d. The code might go into an infinite loop for some n .
- e. The result depends on the number of bits used to store unsigned integers.

[tifr2014](#) [algorithms](#) [identify-function](#)

Answer

1.9.37 Identify Function: TIFR2017-A-12 [top](#)

<https://gateoverflow.in/95299>

Consider the following program modifying an $n \times n$ square matrix A :

```
for i=1 to n:
    for j=1 to n:
        temp=A[i][j]+10
        A[i][j]=A[j][i]
        A[j][i]=temp-10
    end for
end for
```

Which of the following statements about the contents of matrix A at the end of this program must be TRUE?

- A. the new A is the transpose of the old A
- B. all elements above the diagonal have their values increased by 10 and all the values below have their values decreased by 10
- C. all elements above the diagonal have their values decreased by 10 and all the values below have their values increased by 10
- D. the new matrix A is symmetric, that is, $A[i][j] = A[j][i]$ for all $1 \leq i, j \leq n$
- E. A remains unchanged

[tifr2017](#) [algorithms](#) [identify-function](#)

Answer

Answers: Identify Function

1.9.1 Identify Function: CMI2010-B-07b top

<https://gateoverflow.in/47075>

Here, the op function takes two bits as input and outputs their XOR(\oplus).

Let x_1 and x_2 be the two input bits in the function op .

Then, the output is x_3 s.t. $x_3 = x_1 \oplus x_2 \rightarrow (\text{output of } op \text{ function})$

Now, the mystery 1 function takes two lists s and t as input and outputs another list u (say) such that

$u = s \oplus t \rightarrow (\text{output of mystery 1 function})$

and the output of mystery 2 function is

$$u = \underbrace{s \oplus s \oplus s \oplus s \dots \oplus s}_{2^{\text{length}(t)} \text{ times}}$$

- i. Taking XOR of s , $2^{\text{length}(t)}$ times does not change the $\text{length}(s)$. So, our answer will be $\text{length}(s)$.
- ii. In this problem the two bits of length s are 11, so taking the XOR of 1, even number of times gives the output 0. So, our answer is 00.

👍 3 votes

-- Kushagra Chatterjee (8.2k points)

1.9.2 Identify Function: CMI2013-A-09 top

<https://gateoverflow.in/46599>



Selected Answer

Answer is B . Sorted in descending order (selection sorting algorithm is used).

👍 8 votes

-- Dhananjay Kumar Sharma (25.7k points)

1.9.3 Identify Function: GATE1990-11b top

<https://gateoverflow.in/85987>



Selected Answer

The returned value is

$$\sum_{i=1}^n (-1)^n \frac{x^i}{i!}$$

👍 2 votes

-- Arjun Suresh (350k points)

1.9.4 Identify Function: GATE1991-03-viii top

<https://gateoverflow.in/523>



Selected Answer

For $N = 9$, it returns 3.

For $N = 10$ it returns 4.

For $N = 16$ it returns 4.

For $N = 17$ it returns 5.

So answer should be **C**.

10 votes

-- Taymiyyah Bhat (3.5k points)

1.9.5 Identify Function: GATE1993-7.4 top

<https://gateoverflow.in/2292>



Selected Answer

Answer is simply A i.e. it swaps the values of the two.. Take any two values for A and B . and perform the given operations over them.

16 votes

-- Gate Keeda (19.6k points)

1.9.6 Identify Function: GATE1994-6 top

<https://gateoverflow.in/2502>



Selected Answer

answer - x^n

7 votes

-- Ankit Rokde (9k points)

1.9.7 Identify Function: GATE1995-1.4 top

<https://gateoverflow.in/2591>



Selected Answer

Answer of X remains unchanged. As the if condition becomes false.

$X := -10$

Answer is **C** . This is classic example of *if – else* issue. Always *else* matches for nesting to the closest *if* in C Programming & Pascal .

https://en.wikipedia.org/wiki/Dangling_else

```
if (x>y)
{
    if (x<0)
        x=abs(x)
    else
        x=2*x
}
```

21 votes

-- Akash Kanase (42.5k points)

1.9.8 Identify Function: GATE1995-2.3 top

<https://gateoverflow.in/2615>



Selected Answer

Answer: **C**

Let $X = 3$ and $Y = 7$.

1st pass: $X = 3, Y = 4$

2nd pass: $X = 3, Y = 1$

3rd pass: $X = 2, Y = 1$

4th pass: $X = 1, Y = 1$

write (X), which writes 1.

Ref: <http://www.naturalnumbers.org/EuclidSubtract.html>

11 votes

-- Rajarshi Sarkar (34.1k points)

1.9.9 Identify Function: GATE1995-4 top

<https://gateoverflow.in/2640>



Selected Answer

- a. =3
- b.

```
begin
for I:=2 to N do
for J:=1 to ( I-1) do
begin
TMP:= A[I, J];
A[I, J]:= A[J, I];
A[J, I]:= TMP
end
```

Should be the condition...

8 votes

-- papesh (25.7k points)

1.9.10 Identify Function: GATE1998-2.12 top

<https://gateoverflow.in/1684>



Selected Answer

Value returned by $\text{fun}(95) = \text{fun}(\text{fun}(106))$

$$\begin{aligned} &= \text{fun}(96) \\ &= \text{fun}(\text{fun}(107)) \\ &= \text{fun}(97) \\ &= \text{fun}(\text{fun}(108)) \\ &= \text{fun}(98) \\ &= \text{fun}(\text{fun}(109)) \\ &= \text{fun}(99) \\ &= \text{fun}(\text{fun}(110)) \\ &= \text{fun}(100) \\ &= \text{fun}(\text{fun}(111)) \\ &= \text{fun}(101) = 91. \end{aligned}$$

23 votes

-- Digvijay (54.9k points)

1.9.11 Identify Function: GATE1999-2.24 top

<https://gateoverflow.in/1501>



Selected Answer

a	b	c	Return
1	1	1	The final return statement is $c < b$, so this never returns. Answer D.

37 votes

-- Arjun Suresh (350k points)

1.9.12 Identify Function: GATE2000-2.15 top

<https://gateoverflow.in/662>



Selected Answer

Answer is **A.**

Effect of the above 3 reversals for any K is equivalent to left rotation of the array of size n by k .

Let , $S[1.....7]$

1	2	3	4	5	6	7
---	---	---	---	---	---	---

so, $n = 7, k = 2$

reverse ($S, 1, 2$) we get $[2, 1, 3, 4, 5, 6, 7]$

reverse ($S, 3, 7$) we get $[2, 1, 7, 6, 5, 4, 3]$

reverse ($S, 1, 7$) we get $[3, 4, 5, 6, 7, 1, 2]$

Hence, option (A) rotates s left by k positions and is correct.

拇指图标 29 votes

-- Kalpana Bhargav (3.3k points)

1.9.13 Identify Function: GATE2003-1 top

<https://gateoverflow.in/892>



Selected Answer

A simplified version of the given program can be written as:

```
float f(float x, int y) {
    float p=1, s=1;
    int i;
    for (i=1; i<y; i++) {
        p = p * (x/i);
        s = s + p;
    }
    return s;
}
```

We can take $p = 1, s = 1$ initialization outside of for loop because there is no condition checking in for loop involving p, s .

i	$p = p * (x/i)$	$s = s + p$
1	x	$1 + x$
2	$\frac{x^2}{2}$	$1 + x + \frac{x^2}{2}$
3	$\frac{x^3}{6}$	$1 + x + \frac{x^2}{2} + \frac{x^3}{6}$

		$1 + x$
		$+ \frac{x^2}{2}$
		$+ \frac{x^3}{6}$
		$+ \frac{x^4}{24}$
4	$\frac{x^4}{24}$	
n	$\frac{x^n}{n!}$	e^x
$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \dots + \frac{x^n}{n!}$		

Hence, option **B** is answer.

16 votes

-- Rajesh Pradhan (22.6k points)

1.9.14 Identify Function: GATE2003-88 top

<https://gateoverflow.in/971>



Selected Answer

The nested loop is taking all integers from 2 to $2 * \log_2 n$. Take all their non-multiples before n , and make the corresponding entry in A as 1. For example, for 2, and $n = 10$, $A[3], A[5], A[7]$, and $A[9]$ are made 1. Similarly for 3, 4, ... till $2 * \log n$. So, if any entry $A[p]$ is 1 means it must be a multiple of $2, 3, \dots, 2 * \log_2 n$, which is $(2 \log n)!$ and is greater than n . So, for no index p , $A[p]$ will be 0. So, answer is **D**.

Suppose the line

`A[j] = A[j] || (j%k);`

is replaced with

`A[j] = A[j] || !(j%k);`

Now, the nested loop is taking all integers from 2 to $\log_2 n$, take all their multiples before n , and make the corresponding entry in A as 1. For example, for 2, and $n = 10$, $A[4], A[6], A[8]$ and $A[10]$ are made 1. Similarly for 3, 4, ... till $2 * \log n$. So, for all non-prime indices of A , we will have a 1, and for prime indices we have a 0. And we print i if $A[j]$ is 0 meaning j is prime.

20 votes

-- Arjun Suresh (350k points)

1.9.15 Identify Function: GATE2004-41 top

<https://gateoverflow.in/1038>



Selected Answer

It is an algorithm for gcd computation.

Here, while loop executes until $m = n$.

We can test by taking any two numbers as m, n .

Answer will be **(C)**.

Ref: <http://www.naturalnumbers.org/EuclidSubtract.html>

9 votes

-- srestha (87k points)

1.9.16 Identify Function: GATE2004-42 top

<https://gateoverflow.in/1039>



Selected Answer

By putting $y = m/x$ into $x = (x + y)/2$

$$x = (x + m/x)/2$$

$$\implies 2x^2 = x^2 + m$$

$$\implies x = m^{1/2}$$

We can also check by putting 2 or 3 different values also.

30 votes

-- gate_asp (749 points)

1.9.17 Identify Function: GATE2005-31 top

<https://gateoverflow.in/1367>



Selected Answer

Option is D.

foo is printing the lowest digit. But the *printf* inside it is after the recursive call. This forces the output to be in reverse order

2, 0, 4, 8

The final value *sum* printed will be 0 as C uses pass by value and hence the modified value inside *foo* won't be visible inside *main*.

16 votes

-- anshu (3.3k points)

1.9.18 Identify Function: GATE2005-IT-57 top

<https://gateoverflow.in/3818>



Selected Answer

See the following calling sequence:

Hence, answer is option C.

```
f(20,1)
  ↓
  f(10,2)-1
    ↓
    f(5,4)-2
      ↓
      f(2,8)+4
        ↓
        f(1,16)-8
          ↓
          f(0,32)+16
            ↓
            0
```

18 votes

-- Rajesh Pradhan (22.6k points)

1.9.19 Identify Function: GATE2006-50 top

<https://gateoverflow.in/1828>



Selected Answer

Option (D)

In the given algorithm the for loop contains a logical expression

Red Color Shows return values

$z[i] = (x[i] \wedge \sim y[i]) \vee (\sim x[i] \wedge y[i]);$

The equivalent set representation of a given logical expression if we assume

$z[i] = Z, x[i] = X, y[i] = Y$ then
 $Z = (X \wedge \neg Y) \vee (\neg X \wedge Y)$
 $\implies Z = (X - Y) \cup (Y - X) [\because A \wedge \neg B = A - B]$

13 votes

-- Prasanna Ranganathan (4.7k points)

1.9.20 Identify Function: GATE2006-53 top

<https://gateoverflow.in/1831>

The while loop adds elements from a and b (whichever is smaller) to c and terminates when either of them exhausts. So, when loop terminates either $i = n$ or $j = m$.

Suppose $i = n$. This would mean all elements from array a are added to $c \Rightarrow k$ must be incremented by n . c would also contain j elements from array b . So, number of elements in c would be $n + j$ and hence $k = n + j$.

Similarly, when $j = m$, $k = m + i$.

Hence, option (D) is correct. (Had k started from -1 and not 0 and we used $++k$ inside loop, answer would have been option (C))

41 votes

-- Arjun Suresh (350k points)

1.9.21 Identify Function: GATE2006-IT-52 top

<https://gateoverflow.in/3595>

Answer: C

Because $\binom{m}{0} = 1$ and $\binom{n}{n} = 1$.

21 votes

-- Rajarshi Sarkar (34.1k points)

1.9.22 Identify Function: GATE2008-IT-82 top

<https://gateoverflow.in/3406>

Answer: D

The function prints the binary equivalent of the number n .

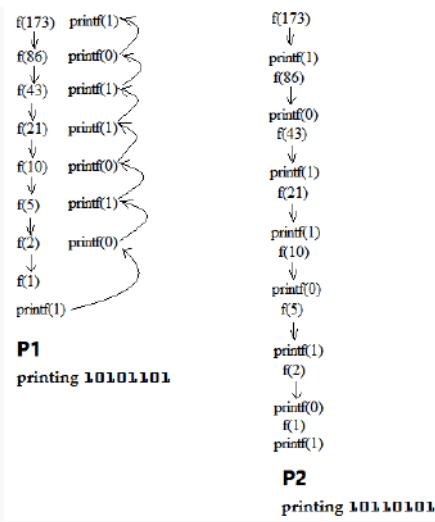
Binary equivalent of 173 is 10101101.

16 votes

-- Rajarshi Sarkar (34.1k points)

1.9.23 Identify Function: GATE2008-IT-83 top

<https://gateoverflow.in/3407>



Here, **P1** and **P2** will print opposite in direction as shown in diagram.

And given code fragment will print like **P1** and not like **P2**

Hence, answer will be (**C**).

14 21 votes

-- srestha (87k points)

1.9.24 Identify Function: GATE2009-18 top

<https://gateoverflow.in/1310>



Selected Answer

The answer is **B**.

Let the address of x be 1000.

$$1. f(5, 1000) = 8$$

$$2. f(4, 1000) = 5$$

$$3. f(3, 1000) = 3$$

$$4. f(2, 1000) = 2$$

$$5. f(1, 1000) = 1.$$

The evaluation is done from 5 to 1. Since recursion is used.

14 11 votes

-- Gate Keeda (19.6k points)

1.9.25 Identify Function: GATE2010-35 top

<https://gateoverflow.in/2336>

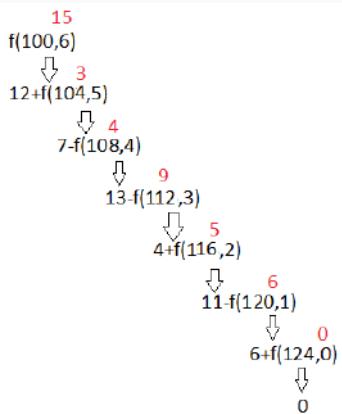


Selected Answer

Suppose int array takes 4 bytes for each element and stored at base address 100.

Follow below image. Red color shows the return value.

So, 15 is the answer.



12 votes

-- Rajesh Pradhan (22.6k points)

It will print

$$\begin{aligned}
 & 12 + (7 - (13 - (4 + (11 - (6 + 0))))) \\
 &= 12 + (7 - (13 - (4 + (11 - 6)))) \\
 &= 12 + 7 - 13 + 9 \\
 &= 15
 \end{aligned}$$

23 votes

-- gatecse (18k points)

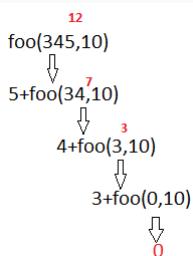
1.9.26 Identify Function: GATE2011-48 top

<https://gateoverflow.in/2154>



Red color represents return values.

Answer is
12.



17 votes

-- Rajesh Pradhan (22.6k points)

1.9.27 Identify Function: GATE2011-49 top

<https://gateoverflow.in/43324>



The function returns the sum of digits in a binary representation of the given number

$$\text{so } 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 = 2$$

16 votes

-- Sandeep_Uniyal (7.5k points)

1.9.28 Identify Function: GATE2013-31 [top](#)

<https://gateoverflow.in/1542>



Selected Answer

The outer loop is running for $n/2$ times and inner loop is running for $\log_2 n$ times (each iteration doubles j and j stops at n means $\log_2 n$ times j loop will iterate).

Now in each iteration k is incremented by $n/2$. So, overall k will be added $n/2 * \log n * n/2$ with an initial value of 0. So, final value of k will be $\Theta(n^2 \log n)$

👍 41 votes

-- Arjun Suresh (350k points)

1.9.29 Identify Function: GATE2014-1-41 [top](#)

<https://gateoverflow.in/1919>



Selected Answer

Answer is (A) maximum possible sum of elements in any sub-array of array E.

```
int MyX ( int * E, unsigned int size )
{
    int Y= 0;
    int z;
    int i, j,k;

//calculate sum of the elements of the array E and stores it in Y
    for (i=0;i<size;i++)
        Y = Y+E[i];

//calculate the sum of all possible subarrays
//(starting from position 0..n-1)
    for (i=0;i<size;i++)
        for(j=i;j<size ;j++)
        {
            z = 0;
            for(k=i; k<=j; k++)
                z=z+E[k];

//checks whether sum of elements of each subarray is greater
//than the current max, if so, then assign it to currentmax
            if(z>Y)
                Y = z;
        }
//ultimately returns the maximum possible sum of elements
//in any sub array of given array E
    return Y;
}
```

👍 25 votes

-- Kalpana Bhargav (3.3k points)

1.9.30 Identify Function: GATE2014-2-10 [top](#)

<https://gateoverflow.in/1964>



Selected Answer

Answer is 9.

435 – (110110011)

`num >>= 1;` implies a num is shifted one bit right in every while loop execution. While loop is executed 9 times successfully and 10th time num is zero.

So count is incremented 9 times.

Note:

Shifting a number "1" bit position to the right will have the effect of dividing by 2:

```
8 >> 1 = $4      // In binary: (00001000) >> 1 = (00000100)
```

1 23 votes

-- Prasanna Ranganathan (4.7k points)

1.9.31 Identify Function: GATE2014-3-10 top

<https://gateoverflow.in/2044>



A.

In the computation of given pseudo code for each row and column of Matrix A , each upper triangular element will be interchanged by its mirror image in the lower triangular and after that the same lower triangular element will be again re-interchanged by its mirror image in the upper triangular, resulting the final computed Matrix A same as input Matrix A .

1 24 votes

-- Gate Keeda (19.6k points)

1.9.32 Identify Function: GATE2015-1-31 top

<https://gateoverflow.in/8263>



i loop is executing n times. j loop is executing $\log n$ times for each i , and so value of p is $\log n$. k loop is executing $\log p$ times, which is $\log \log n$ times for each iteration of i . In each of these q is incremented. So, over all iterations of i , q will be incremented $n \log \log n$ times. So, D choice.

1 41 votes

-- Arjun Suresh (350k points)

1.9.33 Identify Function: GATE2015-2-11 top

<https://gateoverflow.in/8060>



$fun(1) = 1;$

$fun(2) = 1 + fun(1) * fun(1) = 1 + 1 = 2;$

$fun(3) = 1 + fun(1) * fun(2) + fun(2) * fun(1) = 5;$

$fun(4) = 1 + fun(1) * fun(3) + fun(2) * fun(2) + fun(3) * fun(1) = 1 + 5 + 4 + 5 = 15;$

$fun(5) = 1 + fun(1) * fun(4) + fun(2) * fun(3) + fun(3) * fun(2) + fun(4) * fun(1) =$

More formal way:

The recurrence relation is

$$f(n) = \begin{cases} 1, & n = 1 \\ 1 + \sum_{i=1}^{n-1} f(i) \times f(n-i), & n > 1 \end{cases}$$

$$f(1) = 1$$

$$f(2) = 1 + f(1). f(1) = 1 + 1.1 = 2$$

$$f(3) = 1 + f(1). f(2) + f(2). f(1) = 1 + 1.2 + 2.1 = 5$$

$$f(4) = 1 + f(1). f(3) + f(2). f(2) + f(3). f(2) = 1 + 1.5 + 2.2 + 5.1 = 15$$

$$f(5) = 1 + f(1). f(4) + f(2). f(3) + f(3). f(2) + f(4). f(1) = 1 + 1.15 + 2.5 + 5.2 + 15.1 =$$

Up 53 votes

-- Arjun Suresh (350k points)

1.9.34 Identify Function: GATE2015-3-49 [top](#)

<https://gateoverflow.in/8558>



Selected Answer

Initially $k = 4$, $c = [1, 0, 1, 1]$, $a = 2$, $n = 8$.

Now let's iterate through the function step by step :

$z = 1$ (at the start of do-something)

$i = 0$ (start of external for loop)

In the do loop

$z = 1 * 1$ (non zero value so considered as true and continue)

$c[0] = 1$, so in the if clause, $z = 1 * 2 \% 8 = 2$

In the do loop

$z = 2 * 2 \% 8 = 4$ (since now $z = 2$) (non zero value so considered as true and continue)

$c[0] = 1$, so in the if clause, $z = 4 * 2 \% 8 = 0$

Now no need to check further :

Reason : All the operations that update Z are multiplicative operations and hence the value of Z will never change from 0.

Up 26 votes

-- Tamojit Chatterjee (2.3k points)

1.9.35 Identify Function: TIFR2010-B-24 [top](#)

<https://gateoverflow.in/18742>

It prints with $\gcd(x, y)$ and $\text{lcm}(x, y)$.

Consider $x, y, u, v = 17, 3, 3, 17$.

$X = 14, v = 20$

$X = 11, v = 23$

$X = 8, v = 26$

$X = 5, v = 29$

$X = 2, v = 32$

$Y = 1, u = 35$

$X = 1, v = 67$

This is the value obtained.

Lastly, printing $(x + y)/2$ and $(v + u)/2$ gives 1 and 51.

Upvote 8 votes

-- zambus (279 points)

1.9.36 Identify Function: TIFR2014-B-2 top

<https://gateoverflow.in/27136>



Selected Answer

Option C. It returns no of 1's in binary representation of n .
Here, $n \& (n - 1)$ reset rightmost bit of n in each iteration.

For example,

Suppose $n = 15 = 00001111$ (binary)

$n - 1 = 14(00001110)$

```
00001111
^ 00001110
-----
00001110
```

Upvote 6 votes

-- Avdhesh Singh Rana (3.2k points)

1.9.37 Identify Function: TIFR2017-A-12 top

<https://gateoverflow.in/95299>



Selected Answer

```
1. for i=1 to n:
2.   for j=1 to n:
3.     temp=A[i][j]+10
4.     A[i][j]=A[j][i]
5.     A[j][i]=temp-10
6.   end for
7.end for
```

The 3, 4, 5 lines swap $A[j][i]$ and $A[i][j]$.

The same variables are swapped twice. For eg when: $i = 5, j = 10$. $A[10][5]$ and $A[5][10]$ will be swapped. They will be swapped again when $i = 10, j = 5$.

Two times swap of same elements will lead to A remaining unchanged.

Hence, E is correct.

6 votes

-- tarun_svbk (1.5k points)

1.10**Minimum Maximum(4)** top**1.10.1 Minimum Maximum: GATE2014-1-39** top<https://gateoverflow.in/1917>

The minimum number of comparisons required to find the minimum and the maximum of 100 numbers is _____

gate2014-1 algorithms numerical-answers normal minimum-maximum

Answer**1.10.2 Minimum Maximum: TIFR2014-B-10** top<https://gateoverflow.in/27198>

Given a set of n distinct numbers, we would like to determine both the smallest and the largest number. Which of the following statements is TRUE?

- A. These two elements can be determined using $O(\log^{100} n)$ comparisons.
- B. $O(\log^{100} n)$ comparisons do not suffice, however these two elements can be determined using $n + O(\log n)$ comparisons.
- C. $n + O(\log n)$ comparisons do not suffice, however these two elements can be determined using $3\lceil n/2 \rceil$ comparisons.
- D. $3\lceil n/2 \rceil$ comparisons do not suffice, however these two elements can be determined using $2(n - 1)$ comparisons.
- E. None of the above.

tifr2014 algorithms minimum-maximum

Answer**1.10.3 Minimum Maximum: TIFR2014-B-6** top<https://gateoverflow.in/27183>

Consider the problem of computing the minimum of a set of n distinct numbers. We choose a permutation uniformly at random (i.e., each of the $n!$ permutations of $\langle 1, \dots, n \rangle$ is chosen with probability $(1/n!)$) and we inspect the numbers in the order given by this permutation. We maintain a variable MIN that holds the minimum value seen so far. MIN is initialized to ∞ and if we see a value smaller than MIN during our inspection, then MIN is updated. For example, in the inspection given by the following sequence, MIN is updated four times.

5 9 4 2 6 8 0 3 1 7

What is the expected number of times MIN is updated?

- A. $O(1)$
- B. $H_n = \sum_{i=1}^n 1/i$
- C. \sqrt{n}
- D. $n/2$
- E. n

tifr2014 algorithms minimum-maximum

Answer**1.10.4 Minimum Maximum: TIFR2014-B-9** top<https://gateoverflow.in/27194>

Given a set of n distinct numbers, we would like to determine the smallest three numbers in this set using comparisons. Which of the following statements is TRUE?

- A. These three elements can be determined using $O(\log^2 n)$ comparisons.
- B. $O(\log^2 n)$ comparisons do not suffice, however these three elements can be determined using $n + O(1)$ comparisons.
- C. $n + O(1)$ comparisons do not suffice, however these three elements can be determined using $n + O(\log n)$ comparisons.
- D. $n + O(\log n)$ comparisons do not suffice, however these three elements can be determined using $O(n)$ comparisons.
- E. None of the above.

[tifr2014](#) | [algorithms](#) | [minimum-maximum](#)

[Answer](#)

Answers: Minimum Maximum

1.10.1 Minimum Maximum: GATE2014-1-39 [top](#)

<https://gateoverflow.in/1917>



Selected Answer

We can solve this question by using **Tournament Method Technique** -

1. To find the smallest element in the array will take $n - 1$ comparisons = 99.
 2. To find the largest element -
 - a. After the first round of Tournament , there will be exactly $n/2$ numbers = 50 that will loose the round.
 - b. So, the biggest looser (the largest number) should be among these 50 losers. To find the largest number will take $n/2 - 1$ comparisons = 49.
- Total $99 + 49 = 148$.

37 votes

-- Harsh181996 (4k points)

1.10.2 Minimum Maximum: TIFR2014-B-10 [top](#)

<https://gateoverflow.in/27198>



Selected Answer

I think answer will be C.

To be accurate, it will need $3n/2 - 2$ comparisons .

8 votes

-- Pranay Datta (10.2k points)

1.10.3 Minimum Maximum: TIFR2014-B-6 [top](#)

<https://gateoverflow.in/27183>



Selected Answer

Let us consider 3 numbers $\{1, 2, 3\}$

We will consider the permutation along with min no of times MIN is updated .

Permutation : No of times MIN updated (Minimum)

1, 2, 3	1
1, 3, 2	1
2, 1, 3	2
2, 3, 1	2
3, 1, 2	2
3, 2, 1	3

Total number of times MIN updated is : 11.

Average no of times MIN updated is : $(11/6)$

Now going by the options i am getting B .

$$H_3 = 1 + 1/2 + 1/3 = 11/6 .$$

H_3 is the answer and that is option **B** .

17 votes

-- Riya Roy(Arayana) (7.2k points)

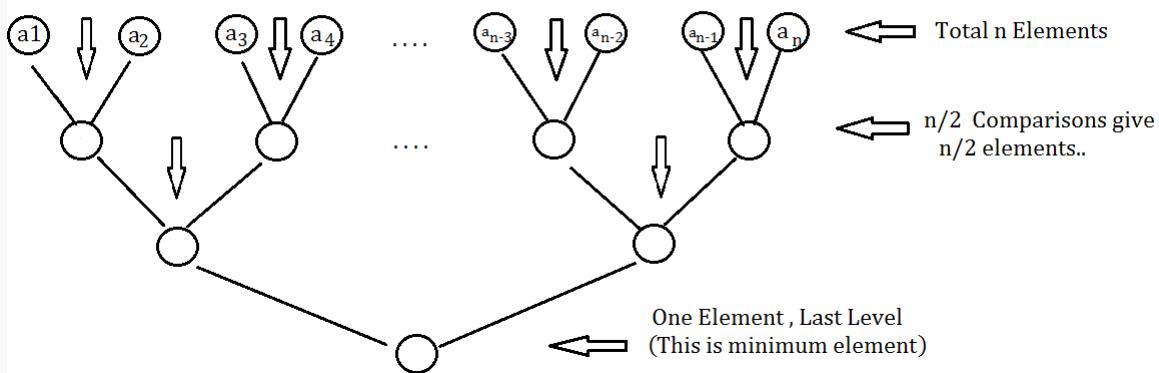
1.10.4 Minimum Maximum: TIFR2014-B-9 top

<https://gateoverflow.in/27194>



Option (C) is correct. Reason is as follows :

One Comparison
to get minimum



Here, at first level we are Given n elements, out of which we have to find smallest 3 numbers.

We compare $2 - 2$ elements as shown in figure & get $n/2$ elements at Second level.

Note: Minimum element is in these $n/2$ elements.

So, comparisons for this is $n/2$.

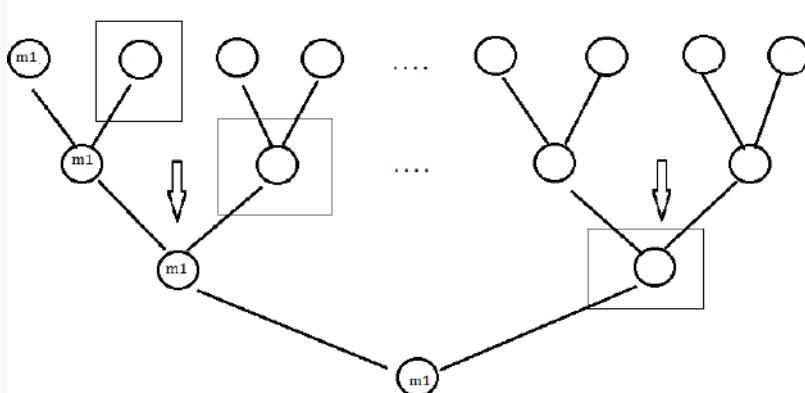
Similarly for next level we have $n/4$ Comparisons & $n/2$ elements and so on.

Total Comparisons till now is $n/2 + n/4 + n/8 + \dots + 4 + 2 + 1 = (2^{\log n} - 1) = n - 1$ {Use G.P. sum}

Now we have to get smallest 3.

We have 1st smallest at last level already $\Rightarrow 0$ Comparison for this.

\Rightarrow 2nd & 3rd smallest can be found in $O(\log n)$ time as shown below:



Minimum Element must have descended down from some path from top to Bottom.

\Rightarrow SQUARES represent Candidates for 2nd minimum.

Every element that is just below $m1$ (first minimum) is a candidate for second minimum.

So, $O(\log n)$ Comparisons for finding second smallest.

\Rightarrow Similarly for 3rd minimum we get $O(\log n)$ time.. As, every element that is just below 1st & 2nd minimum is a candidate for 3rd minimum.

18 votes

-- Himanshu Agarwal (15.3k points)

1.11

Minimum Spanning Trees(2) top

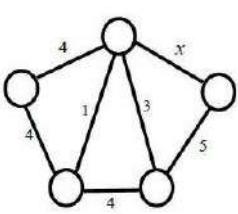
1.11.1 Minimum Spanning Trees: CMI2012-B-05b top <https://gateoverflow.in/47090>

Given an undirected weighted graph $G = (V, E)$ with non-negative edge weights, we can compute a minimum cost spanning tree $T = (V, E')$. We can also compute, for a given source vertex $s \in V$, the shortest paths from s to every other vertex in V . We now increase the weight of every edge in the graph by 1. Are the following true or false, regardless of the structure of G ? Give a mathematically sound argument if you claim the statement is true or a counterexample if the statement is false.

- i. All the shortest paths from s to the other vertices are unchanged.

Answer**1.11.2 Minimum Spanning Trees: GATE2018-47** [top](#)<https://gateoverflow.in/204122>

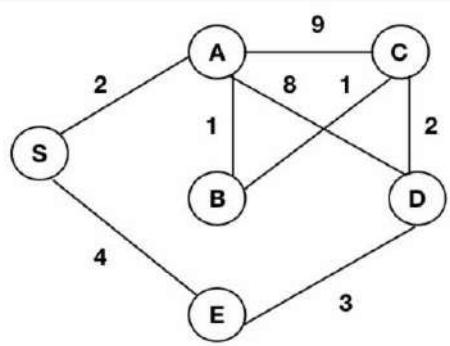
Consider the following undirected graph G :



Choose a value for x that will maximize the number of minimum weight spanning trees (MWSTs) of G . The number of MWSTs of G for this value of x is ____.

[gate2018](#) [algorithms](#) [graph-algorithms](#) [minimum-spanning-trees](#) [numerical-answers](#)
Answer**Answers: Minimum Spanning Trees****1.11.1 Minimum Spanning Trees: CMI2012-B-05b** [top](#)<https://gateoverflow.in/47090>

Selected Answer



The given statement "All the shortest paths from s to the other vertices are unchanged." is false . From the above graph it is clear that the shortest path from S to D is $S \Rightarrow A \Rightarrow B \Rightarrow C \Rightarrow D$ and the cost is 6.

Now, we increment the edge cost of all the edges by 1.

After incrementation, the shortest path from S to D gets changed. Now the shortest path becomes $S \Rightarrow E \Rightarrow D$ and shortest path cost is 9. The above graph is the proof.

👍 5 votes
-- **balaeinstein** (1.3k points)**1.11.2 Minimum Spanning Trees: GATE2018-47** [top](#)<https://gateoverflow.in/204122>

Selected Answer

Number of possible MSTs increase, when we have multiple edges with same edge weights.

To maximize the number of MST, x should be 5.

In the question, number of MST is asked for the value of X .

So, number of MST = $2 \times 2 = 4$ (**Answer**)

(Because one 4 forms cycle, can't be included in any way. Now from two 4 and 5 we can select one in $2 \times 2 = 4$ ways)

11 votes

-- Ahwan Mishra (10.5k points)

1.12

Numerical Computation(2) top

1.12.1 Numerical Computation: GATE2014-1-37 top

<https://gateoverflow.in/1915>

There are 5 bags labeled 1 to 5. All the coins in a given bag have the same weight. Some bags have coins of weight 10 gm, others have coins of weight 11 gm. I pick 1, 2, 4, 8, 16 coins respectively from bags 1 to 5. Their total weight comes out to 323 gm. Then the product of the labels of the bags having 11 gm coins is ____.

gate2014-1 algorithms numerical-answers normal numerical-computation

Answer

1.12.2 Numerical Computation: TIFR2014-B-20 top

<https://gateoverflow.in/27354>

Consider the following game. There is a list of distinct numbers. At any round, a player arbitrarily chooses two numbers a, b from the list and generates a new number c by subtracting the smaller number from the larger one. The numbers a and b are put back in the list. If the number c is non-zero and is not yet in the list, c is added to the list. The player is allowed to play as many rounds as the player wants. The score of a player at the end is the size of the final list.

Suppose at the beginning of the game the list contains the following numbers: 48, 99, 120, 165 and 273. What is the score of the best player for this game?

- A. 40
- B. 16
- C. 33
- D. 91
- E. 123

tifr2014 algorithms numerical-computation

Answer

Answers: Numerical Computation

1.12.1 Numerical Computation: GATE2014-1-37 top

<https://gateoverflow.in/1915>



Selected Answer

Suppose X is the number of coins of 11 gm and Y is the number of 10 gm coins.
According to question,

$$11X + 10Y = 323 \rightarrow (1)$$

$$X + Y = 31 \rightarrow (2)$$

Solving (1), (2), we get $X = 13$ and $Y = 18$. So, here number of coins of 11 gm is 13 and the only possible combination for 13 coins is

- 1 coin from bag1
- 4 coins from bag3
- 8 coins from bag4

So, product of label of bags will be $= 1 \times 3 \times 4 = 12$.

40 votes

-- sonam vyas (14.8k points)

1.12.2 Numerical Computation: TIFR2014-B-20 [top](#)

<https://gateoverflow.in/27354>



Selected Answer

Option D is correct

Here the list is (48, 99, 120, 165, 273).

$Gcd(48, 99) = 3$, means if we subtract $(99 - 48 = 51)$ then that number is also % 3,

So the numbers like (3, 6, 9, ..., 99) are added. Total numbers $= 99/3 = 33$

//y $Gcd(48, 120) = 24$, so the numbers %24 are added like (24, 48, ..., 120). Total numbers $= 120/24 = 5$

//y $Gcd(48, 165) = 3$, so the numbers (3, 6, 9, ..., 24, ..., 48, ..., 99, ..., 120, ..., 165) are added. Totally, $165/3 = 55$

At end, $Gcd(48, 273) = 3$, so the numbers

(3, 6, 9, ..., 24, ..., 48, ..., 99, ..., 120, ..., 165, ..., 273) are added (which covers all the above numbers)

So total numbers added to this list $= 273/3 = 91$

10 votes

-- venky.victory35 (753 points)

1.13

P Np Npc Nph(12) [top](#)

1.13.1 P Np Npc Nph: CMI2010-A-10 [top](#)

<https://gateoverflow.in/46142>

Consider the following statements.

1. NP-complete problems are those that we know we can never solve efficiently.
2. If we find an efficient algorithm for one NP-complete problem, then we can solve all NP-complete problems efficiently.
3. Checking whether a number is a prime is an NP-complete problem.

Then:

- A. 1 and 2 are true but 3 is false.
- B. 1 and 3 are false but 2 is true.
- C. 2 and 3 are true but 1 is false.
- D. All three statements are false.

[cmi2010](#) [algorithms](#) [p-np-npc-nph](#)

Answer**1.13.2 P Np Npc Nph: CMI2015-A-06** [top](#)<https://gateoverflow.in/47041>

Suppose we have constructed a polynomial time reduction from problem A to problem B . Which of the following can we infer from this fact?

- A. If the best algorithm for B takes exponential time, there is no polynomial time algorithm for A .
- B. If the best algorithm for A takes exponential time, there is no polynomial time algorithm for B .
- C. If we have a polynomial time algorithm for A , we must also have a polynomial time algorithm for B .
- D. If we don't know whether there is a polynomial time algorithm for B , there cannot be a polynomial time algorithm for A .

[cmi2015](#) [algorithms](#) [p-np-npc-nph](#)**Answer****1.13.3 P Np Npc Nph: CMI2017-A-10** [top](#)<https://gateoverflow.in/203287>

We have constructed a polynomial time reduction from problem A to problem B . Which of the following is a valid inference?

- A. If the best algorithm for B takes exponential time, then there is no polynomial time algorithm for A .
- B. If the best algorithm for A takes exponential time, then there is no polynomial time algorithm for B .
- C. If we have a polynomial time algorithm for A , then we must also have a polynomial time algorithm for B .
- D. If we don't know whether there is a polynomial time algorithm for B , then there cannot be a polynomial time algorithm for A .

[cmi2017](#) [algorithms](#) [theory-of-computation](#) [p-np-npc-nph](#)**Answer****1.13.4 P Np Npc Nph: GATE1992-02,vi** [top](#)<https://gateoverflow.in/561>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Which of the following problems is not NP -hard?

- a. Hamiltonian circuit problem
- b. The 0/1 Knapsack problem
- c. Finding bi-connected components of a graph
- d. The graph coloring problem

[gate1992](#) [p-np-npc-nph](#) [algorithms](#)**Answer****1.13.5 P Np Npc Nph: GATE2003-12** [top](#)<https://gateoverflow.in/903>

Ram and Shyam have been asked to show that a certain problem Π is NP-complete. Ram shows a polynomial time reduction from the 3-SAT problem to Π , and Shyam shows a polynomial time reduction from Π to 3-SAT. Which of the following can be inferred from these reductions?

- A. Π is NP-hard but not NP-complete
- B. Π is in NP, but is not NP-complete
- C. Π is NP-complete
- D. Π is neither NP-hard, nor in NP

gate2003 algorithms p-np-npc-nph normal

[Answer](#)

1.13.6 P Np Npc Nph: GATE2004-30, ISRO2017-10 [top](#) <https://gateoverflow.in/1027>

The problem 3-SAT and 2-SAT are

- A. both in P
- B. both NP complete
- C. NP -complete and in P respectively
- D. undecidable and NP complete respectively

gate2004 algorithms p-np-npc-nph easy isro2017

[Answer](#)

1.13.7 P Np Npc Nph: GATE2006-16, ISRO-DEC2017-27 [top](#) <https://gateoverflow.in/977>

Let S be an NP-complete problem and Q and R be two other problems not known to be in NP. Q is polynomial time reducible to S and S is polynomial-time reducible to R . Which one of the following statements is true?

- A. R is NP-complete
- B. R is NP-hard
- C. Q is NP-complete
- D. Q is NP-hard

gate2006 algorithms p-np-npc-nph normal isrodec2017

[Answer](#)

1.13.8 P Np Npc Nph: GATE2008-44 [top](#) <https://gateoverflow.in/456>

The subset-sum problem is defined as follows: Given a set S of n positive integers and a positive integer W , determine whether there is a subset of S whose elements sum to W . An algorithm Q solves this problem in $O(nW)$ time. Which of the following statements is false?

- A. Q solves the subset-sum problem in polynomial time when the input is encoded in unary
- B. Q solves the subset-sum problem in polynomial time when the input is encoded in binary
- C. The subset sum problem belongs to the class NP
- D. The subset sum problem is NP-hard

gate2008 algorithms p-np-npc-nph normal

[Answer](#)

1.13.9 P Np Npc Nph: TIFR2010-B-39 [top](#) <https://gateoverflow.in/18754>

Suppose a language L is **NP** complete. Then which of the following is FALSE?

- A. $L \in \mathbf{NP}$
- B. Every problem in **P** is polynomial time reducible to L .
- C. Every problem in **NP** is polynomial time reducible to L .
- D. The Hamilton cycle problem is polynomial time reducible to L .
- E. $\mathbf{P} \neq \mathbf{NP}$ and $L \in \mathbf{P}$.

[tifr2010](#) [algorithms](#) [p-np-npc-nph](#)

Answer

1.13.10 P Np Npc Nph: TIFR2011-B-37 [top](#)

<https://gateoverflow.in/20922>

Given an integer $n \geq 3$, consider the problem of determining if there exist integers $a, b \geq 2$ such that $n = a^b$. Call this the forward problem. The reverse problem is: given a and b , compute $a^b \pmod{b}$. Note that the input length for the forward problem is $\lfloor \log n \rfloor + 1$, while the input length for the reverse problem is $\lfloor \log a \rfloor + \lfloor \log b \rfloor + 2$. Which of the following statements is TRUE?

- a. Both the forward and reverse problems can be solved in time polynomial in the lengths of their respective inputs.
- b. The forward problem can be solved in polynomial time, however the reverse problem is *NP-hard*.
- c. The reverse problem can be solved in polynomial time, however the forward problem is *NP-hard*.
- d. Both the forward and reverse problem are *NP-hard*.
- e. None of the above.

[tifr2011](#) [algorithms](#) [p-np-npc-nph](#)

Answer

1.13.11 P Np Npc Nph: TIFR2013-B-7 [top](#)

<https://gateoverflow.in/25668>

Which of the following is not implied by $P = NP$?

- a. 3SAT can be solved in polynomial time.
- b. Halting problem can be solved in polynomial time.
- c. Factoring can be solved in polynomial time.
- d. Graph isomorphism can be solved in polynomial time.
- e. Travelling salesman problem can be solved in polynomial time.

[tifr2013](#) [algorithms](#) [p-np-npc-nph](#)

Answer

1.13.12 P Np Npc Nph: TIFR2017-B-2 [top](#)

<https://gateoverflow.in/95673>

Consider the following statements:

- i. Checking if a given *undirected* graph has a cycle is in **P**
- ii. Checking if a given *undirected* graph has a cycle is in **NP**
- iii. Checking if a given *directed* graph has a cycle is in **P**
- iv. Checking if a given *directed* graph has a cycle is in **NP**

Which of the above statements is/are TRUE? Choose from the following options.

- A. Only i and ii
- B. Only ii and iv

- C. Only ii, iii, and iv
 D. Only i, ii and iv
 E. All of them

tifr2017 | algorithms | p-np-npc-nph

[Answer](#)

Answers: P Np Npc Nph

1.13.1 P Np Npc Nph: CMI2010-A-10 [top](#)

<https://gateoverflow.in/46142>

Option is *B*.

1 upvotes

-- shubham (221 points)

1.13.2 P Np Npc Nph: CMI2015-A-06 [top](#)

<https://gateoverflow.in/47041>



Selected Answer

Problem

A reduces to Problem
B,

Option **(B)** If the best algorithm for *A* takes exponential time, there is no polynomial time algorithm for *B*. will be correct.

1 upvotes

-- Muktinath Vishwakarma (35.4k points)

1.13.3 P Np Npc Nph: CMI2017-A-10 [top](#)

<https://gateoverflow.in/203287>

A is reducible to *B* implies *B* is as tough as *A*. (*A* cannot be harder than *B*)

Option *A* - False. If *A* is polynomial then *B* must be Polynomial (*A* polynomial algorithm can be easily converted into exponential. Converse is not true).

Option *B*- True. As per first line above, if *A* is exponential then *B* cannot be a polynomial time.

Option *C* - False. If we have polynomial time algorithm for *A* then we can have polynomial, exponential, sub exponential algorithm for *B*.

Option *D*- False. *A* can be polynomial. *B* can be harder than polynomial (as per first line, *B* is as hard as *A* can be termed as $B = \Omega(A)$).

1 upvotes

-- Keval Malde (16.1k points)

1.13.4 P Np Npc Nph: GATE1992-02,vi [top](#)

<https://gateoverflow.in/561>



Selected Answer

- Is NPC and hence NP hard.
- Is again NP hard (optimization version is NP hard and decision version is NPC).
 Ref: <http://stackoverflow.com/questions/3907545/how-to-understand-the-knapsack-problem-is-np-complete>

- c. Is in P. See the algorithm here based on DFS: http://en.wikipedia.org/wiki/Biconnected_component
 - d. NPC and hence NP hard.

 11 votes

-- Arjun Suresh (350k points)

1.13.5 P Np Npc Nph: GATE2003-12 top

<https://gateoverflow.in/903>



C. For a problem to be NP-Complete, it must be NP-hard and it must also be in NP.

Ram's reduction shows that Π is NP hard because it must be at least as hard as 3-SAT which is a known NP-Complete problem. Here, Π need not be NP-Complete.

Now, Shyam's reduction shows that 3-SAT problem is at least as hard as Π or equivalently Π is not harder than 3-SAT. Since 3-SAT is in NP, Π must also be in NP.

Thus, Π is NP-hard and also in NP $\implies \Pi$ is NP-Complete.

 10 votes

-- Anurag Semwal (8k points)

1.13.6 P Np Npc Nph: GATE2004-30, ISRO2017-10

<https://gateoverflow.in/1037>



Option is C.

<http://cstheory.stackexchange.com/questions/6864/why-is-2sat-in-p>

 9 votes

-- anshu (3,3k points)

1.13.7 P Np Npc Nph: GATE2006-16, ISRO-DEC2017-27 top



Answer B.

As S is NPC i.e NP -Hard and NP .

- We know that, if *NP-Hard* problem is reducible to another problem in Polynomial Time, then that problem is also *NP-Hard* which means every *NP* problem can be reduced to this problem in Polynomial Time.

Therefore, R is *NP-Hard*.

Now Q is reduced to S in polynomial time.

- If Q is reducible to S in polynomial time, Q could be NP because all NP problems can be reduced to S . Since Q could be NP therefore Q could be P also as P is subset of NP . Also Q could be NPC because every NPC problem can be reduced to another NPC problem in polynomial time.

So, nothing can be concluded about Q .

11 votes

-- Mehak Sharma (1.6k points)

1.13.8 P Np Npc Nph: GATE2008-44 top

<https://gateoverflow.in/456>

Selected Answer

Subset problem is NP-Complete - there is reduction proof but I don't remember (Can see the below link). So, (C) and (D) are true as an NPC problem is in NP as well as NPH.

https://en.wikipedia.org/wiki/Subset_sum_problem

Now, complexity of Q is $O(nW)$, where W is an integer.

- A. Input is encoded in unary. So, length of input is equal to the value of the input. So, complexity $= O(nW)$ where both n and W are linear multiples of the length of the inputs. So, the complexity is polynomial in terms of the input length. So, (A) is true.
- B. Input is encoded in binary. So, length of W will be $\lg W$. (for $W = 1024$, input length will be just 10). So, now W is exponential in terms of the input length of W and $O(nW)$ also becomes exponential in terms of the input lengths. So, Q is not a polynomial time algorithm. So, (B) is false.

17 votes

-- Arjun Suresh (350k points)

1.13.9 P Np Npc Nph: TIFR2010-B-39 top

<https://gateoverflow.in/18754>

Selected Answer

Option E leads to a contradiction, hence is false.

We know that L is NPC, hence $\in \text{NP}$. If $P \neq \text{NP}$, then L can't be in P

6 votes

-- Pragy Agarwal (20.6k points)

1.13.10 P Np Npc Nph: TIFR2011-B-37 top

<https://gateoverflow.in/20922>

Selected Answer

The reverse problem can be solved in polynomial time as a^b requires at most $\log b$ recursive calls using the approach given below:

```
pow(int a, int b)
{
    if(b%2)
        return a * pow(a*a, b/2);
    else
        return pow(a*a, b/2);
}
```

Now, the forward problem is also solvable in polynomial time. We need to check for all the roots of n (from \sqrt{n} till $n^{\frac{1}{\log n}}$) whether it is an integer. But each of these check can be done in $\log n$ time using a binary search on the set of integers from $2..n$ and so, the overall complexity will be $(\log n)^2$ which is polynomial in $\log n$ ($\log n$ is the size of input). So, (a) must be the answer.

6 votes

-- gatecse (18k points)

1.13.11 P Np Npc Nph: TIFR2013-B-7 top

<https://gateoverflow.in/25668>

Selected Answer

I believe Umang is right, option **B** is the correct answer.

Intractability : We are looking for EFFICIENT algorithms.

Intractable Problems are problems that are decidable, although the algorithm to decide that problem might be efficient (P) or inefficient (NP), but at least an algorithm exists to solve these problems.

Here we talk about efficient vs inefficient computations.

Thus the language of problems in P and NP classes is the language of Decidable Problems i.e. Recursive Language.

Undecidability: We are looking for algorithms.

Undecidable Problems are problems for which there is no algorithm to solve these problems.

Here we talk about what can or can not be computed.

The language of Undecidable Problems are "Recursively Enumerable but not recursive languages" & "Not Recursively Enumerable Languages".

Clearly we can talk about the intractability of any problem if we know at least one algorithm to solve the problem, if there is no algorithm to solve a problem how can we talk about efficiency?

Halting Problem is undecidable.

I guess, all other problems mentioned here are decidable.

I don't know the most efficient algorithms to solve these problems but at least I can say that Brute force approach will work on all the other options except the Halting Problem.

What $P = NP$ implies?

"Any problem that is solved by a non deterministic Turing machine in polynomial time also be solved by some deterministic Turing machine in polynomial time, even if the degree of the polynomial is higher."

and

There is neither a Non Deterministic Turing Machine nor Deterministic Turing Machine that can solve the Halting Problem.

So any inference about P & NP is not going to affect the solvability of Halting Problem, since it is undecidable.

10 votes

-- Anurag Pandey (13.6k points)

1.13.12 P Np Npc Nph: TIFR2017-B-2 [top](#)

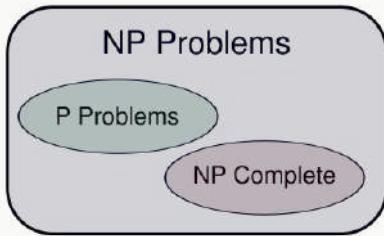
<https://gateoverflow.in/95673>



Selected Answer

E. All of them. Because all of them can be solved by Depth first traversal.

Every P problem is a subset of NP.



👍 9 votes

-- Motamarri Anusha (12.3k points)

1.14

Recurrence(32) [top](#)

1.14.1 Recurrence: GATE1987-10a [top](#)

<https://gateoverflow.in/82450>

Solve the recurrence equations:

- $T(n) = T(n - 1) + n$
- $T(1) = 1$

gate1987 algorithms recurrence

Answer

1.14.2 Recurrence: GATE1988-13iv [top](#)

<https://gateoverflow.in/94637>

Solve the recurrence equations:

- $T(n) = T(\frac{n}{2}) + 1$
- $T(1) = 1$

gate1988 descriptive algorithms recurrence

Answer

1.14.3 Recurrence: GATE1989-13b [top](#)

<https://gateoverflow.in/93175>

Find a solution to the following recurrence equation:

$$T(n) = \sqrt{n} + T\left(\frac{n}{2}\right)$$

$$T(1) = 1$$

gate1989 descriptive algorithms recurrence

Answer

1.14.4 Recurrence: GATE1990-17a [top](#)

<https://gateoverflow.in/86878>

Express $T(n)$ in terms of the harmonic number $H_n = \sum_{t=1}^n 1/t, n \geq 1$, where $T(n)$ satisfies the recurrence relation,

$$T(n) = \frac{n+1}{n} T(n-1) + 1, \text{ for } n \geq 1 \text{ and } T(1) = 1$$

What is the asymptotic behaviour of $T(n)$ as a function of n ?

gate1990 descriptive algorithms recurrence

Answer

1.14.5 Recurrence: GATE1992-07a [top](#)

<https://gateoverflow.in/586>

Consider the function $F(n)$ for which the pseudocode is given below :

```
Function F(n)
begin
F1 ← 1
if(n=1) then F ← 3
else
  For i = 1 to n do
    begin
      C ← 0
      For j = 1 to n - 1 do
        begin C ← C + 1 end
      F1 = F1 * C
    end
  F = F1
end
```

[n is a positive integer greater than zero]

(a) Derive a recurrence relation for $F(n)$

gate1992 algorithms recurrence descriptive

Answer

1.14.6 Recurrence: GATE1992-07b [top](#)

<https://gateoverflow.in/43600>

Consider the function $F(n)$ for which the pseudocode is given below :

```
Function F(n)
begin
F1 ← 1
if(n=1) then F ← 3
else
  For i = 1 to n do
    begin
      C ← 0
      For j = 1 to n - 1 do
        begin C ← C + 1 end
      F1 = F1 * C
    end
  F = F1
end
```

[n is a positive integer greater than zero]

Solve the recurrence relation for a closed form solution of $F(n)$.

[gate1992](#) [algorithms](#) [recurrence](#) [descriptive](#)
Answer

1.14.7 Recurrence: GATE1993-15 [top](#)

<https://gateoverflow.in/2312>

Consider the recursive algorithm given below:

```
procedure bubblesort (n);
var i,j: index; temp : item;
begin
    for i:=1 to n-1 do
        if A[i] > A[i+1] then
            begin
                temp := A[i];
                A[i] := A[i+1];
                A[i+1] := temp;
            end;
    bubblesort (n-1)
end
```

Let a_n be the number of times the 'if...then...' statement gets executed when the algorithm is run with value n . Set up the recurrence relation by defining a_n in terms of a_{n-1} . Solve for a_n .

[gate1993](#) [algorithms](#) [recurrence](#) [normal](#)
Answer

1.14.8 Recurrence: GATE1994-1.7, ISRO2017-14 [top](#)

<https://gateoverflow.in/2444>

The recurrence relation that arises in relation with the complexity of binary search is:

- A. $T(n) = 2T\left(\frac{n}{2}\right) + k$, k is a constant
- B. $T(n) = T\left(\frac{n}{2}\right) + k$, k is a constant
- C. $T(n) = T\left(\frac{n}{2}\right) + \log n$
- D. $T(n) = T\left(\frac{n}{2}\right) + n$

[gate1994](#) [algorithms](#) [recurrence](#) [easy](#) [isro2017](#)
Answer

1.14.9 Recurrence: GATE1996-2.12 [top](#)

<https://gateoverflow.in/2741>

The recurrence relation

- $T(1) = 2$
- $T(n) = 3T\left(\frac{n}{4}\right) + n$

has the solution $T(n)$ equal to

- A. $O(n)$
- B. $O(\log n)$
- C. $O\left(n^{\frac{3}{4}}\right)$

- D. None of the above

gate1996 algorithms recurrence normal

Answer

1.14.10 Recurrence: GATE1997-4.6 [top](#)

<https://gateoverflow.in/2247>

Let $T(n)$ be the function defined by $T(1) = 1$, $T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + \sqrt{n}$ for $n \geq 2$.

Which of the following statements is true?

- A. $T(n) = O\sqrt{n}$
- B. $T(n) = O(n)$
- C. $T(n) = O(\log n)$
- D. None of the above

gate1997 algorithms recurrence normal

Answer

1.14.11 Recurrence: GATE1998-6a [top](#)

<https://gateoverflow.in/44584>

Solve the following recurrence relation

$$x_n = 2x_{n-1} - 1, n > 1$$

$$x_1 = 2$$

gate1998 algorithms recurrence descriptive

Answer

1.14.12 Recurrence: GATE2002-1.3 [top](#)

<https://gateoverflow.in/807>

The solution to the recurrence equation $T(2^k) = 3T(2^{k-1}) + 1, T(1) = 1$ is

- A. 2^k
- B. $\frac{(3^{k+1}-1)}{2}$
- C. $3^{\log_2 k}$
- D. $2^{\log_3 k}$

gate2002 algorithms recurrence normal

Answer

1.14.13 Recurrence: GATE2002-2.11 [top](#)

<https://gateoverflow.in/841>

The running time of the following algorithm

Procedure A(n)

If $n \leq 2$ return (1) else return ($A(\lceil \sqrt{n} \rceil)$);

is best described by

- A. $O(n)$
- B. $O(\log n)$
- C. $O(\log \log n)$
- D. $O(1)$

gate2002 algorithms recurrence normal

Answer

1.14.14 Recurrence: GATE2003-35 [top](#)

<https://gateoverflow.in/925>

Consider the following recurrence relation

$$T(1) = 1$$

$$T(n+1) = T(n) + \lfloor \sqrt{n+1} \rfloor \text{ for all } n \geq 1$$

The value of $T(m^2)$ for $m \geq 1$ is

- A. $\frac{m}{6}(21m - 39) + 4$
- B. $\frac{m}{6}(4m^2 - 3m + 5)$
- C. $\frac{m}{2}(3m^{2.5} - 11m + 20) - 5$
- D. $\frac{m}{6}(5m^3 - 34m^2 + 137m - 104) + \frac{5}{6}$

gate2003 algorithms time-complexity recurrence

Answer

1.14.15 Recurrence: GATE2004-83, ISRO2015-40 [top](#)

<https://gateoverflow.in/1077>

The time complexity of the following C function is (assume $n > 0$)

```
int recursive (int n) {
    if (n == 1)
        return (1);
    else
        return (recursive (n-1) + recursive (n-1));
}
```

- A. $O(n)$
- B. $O(n \log n)$
- C. $O(n^2)$
- D. $O(2^n)$

gate2004 algorithms recurrence time-complexity normal isro2015

Answer

1.14.16 Recurrence: GATE2004-84 [top](#)

<https://gateoverflow.in/1078>

The recurrence equation

$$T(1) = 1$$

$$T(n) = 2T(n-1) + n, n \geq 2$$

evaluates to

- A. $2^{n+1} - n - 2$

- B. $2^n - n$
 C. $2^{n+1} - 2n - 2$
 D. $2^n + n$

gate2004 algorithms recurrence normal

Answer

1.14.17 Recurrence: GATE2004-IT-57 [top](#)

<https://gateoverflow.in/3700>

Consider a list of recursive algorithms and a list of recurrence relations as shown below. Each recurrence relation corresponds to exactly one algorithm and is used to derive the time complexity of the algorithm.

	Recursive Algorithm	Recurrence Relation
P.	Binary search	I. $T(n) = T(n - k) + T(k) + cn$
Q.	Merge sort	II. $T(n) = 2T(n - 1) + 1$
R.	Quick sort	III. $T(n) = 2T(n/2) + cn$
S.	Tower of Hanoi	IV. $T(n) = T(n/2) + 1$

Which of the following is the correct match between the algorithms and their recurrence relations?

- A. P-II, Q-III, R-IV, S-I
 B. P-IV, Q-III, R-I, S-II
 C. P-III, Q-II, R-IV, S-I
 D. P-IV, Q-II, R-I, S-III

gate2004-it algorithms recurrence normal

Answer

1.14.18 Recurrence: GATE2005-IT-51 [top](#)

<https://gateoverflow.in/3812>

Let $T(n)$ be a function defined by the recurrence

$$T(n) = 2T(n/2) + \sqrt{n} \text{ for } n \geq 2 \text{ and} \\ T(1) = 1$$

Which of the following statements is **TRUE**?

- A. $T(n) = \Theta(\log n)$
 B. $T(n) = \Theta(\sqrt{n})$
 C. $T(n) = \Theta(n)$
 D. $T(n) = \Theta(n \log n)$

gate2005-it algorithms recurrence easy

Answer

1.14.19 Recurrence: GATE2006-51, ISRO2016-34 [top](#)

<https://gateoverflow.in/1829>

Consider the following recurrence:

$$T(n) = 2T(\sqrt{n}) + 1, T(1) = 1$$

Which one of the following is true?

- A. $T(n) = \Theta(\log \log n)$
- B. $T(n) = \Theta(\log n)$
- C. $T(n) = \Theta(\sqrt{n})$
- D. $T(n) = \Theta(n)$

[algorithms](#) [recurrence](#) [isro2016](#) [gate2006](#)

[Answer](#)

1.14.20 Recurrence: GATE2008-78 [top](#)

<https://gateoverflow.in/497>

Let x_n denote the number of binary strings of length n that contain no consecutive 0s.

Which of the following recurrences does x_n satisfy?

- A. $x_n = 2x_{n-1}$
- B. $x_n = x_{\lfloor n/2 \rfloor} + 1$
- C. $x_n = x_{\lfloor n/2 \rfloor} + n$
- D. $x_n = x_{n-1} + x_{n-2}$

[gate2008](#) [algorithms](#) [recurrence](#) [normal](#)

[Answer](#)

1.14.21 Recurrence: GATE2008-79 [top](#)

<https://gateoverflow.in/43485>

Let x_n denote the number of binary strings of length n that contain no consecutive 0s.

The value of x_5 is

- A. 5
- B. 7
- C. 8
- D. 16

[gate2008](#) [algorithms](#) [recurrence](#) [normal](#)

[Answer](#)

1.14.22 Recurrence: GATE2008-IT-44 [top](#)

<https://gateoverflow.in/3354>

When $n = 2^{2k}$ for some $k \geq 0$, the recurrence relation

$$T(n) = \sqrt{2}T(n/2) + \sqrt{n}, T(1) = 1$$

evaluates to :

- A. $\sqrt{n}(\log n + 1)$
- B. $\sqrt{n} \log n$

- C. $\sqrt{n} \log \sqrt{n}$
 D. $n \log \sqrt{n}$

gate2008-it | algorithms | recurrence | normal

Answer

1.14.23 Recurrence: GATE2009-35 top

<https://gateoverflow.in/1321>

The running time of an algorithm is represented by the following recurrence relation:

$$T(n) = \begin{cases} n & n \leq 3 \\ T\left(\frac{n}{3}\right) + cn & \text{otherwise} \end{cases}$$

Which one of the following represents the time complexity of the algorithm?

- A. $\Theta(n)$
 B. $\Theta(n \log n)$
 C. $\Theta(n^2)$
 D. $\Theta(n^2 \log n)$

gate2009 | algorithms | recurrence | time-complexity | normal

Answer

1.14.24 Recurrence: GATE2012-16 top

<https://gateoverflow.in/48>

The recurrence relation capturing the optimal execution time of the *Towers of Hanoi* problem with n discs is

- A. $T(n) = 2T(n - 2) + 2$
 B. $T(n) = 2T(n - 1) + n$
 C. $T(n) = 2T(n/2) + 1$
 D. $T(n) = 2T(n - 1) + 1$

gate2012 | algorithms | easy | recurrence

Answer

1.14.25 Recurrence: GATE2014-2-13 top

<https://gateoverflow.in/1968>

Which one of the following correctly determines the solution of the recurrence relation with $T(1) = 1$?

$$T(n) = 2T\left(\frac{n}{2}\right) + \log n$$

- A. $\Theta(n)$
 B. $\Theta(n \log n)$
 C. $\Theta(n^2)$
 D. $\Theta(\log n)$

gate2014-2 | algorithms | recurrence | normal

Answer

1.14.26 Recurrence: GATE2015-1-49 [top](#)

<https://gateoverflow.in/8355>

Let a_n represent the number of bit strings of length n containing two consecutive 1s. What is the recurrence relation for a_n ?

- A. $a_{n-2} + a_{n-1} + 2^{n-2}$
- B. $a_{n-2} + 2a_{n-1} + 2^{n-2}$
- C. $2a_{n-2} + a_{n-1} + 2^{n-2}$
- D. $2a_{n-2} + 2a_{n-1} + 2^{n-2}$

[gate2015-1](#) [algorithms](#) [recurrence](#) [normal](#)

Answer

1.14.27 Recurrence: GATE2015-3-39 [top](#)

<https://gateoverflow.in/8498>

Consider the following recursive C function.

```
void get(int n)
{
    if (n<1) return;
    get (n-1);
    get (n-3);
    printf("%d", n);
}
```

If $get(6)$ function is being called in $main()$ then how many times will the $get()$ function be invoked before returning to the $main()$?

- A. 15
- B. 25
- C. 35
- D. 45

[gate2015-3](#) [algorithms](#) [recurrence](#) [normal](#)

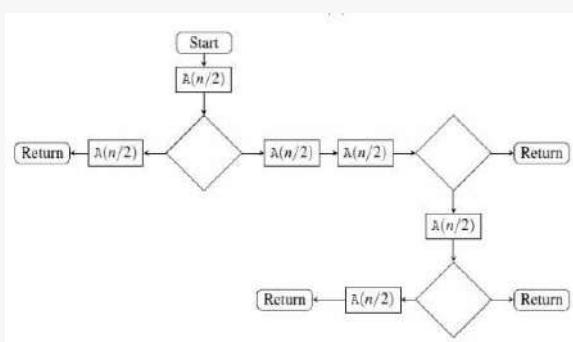
Answer

1.14.28 Recurrence: GATE2016-2-39 [top](#)

<https://gateoverflow.in/39581>

The given diagram shows the flowchart for a recursive function $A(n)$. Assume that all statements, except for the recursive calls, have $O(1)$ time complexity. If the worst case time complexity of this function is $O(n^\alpha)$, then the least possible value (accurate up to two decimal positions) of α is _____.

Flow chart for Recursive Function $A(n)$.



[gate2016-2](#)
[algorithms](#)
[time-complexity](#)
[recurrence](#)
[normal](#)
[numerical-answers](#)
Answer**1.14.29 Recurrence: GATE2017-2-30** [top](#)<https://gateoverflow.in/118623>

Consider the recurrence function

$$T(n) = \begin{cases} 2T(\sqrt{n}) + 1, & n > 2 \\ 2, & 0 < n \leq 2 \end{cases}$$

Then $T(n)$ in terms of θ notation is

- A. $\theta(\log \log n)$
- B. $\theta(\log n)$
- C. $\theta(\sqrt{n})$
- D. $\theta(n)$

[gate2017-2](#)
[algorithms](#)
[recurrence](#)
Answer**1.14.30 Recurrence: TIFR2014-B-11** [top](#)<https://gateoverflow.in/27308>

Consider the following recurrence relation:

$$T(n) = \begin{cases} T\left(\frac{n}{k}\right) + T\left(\frac{3n}{4}\right) + n & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

Which of the following statements is FALSE?

- a. $T(n)$ is $O(n^{3/2})$ when $k = 3$.
- b. $T(n)$ is $O(n \log n)$ when $k = 3$.
- c. $T(n)$ is $O(n \log n)$ when $k = 4$.
- d. $T(n)$ is $O(n \log n)$ when $k = 5$.
- e. $T(n)$ is $O(n)$ when $k = 5$.

[tifr2014](#)
[algorithms](#)
[recurrence](#)
Answer**1.14.31 Recurrence: TIFR2015-B-1** [top](#)<https://gateoverflow.in/29657>

Consider the following recurrence relation:

$$T(n) = \begin{cases} 2T(\lfloor \sqrt{n} \rfloor) + \log n & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

Which of the following statements is TRUE?

- a. $T(n)$ is $O(\log n)$.
- b. $T(n)$ is $O(\log n \cdot \log \log n)$ but not $O(\log n)$.
- c. $T(n)$ is $O(\log^{3/2} n)$ but not $O(\log n \cdot \log \log n)$.
- d. $T(n)$ is $O(\log^2 n)$ but not $O(\log^{3/2} n)$.
- e. $T(n)$ is $O(\log^2 n \cdot \log \log n)$ but not $O(\log^2 n)$.

[tifr2015](#) [algorithms](#) [recurrence](#) [time-complexity](#)

Answer

1.14.32 Recurrence: TIFR2017-A-15 [top](#)

<https://gateoverflow.in/95664>

Let $T(a, b)$ be the function with two arguments (both nonnegative integral powers of 2) defined by the following recurrence:

- $T(a, b) = T\left(\frac{a}{2}, b\right) + T\left(a, \frac{b}{2}\right)$ if $a, b \geq 2$;
- $T(a, 1) = T\left(\frac{a}{2}, 1\right)$ if $a \geq 2$;
- $T(1, b) = T\left(1, \frac{b}{2}\right)$ if $b \geq 2$;
- $T(1, 1) = 1$.

What is $T(2^r, 2^s)$?

- A. rs
 B. $r + s$
 C. $\binom{2^r + 2^s}{2^r}$
 D. $\binom{r + s}{r}$
 E. 2^{r-s} if $r \geq s$, otherwise 2^{s-r}

[tifr2015](#) [algorithms](#) [recurrence](#)

Answer

Answers: Recurrence

1.14.1 Recurrence: GATE1987-10a [top](#)

<https://gateoverflow.in/82450>



Selected Answer

$$\begin{aligned}
 T(n) &= T(n-1) + n \\
 &= T(n-2) + (n-1) + n \\
 &= T(n-3) + (n-2) + (n-1) + n \\
 &\quad \vdots \\
 &\quad \vdots \\
 &= T(n-k) + [(n-k+1) + (n-k+2) + \dots + (n-1) + n]
 \end{aligned}$$

Recurrence stops at,

$$\begin{aligned}
 n - k &= 1 \\
 k &= n - 1
 \end{aligned}$$

$$\therefore T(n) = T(1) + [2 + 3 + \dots + n] = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

PS: Unless explicitly asked for asymptotic bound, we should always try to get the exact answer.

19 votes

-- kirti singh (3.8k points)

1.14.2 Recurrence: GATE1988-13iv top

<https://gateoverflow.in/94637>

Selected Answer

$$T(n) = T(n/2) + 1$$

$$= T(n/4) + 2$$

$$= T(n/8) + 3$$

⋮

$$= T(n/2^k) + k.$$

Recurrence stops when $2^k \geq n$.

When $2^k = n, k = \lg n$

$$\text{So, } T(n) = T(1) + \lg n = 1 + \lg n$$

PS: Unless explicitly asked for asymptotic bound, we should give exact answers for solutions of recurrence equations.

4 votes

-- Arjun Suresh (350k points)

1.14.3 Recurrence: GATE1989-13b top

<https://gateoverflow.in/93175>

$$O(n^{1/2})$$

2 votes

-- Purvi Agrawal (2.9k points)

1.14.4 Recurrence: GATE1990-17a top

<https://gateoverflow.in/86878>

Selected Answer

$$T(n) = \frac{n+1}{n} T(n-1) + 1 \quad \rightarrow (1)$$

$$T(n-1) = \frac{n}{n-1} T(n-2) + 1 \quad \rightarrow (2)$$

$$T(n-2) = \frac{n-1}{n-2} T(n-3) + 1 \quad \rightarrow (3)$$

Substituting value of $T(n-1)$ from (2) in (1)

$$\Rightarrow T(n) = \frac{n+1}{n} * \frac{n}{n-1} T(n-2) + \frac{n+1}{n} + 1$$

$$\Rightarrow T(n) = \frac{n+1}{n-1} T(n-2) + \frac{n+1}{n} + 1$$

Now substituting value of $T(n-2)$ in above equation

$$\Rightarrow T(n) = \frac{n+1}{n-1} * \frac{n-1}{n-2} T(n-3) + \frac{n+1}{n-1} + \frac{n+1}{n} + 1$$

$$\implies T(n) = \frac{n+1}{n-2}T(n-3) + \frac{n+1}{n-1} + \frac{n+1}{n} + 1$$

⋮

so on

$$T(n) = \frac{n+1}{n-k+1}T(n-k) + \frac{n+1}{n} + \frac{n+1}{n-1} + \dots + \frac{n+1}{n-k+2} + 1$$

$$T(n) = \frac{n+1}{n-k+1}T(n-k) + (n+1) * \left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{n-k+2} \right) + 1$$

Now let $n - k = 1$ so $k = n - 1$, substitute value of k in above equation

$$\implies T(n) = \frac{n+1}{n-(n-1)+1}T(1) + (n+1) * \left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{n-(n-1)+2} \right) + 1$$

$$\implies T(n) = \frac{n+1}{2} + (n+1) * \left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{3} \right) + 1$$

$$\implies T(n) = \frac{n+1}{2} + (n+1) * \left(H_n - \frac{1}{2} - 1 \right) + 1$$

$$\implies T(n) = \frac{n+1}{2} + (n+1) * H_n - \frac{n+1}{2} - (n+1) + 1$$

$$\implies \mathbf{T(n)} = (\mathbf{n+1}) \times \mathbf{H_n} - \mathbf{n}$$

Now, $H_n \approx \log n + \gamma$

where γ is the Euler-Mascheroni constant.

$$\mathbf{T(n)} = \mathbf{O(n \log n)}$$

 15 votes

-- Digvijaysingh Gautam (9k points)

1.14.5 Recurrence: GATE1992-07a top

<https://gateoverflow.in/586>



Selected Answer

1. The function $F(n)$ is NOT a recursive function. You can't have a recurrence relation for it in the first place!
2. $F(n)$ calculates $(n-1)^n$.

The equivalent C++ code is as follows: ([You can try it out here: http://ideone.com/w0u4lk](http://ideone.com/w0u4lk))

```
long F(long n) {
    long F1 = 1;

    if(n==1) { return 3; }
    else {
        for(long i = 1; i <= n; i++) {
            long C = 0;
            // Note: the before For loop only has one line
            for(long j = 1; j <= n-1; j++) { C = C+1; }
            // At the end of this for loop, C will be = (n-1)
            F1 = F1 * C;
        }
        return F1;
    }
}
```

It is clear that the inner for loop can be replaced by a single statement as follows:

```
long F(long n) {
    long F1 = 1;

    if(n==1) { return 3; }
    else {
        for(long i = 1; i <= n; i++)
            F1 = F1 * (n-1);
    }
    return F1;
}
```

And this calculates $(n - 1)^n$

17 votes

-- Pragy Agarwal (20.6k points)

1.14.6 Recurrence: GATE1992-07b top

<https://gateoverflow.in/43600>



Selected Answer

```
Function F(n)
begin
F1 ← 1
if(n=1) then F ← 3 //if (n==1) then return 3
else
    For i = 1 to n do
        begin
            C ← 0
            For j = 1 to n - 1 do //inner loop runs n-1 times outer loop runs for n time
                begin C ← C + 1 end //means C=n-1
                F1 = F1 * C //means n-1 is getting multiplied n times so ans is (n-1)^n for
            end
        F = F1
    end
```

10 votes

-- Rajesh Pradhan (22.6k points)

1.14.7 Recurrence: GATE1993-15 top

<https://gateoverflow.in/2312>



Selected Answer

$$a_n = a_{n-1} + n - 1 \quad (n - 1 \text{ comparisons for } n \text{ numbers})$$

$$a_n = a_{n-2} + (n - 2) + (n - 1)$$

$$a_n = a_{n-3} + (n - 3) + (n - 2) + (n - 1)$$

.

.

.

$$a_n = a_{n-n} + (n - n) + (n - (n - 1)) + \dots + (n - 3) + (n - 2) + (n - 1)$$

$$a_n = 0 + 1 + 2 + \dots + (n - 3) + (n - 2) + (n - 1)$$

$$\text{which given } a_n = \frac{(n-1) \times (n)}{2}$$

19 votes

-- Rajarshi Sarkar (34.1k points)

1.14.8 Recurrence: GATE1994-1.7, ISRO2017-14 [top](#)

<https://gateoverflow.in/2444>

Selected Answer

It is **B**. searching for only one half of the list. leading to $T(n/2) + \text{constant time}$ in comparing and finding mid element.

26 votes

-- Gate Keeda (19.6k points)

1.14.9 Recurrence: GATE1996-2.12 [top](#)

<https://gateoverflow.in/2741>

Selected Answer

Answer: **A**

According to Master theorem,

$T(n) = aT(\frac{n}{b}) + f(n)$ can be expressed as:

$$T(n) = [n^{\log_b a}][T(1) + u(n)]$$

where $u(n) = \Theta(h(n))$ where $h(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n}{n^{\log_4 3}} = n^{1-\log_4 3}$ as $h(n) = n^r$ where $r > 0$.

So, $T(n) = [n^{\log_b a}][T(1) + u(n)] = T(n) = [n^{\log_4 3}][T(1) + \Theta(n^{1-\log_4 3})] = \Theta(n^1)$.

14 votes

-- Rajarshi Sarkar (34.1k points)

1.14.10 Recurrence: GATE1997-4.6 [top](#)

<https://gateoverflow.in/2247>

Selected Answer

Answer is **B**.

using master method (case 1)

where $a = 2, b = 2$

$$O(n^{1/2}) < O(n^{\log_b a})$$

$$O(n^{1/2}) < O(n^{\log_2 2})$$

15 votes

-- Ankit Rokde (9k points)

1.14.11 Recurrence: GATE1998-6a [top](#)

<https://gateoverflow.in/44584>

Selected Answer

$$\begin{aligned} T(n) &= 2T(n-1) - 1 \\ &= 2(2T(n-2) - 1) - 1 \\ &= 2^2T(n-2) - 2 - 1 \end{aligned}$$

$$\begin{aligned}
&= 2^2(2T(n-3) - 1) - 2 - 1 \\
&= 2^3T(n-3) - 2^2 - 2 - 1 \\
&\dots \\
&= 2^{n-1}T(n-(n-1)) - (2^{n-2} + 2^{n-3} + \dots + 2^2 + 2 + 1) \\
&= 2^{n-1} \times 2 - \frac{2^{n-1}-1}{2-1} \quad \because T(1) = 2, S_n = \frac{a(r^n-1)}{r-1} \\
&= 2^n - (2^{n-1} - 1) \\
&= 2^{n-1} + 1
\end{aligned}$$

20 votes

-- srestha (87k points)

1.14.12 Recurrence: GATE2002-1.3 top

<https://gateoverflow.in/807>



Selected Answer

Let $x = 2^k$
 $T(x) = 3T\left(\frac{x}{2}\right) + 1$

We can apply Master's theorem case 1 with $a = 3$ and $b = 2$ as $f(x) = 1 = O(x^{\log_2 3 - \epsilon})$, $\epsilon > 0$

So, $T(x) = \Theta(x^{\log_2 3}) = \Theta(2^{k \log_2 3}) = \Theta(2^{\log_2 3^k}) = \Theta(3^k)$

So, only option possible is **B**.

We can also directly solve as follows:

$$\begin{aligned}
T(x) &= 3T\left(\frac{x}{2}\right) + 1 \\
&= 9T\left(\frac{x}{4}\right) + 1 + 3 \\
&\vdots \\
&= 3^{\log_2 2^k} + (1 + 3 + 9 + \dots + 3^{\log_2 2^k - 1}) \\
&\quad (\text{recursion depth is } \log_2 x \text{ and } x = 2^k) \\
&= 3^k + \frac{3^{\log_2 2^k} - 1}{3 - 1} \\
&\quad (\text{Sum to n terms of GP with } a = 1 \text{ and } r = 3) \\
&= 3^k + \frac{3^k - 1}{2} \\
&= \frac{3 \cdot 3^k - 1}{2} \\
&= \frac{3^{k+1} - 1}{2}
\end{aligned}$$

OR

$$\begin{aligned}
T(2^k) &= 3T(2^{k-1}) + 1 \\
&= 3^2T(2^{k-2}) + 1 + 3
\end{aligned}$$

$$\begin{aligned}
 & \vdots \\
 & = 3^k T(2^{k-k}) + (1 + 3 + 9 + \dots + 3^{k-1}) \\
 & \text{(recursion depth is } k) \\
 & = 3^k + \frac{3^{k-1}}{3-1} \\
 & \text{(Sum to } n \text{ terms of GP with } a = 1 \text{ and } r = 3) \\
 & = 3^k + \frac{3^{k-1}}{2} \\
 & = \frac{3 \cdot 3^k - 1}{2} \\
 & = \frac{3^{k+1} - 1}{2}
 \end{aligned}$$

34 votes

-- Arjun Suresh (350k points)

1.14.13 Recurrence: GATE2002-2.11 top

<https://gateoverflow.in/841>



The complexity will be the number of times the recursion happens which is equal to the number of times we can take square root of n recursively, till n becomes 2.

$$T(n) = T(\lceil \sqrt{n} \rceil) + 1$$

$$T(2) = 1$$

$$T(2^2) = T(2) + 1 = 2$$

$$T(2^{2^2}) = T(4) + 1 = 3$$

$$T(2^{2^3}) = T(16) + 1 = 4$$

$$\text{So, } T(n) = \lg \lg n + 1 = O(\log \log n)$$

Answer : Option C

40 votes

-- Arjun Suresh (350k points)

1.14.14 Recurrence: GATE2003-35 top

<https://gateoverflow.in/925>



$$\begin{aligned}
 T(m^2) &= T(m^2 - 1) + \lfloor \sqrt{(m^2)} \rfloor \\
 &= T(m^2 - 2) + \lfloor \sqrt{(m^2 - 1)} \rfloor + \lfloor \sqrt{(m^2)} \rfloor \\
 &= T(m^2 - 3) + \lfloor \sqrt{(m^2 - 2)} \rfloor + \lfloor \sqrt{(m^2 - 1)} \rfloor + \lfloor \sqrt{(m^2)} \rfloor \\
 &\vdots \\
 &= T(1) + \lfloor \sqrt{(2)} \rfloor + \lfloor \sqrt{(3)} \rfloor + \dots + \lfloor \sqrt{(m^2)} \rfloor \\
 &= 3 \times 1 + 5 \times 2 + \dots + (2m - 1) \times (m - 1) + m
 \end{aligned}$$

(We are taking floor of square root of numbers, and between successive square roots number of numbers are in the series 3, 5, 7... like 3 numbers from 1..4, 5 numbers from 5 – 9 and so on).

We can try out options here or solve as shown at end:

Put $m = 5$, $T(25) = 3 \times 1 + 5 \times 2 + 7 \times 3 + 9 \times 4 + 5 = 75$

- A. 59
- B. 75
- C. non-integer
- D. 297.5

So, answer must be **B**.

$$\begin{aligned}
 T(m^2) &= 3 \times 1 + 5 \times 2 + \dots + (2m-1) \times (m-1) + m \\
 &= m + \sum_i 1^{m-1} (2i+1) \cdot (i) \\
 &= m + \sum_i 1^{m-1} 2i^2 + i \\
 &= m + \frac{(m-1)m(2m-1)}{3} + \frac{(m-1)m}{2} \\
 &= \frac{m}{6} (6 + 4m^2 - 2m - 4m + 2 + 3m - 3) \\
 &= \frac{m}{6} (4m^2 - 3m + 5)
 \end{aligned}$$

[Sum of the first n natural numbers = $\frac{n(n+1)}{2}$.

Sum of the squares of first n natural numbers = $\frac{n(n+1)(2n+1)}{6}$.]

19 votes

-- Arjun Suresh (350k points)

1.14.15 Recurrence: GATE2004-83, ISRO2015-40 [top](#)

Selected Answer

Option is D.

```

int recursive (int n) {
    if(n == 1) // takes constant time say 'A' time
        return (1); // takes constant time say 'A' time
    else
        return (recursive (n-1) + recursive (n-1)); //takes
        //T(n-1) + T(n-1) time
}

```

$T(n) = 2T(n-1) + a$ is the recurrence equation found from the pseudo code .

Solving the Recurrence Equation By Back Substitution Method

$$T(n) = 2T(n-1) + a \rightarrow (1)$$

$$T(n-1) = 2T(n-2) + a$$

$$T(n-2) = 2T(n-3) + a$$

We can re-write Equation (1) as

$$\begin{aligned}
 T(n) &= 2[2T(n-2) + a] + a = 4T(n-2) + 3a \\
 &= 2[2T(n-3) + a] + 3a = 8T(n-3) + 7a \\
 &\quad \vdots \\
 &= 2^k T(n-k) + (2^k - 1)a
 \end{aligned} \tag{2}$$

On Substituting Limiting Condition

$$T(1) = 1 \text{ implies } n - k = 1 \implies k = n - 1$$

Therefore, equation (2) becomes:

$$2^{n-1} + (2^{n-1} - 1)a = O(2^n)$$

22 votes

-- pC (22k points)

1.14.16 Recurrence: GATE2004-84 top

<https://gateoverflow.in/1078>



Selected Answer

$$T(n) = 2T(n-1) + n, n \geq 2, T(1) = 1$$

$$\begin{aligned}
 T(n) &= n + 2(n-1) + 2^2(n-2) + \cdots + 2^{(n-1)}(n-(n-1)) \\
 &= n(1+2+\cdots+2^{n-1}) - (1.2+2.2^2+3.2^3+\cdots+(n-1).2^{n-1}) \\
 &= n(2^n - 1) - (n.2^n - 2^{n+1} + 2) \\
 &= 2^{n+1} - n - 2
 \end{aligned}$$

37 votes

-- suraj (5.6k points)

$$T(1) = 1$$

$$T(2) = 4$$

$$T(3) = 11$$

$$T(4) = 26$$

$$T(5) = 57$$

$$T(6) = 120$$

$$T(7) = 247$$

So,

$$T(n) = 2^{n+1} - n - 2$$

44 votes

-- Arjun Suresh (350k points)

1.14.17 Recurrence: GATE2004-IT-57 top

<https://gateoverflow.in/3700>



Selected Answer

Answer is **B.**

8 votes

-- Sankaranarayanan P.N (11.5k points)

1.14.18 Recurrence: GATE2005-IT-51 top <https://gateoverflow.in/3812>



Selected Answer

Option **C** is the answer. It can be done by Master's theorem.

$$n^{\log_b a} = n^{\log_2 2} = n.$$

$$f(n) = \sqrt{n} = n^{\frac{1}{2}}.$$

So, $f(n) = O(n^{\log_b a - \epsilon})$ is true for any real ϵ , $0 < \epsilon < \frac{1}{2}$. Hence Master theorem Case 1 satisfied,

$$T(n) = \Theta(n^{\log_b a}) = \Theta(n).$$

22 votes

-- Bhagirathi Nayak (14.1k points)

1.14.19 Recurrence: GATE2006-51, ISRO2016-34 top <https://gateoverflow.in/1829>



Selected Answer

$$\begin{aligned} T(n) &= 2T\left(n^{\frac{1}{2}}\right) + 1 \\ &= 2\left(2T\left(n^{\frac{1}{2^2}}\right) + 1\right) + 1 \\ &= 4 \times T\left(n^{\frac{1}{2^2}}\right) + 5 \\ &= 8 \times T\left(n^{\frac{1}{2^3}}\right) + 13 \dots \\ &= 2^{(\lg \lg n)} + 2 \times \lg \lg n + 1 \text{ (Proved below)} \\ &= \Theta(\lg n) \end{aligned}$$

$n^{\frac{1}{2^k}} = 2$ (Putting 2 so that we can take log. One more step of recurrence can't change the complexity)

So, answer is **B**, $T(n) = \Theta(\log n)$

33 votes

-- Arjun Suresh (350k points)

1.14.20 Recurrence: GATE2008-78 top <https://gateoverflow.in/497>



Selected Answer

0 1 – 2
 01 10 11 – 3
 010 011 101 110 111 – 5
 0101 0110 0111 1010 1011 1101 1110 1111 – 8

So, $x_n = x_{n-1} + x_{n-2}$ (For all the strings ending in 1, we get two new strings and for all strings ending in 0, we get a new string. So, the new set of strings for $n+1$, will have exactly n strings ending in 1)

$$x_5 = 8 + 5 = 13$$

19 votes

-- Arjun Suresh (350k points)

1.14.21 Recurrence: GATE2008-79 top

<https://gateoverflow.in/43485>



Selected Answer

Number of binary strings of length n that contain no consecutive 0s, following will be the required recurrence relation:

$$T(n) = T(n-1) + T(n-2) \quad n > 2$$

base condition $T(1) = 2$ and $T(2) = 3$

$T(1) = 2$	There will be 2 strings of length 1, i.e 0 & 1
$T(2) = 3$	There will be 3 strings of length 2, i.e. 01, 10, 11
$T(3) = T(1) + T(2) = 2 + 3 = 5$	
$T(4) = T(3) + T(2) = 5 + 3 = 8$	
$T(5) = T(4) + T(3) = 8 + 5 = 13$	

Hence, answer is **13**, but no option matches!

13 votes

-- Vijay Thakur (17.1k points)

1.14.22 Recurrence: GATE2008-IT-44 top

<https://gateoverflow.in/3354>



Selected Answer

$$\begin{aligned} T(n) &= \sqrt{2}T\left(\frac{n}{2}\right) + \sqrt{n} \\ &= \sqrt{2}^2 T\left(\frac{n}{2^2}\right) + \sqrt{2}\sqrt{\frac{n}{2}} + \sqrt{n} \\ &\vdots \\ &= \sqrt{2}^{\lg n} T(1) + \lg n \sqrt{n} \\ &= \sqrt{n} + \lg n \sqrt{n} \\ &= \sqrt{n} (\lg n + 1) \end{aligned}$$

If we use Master theorem we get option **B**. But one must know that Master theorem is used to find the asymptotic bound and not an EXACT value. And in the question here it explicitly says "**evaluates to**".

40 votes

-- Arjun Suresh (350k points)

1.14.23 Recurrence: GATE2009-35 top

<https://gateoverflow.in/1321>



Selected Answer

$$a = 1, b = 3, \log_b a = 0$$

$$\text{So } n^{\log_b a} = n^0 = 1$$

$$f(n) = n$$

$$\text{So, } f(n) = \Omega(1)$$

To, check Master theorem case 3, we need $c > 0$,

$$f(n/3) \leq c f(n)$$

$$c = 1$$

So using case three of master theorem

$$T(n) = \Theta(f(n)) = \Theta(n)$$

answer is **A**.

19 votes

-- Pooja Palod (31.3k points)

1.14.24 Recurrence: GATE2012-16 top

<https://gateoverflow.in/48>



Selected Answer

Recurrence relation for **Towers of Hanoi** is

$$T(1) = 1$$

$$T(n) = 2T(n - 1) + 1$$

So Answer should be **(D)**

25 votes

-- Narayan Kunal (421 points)

1.14.25 Recurrence: GATE2014-2-13 top

<https://gateoverflow.in/1968>



Selected Answer

$$f(n) = \log n$$

$$a = 2, b = 2 \implies n^{\log_b a} = n$$

So, $f(n) = \log n = O(n^{1-\epsilon})$, we can take any ϵ from 0-1 for example 0.5 which gives $\log n = O(\sqrt{n})$, whose proof is given

here: <http://math.stackexchange.com/questions/145739/prove-that-logn-o-sqrtn>

So, Master theorem Case 1, and answer will be $O(n^{\log_2 2}) = O(n)$

Alternate way:

$$T(1) = 1T(2) = 2T(1) + \log 2 = 3 = 3n - 2T(4) = 2T(2) + \log 4 = 8 = 3n - 4T(8) = 2T(4)$$

The second term being subtracted is growing at a lower rate than the first term. So, we can say $T(n) = O(n)$.

Upvotes: 28 votes

-- Arjun Suresh (350k points)

1.14.26 Recurrence: GATE2015-1-49 top

<https://gateoverflow.in/8355>



Counting the number of bit strings NOT containing two consecutive 1's. (It is easy to derive a recurrence relation for the NOT case as shown below)

0	1							
00	01	10	- 3 (append both 0 and 1 to any string ending in 0, and append 0 to any string ending in 1)					
000	001	010	100	101	- 5 (all strings ending in 0 give two strings and those ending in 1 give 1 string)			
0000	0001	0010	0100	0101	1000	1001	1010	- 8
:	:							

$a'_n = a'_{n-1} + a'_{n-2}$ (where a_n denote the number of bit strings of length n containing two consecutive 1s)

$$2^n - a_n = (2^{n-1} - a_{n-1}) + (2^{n-2} - a_{n-2})$$

$$a_n = 2^{n-2}(4 - 2 - 1) + a_{n-1} + a_{n-2}$$

$$a_n = a_{n-1} + a_{n-2} + 2^{n-2}$$

A is choice.

Upvotes: 39 votes

-- Arjun Suresh (350k points)

1.14.27 Recurrence: GATE2015-3-39 top

<https://gateoverflow.in/8498>



Answer : Option B

$T(n) = T(n-1) + T(n-3) + 2$, here $T(n)$ denotes the number of times a recursive call is made for input n . 2 denotes the two direct recursive calls.

$T(n \leq 0) = 0$
$T(1) = 2$
$T(2) = 4$
$T(3) = 6$
$T(4) = 10$
$T(5) = 16$
$T(6) = 24$

So, answer is $24 + 1$ call from main = 25.

41 votes

-- Arjun Suresh (350k points)

1.14.28 Recurrence: GATE2016-2-39 top<https://gateoverflow.in/39581>

Selected Answer

If they are asking for worst case complexity hence,
By calling $A(n)$ we get $A(n/2)$ 5 times,

$$A(n) = 5A(n/2) + O(1)$$

Hence, by applying masters theorem,
Case 1 : $a > b^k$

$$n^{\log_2 5}$$

Thus value of alpha will be 2.32

51 votes

-- Shashank Chavan (3.3k points)

1.14.29 Recurrence: GATE2017-2-30 top<https://gateoverflow.in/118623>

Selected Answer

$$T(n) = 2T(\sqrt{n}) + 1$$

$$\text{Put, } n = 2^m$$

$$T(2^m) = 2T(2^{m/2}) + 1$$

$$\text{put, } T(2^m) = s(m)$$

$$s(m) = 2s(m/2) + 1$$

Using case 1 of master method ,

$$= \Theta(m) = \Theta(\log n)$$

<https://gateoverflow.in/1829/gate2006-51-isro2016-34?show=37791#c37791>

26 votes

-- jatin saini (4k points)

1.14.30 Recurrence: TIFR2014-B-11 top<https://gateoverflow.in/27308>

We have to analyze it at three points $k = 3, 4, 5$.

At $k = 4$ it is standard quick sort kind of recurrence relation. So its complexity is $O(n \times \log_4 n)$

Now if $k > 4$ then complexity will come less than $O(n \times \log_4 n)$ and vice versa for $k < 4$.

So option c, d are correct but b is false.

Although partial answer is achieved but do not know how part (a) and (e) could be investigated.

Some people are taking about Akra-Bazzi method. Some ex. of this method is mentioned in following PDF.

<http://www3.cs.stonybrook.edu/~rezaul/Fall-2012/CSE548/CSE548-lectures-7-8.pdf>

<https://math.stackexchange.com/questions/215984/solve-the-relation-tn-tn-4t3n-4n>

1 3 votes

-- Chhotu Ram Chauhan (10.5k points)

1.14.31 Recurrence: TIFR2015-B-1 top

<https://gateoverflow.in/29657>



Selected Answer

Let $n = 2^k$

$$T(2^k) = 2T(2^{k/2}) + k$$

$$\text{Let } T(2^k) = S(k) \implies T(2^{k/2}) = S(k/2)$$

$$S(k) = 2S(k/2) + k$$

This gives $S(k) = \Theta(k \log k)$ (Master theorem case 2)

$$= \Theta(\log n \log \log n)$$

So ans is b

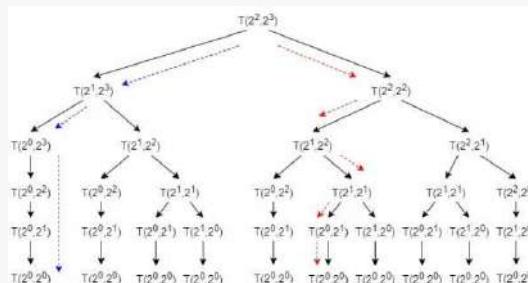
1 12 votes

-- Pooja Palod (31.3k points)

1.14.32 Recurrence: TIFR2017-A-15 top

<https://gateoverflow.in/95664>

Value of $T(2^r, 2^s)$ is nothing but the number of leaf nodes in its recursion tree. The reason for this is that there is only one base condition $T(1, 1)$ which returns 1 which are all ultimately added. Here is an example recursion tree for $T(2^2, 2^3)$.



We will try to use some combinatorial technique to count the number of leaf nodes. One way of counting the number of leaf nodes in the recurrence tree is by counting all possible paths that one can take from the root to the leaf node. The reason for doing this would be clear in a moment. This idea would be used to establish a link with a very common combinatorial problem.

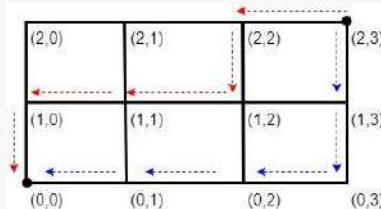
Claim: If we consider a grid of size $r \times s$ then the possible paths from (r, s) to $(0, 0)$ in that grid such that we can either move backwards or downward is nothing but all possible paths in recurrence tree from the root node to leaf node.

This happens to be true because arguments of recurrence relation are in powers of two, and at

each level of recurrence tree, the power of one of the arguments get reduced by one.

$$\begin{aligned} T(2^r, 2^s) &= T(2^{r-1}, 2^s) + T(2^r, 2^{s-1}) \\ T(1, 2^s) &= T(1, 2^{s-1}) \\ T(2^r, 1) &= T(2^{r-1}, 1) \\ T(1, 1) &= 1 \end{aligned}$$

Continuing from our previous example here is a grid of size 2×3 with two valid paths highlighted.



The number of valid paths possible in this grid will give us the count of the number of leaf nodes in the recurrence tree. Interestingly this problem, when interpreted exactly in opposite way, happens to be a very well known problem of combinatorics.

In a grid of size $r \times s$, count the number of possible paths from $(0, 0)$ to (r, s) such that we can either move forward(F) or upward(U).

Video:

So there are $\binom{r+s}{r}$ many different valid paths on this grid to reach (r, s) from $(0, 0)$ or vice versa. And in turn, there are same numbers of leaf nodes in the recursion tree. Therefore, we can conclude with -

$$T(2^r, 2^s) = \binom{r+s}{r}$$

Video:

4 votes

-- Prateek Dwivedi (4.4k points)

1.15

Searching(8) top

1.15.1 Searching: GATE1996-18 top

<https://gateoverflow.in/2770>

Consider the following program that attempts to locate an element x in an array $a[]$ using binary search. Assume $N > 1$. The program is erroneous. Under what conditions does the program fail?

```

var i,j,k: integer; x: integer;
a: array[1..N] of integer;
begin i:= 1; j:= n;
repeat
  k:=(i+j) div 2;
  if a[k] < x then i:= k
  else j:= k
until (a[k] = x) or (i >= j);

if (a[k] = x) then
  writeln ('x is in the array')
else
  writeln ('x is not in the array')
end;

```

[gate1996](#) [algorithms](#) [searching](#) [normal](#)
Answer

1.15.2 Searching: GATE1996-2.13, ISRO2016-28 [top](#)

<https://gateoverflow.in/2742>

The average number of key comparisons required for a successful search for sequential search on n items is

- A. $\frac{n}{2}$
- B. $\frac{n-1}{2}$
- C. $\frac{n+1}{2}$
- D. None of the above

[gate1996](#) [algorithms](#) [easy](#) [isro2016](#) [searching](#)
Answer

1.15.3 Searching: GATE2002-2.10 [top](#)

<https://gateoverflow.in/840>

Consider the following algorithm for searching for a given number x in an unsorted array $A[1..n]$ having n distinct values:

1. Choose an i at random from $1..n$
2. If $A[i] = x$, then Stop else Goto 1;

Assuming that x is present in A , what is the expected number of comparisons made by the algorithm before it terminates?

- A. n
- B. $n - 1$
- C. $2n$
- D. $\frac{n}{2}$

[gate2002](#) [searching](#) [normal](#)
Answer

1.15.4 Searching: GATE2008-84 [top](#)

<https://gateoverflow.in/394>

Consider the following C program that attempts to locate an element x in an array $Y[]$ using binary search. The program is erroneous.

```
f (int Y[10] , int x) {
    int u, j, k;
    i= 0; j = 9;
    do {
        k = (i+ j) / 2;
        if( Y[k] < x) i = k;else j = k;
    } while (Y[k] != x) && (i < j) ;
    if(Y[k] == x) printf(" x is in the array " );
    else printf(" x is not in the array " );
}
```

On which of the following contents of Y and x does the program fail?

- A. Y is $[1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10]$ and $x < 10$
- B. Y is $[1\ 3\ 5\ 7\ 9\ 11\ 13\ 15\ 17\ 19]$ and $x < 1$

- C. Y is $[2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2]$ and $x > 2$
D. Y is $[2\ 4\ 6\ 8\ 10\ 12\ 14\ 16\ 18\ 20]$ and $2 < x < 20$ and x is even

gate2008 algorithms searching normal

Answer

1.15.5 Searching: GATE2008-85 [top](#)

<https://gateoverflow.in/43508>

Consider the following C program that attempts to locate an element x in an array $Y[]$ using binary search. The program is erroneous.

```
f (int Y[10] , int x) {
    int u, j, k;
    i= 0; j = 9;
    do {
        k = (i+ j) / 2;
        if( Y[k] < x) i = k;else j = k;
        } while (Y[k] != x) && (i < j) ;
    if(Y[k] == x) printf(" x is in the array " ) ;
    else printf(" x is not in the array " ) ;
}
```

The correction needed in the program to make it work properly is

- A. Change line 6 to: if ($Y[k] < x$) $i = k + 1$; else $j = k - 1$;
B. Change line 6 to: if ($Y[k] < x$) $i = k - 1$; else $j = k + 1$;
C. Change line 6 to: if ($Y[k] < x$) $i = k$; else $j = k$;
D. Change line 7 to: } while (($Y[k] == x$)&&($i < j$));

gate2008 algorithms searching normal

Answer

1.15.6 Searching: GATE2017-1-48 [top](#)

<https://gateoverflow.in/118331>

Let A be an array of 31 numbers consisting of a sequence of 0's followed by a sequence of 1's. The problem is to find the smallest index i such that $A[i]$ is 1 by probing the minimum number of locations in A . The worst case number of probes performed by an optimal algorithm is _____.

gate2017-1 algorithms normal numerical-answers searching

Answer

1.15.7 Searching: TIFR2010-B-29 [top](#)

<https://gateoverflow.in/18752>

Suppose you are given an array A with $2n$ numbers.

The numbers in odd positions are sorted in ascending order, that is, $A[1] \leq A[3] \leq \dots \leq A[2n - 1]$.

The numbers in even positions are sorted in descending order, that is, $A[2] \geq A[4] \geq \dots \geq A[2n]$.

What is the method you would recommend for determining if a given number is in the array?

- A. Sort the array using quick-sort and then use binary search.
B. Merge the sorted lists and perform binary search.
C. Perform a single binary search on the entire array.

- D. Perform separate binary searches on the odd positions and the even positions.
 E. Search sequentially from the end of the array.

tifr2010 | searching

Answer

1.15.8 Searching: TIFR2012-B-11 [top](#)

<https://gateoverflow.in/25140>

Consider the following three version of the binary search program. Assume that the elements of type T can be compared with each other; also assume that the array is sorted.

```
i, j, k : integer;
a : array [1.....N] of T;
x : T;

Program 1 : i := 1; j := N;
            repeat
                k := (i + j) div 2;
                if a[k] < x then i := k else j := k
            until (a[k] = x) or (i > j)
Program 2 : i := 1; j := N;
            repeat
                k := (i + j) div 2;
                if x < a[k] then j := k - 1;
                if a[k] < x then i := k + 1;
            until i > j
Program 3 := i := 1; j := N
            repeat
                k := (i + j) div 2;
                if x < a[k] then j := k else i := k + 1
            until i > j
```

A binary search program is called correct provided it terminates with $a[k] = x$ whenever such an element exists, or it terminates with $a[k] \neq x$ if there exists no array element with value x . Which of the following statements is correct?

- A. Only Program 1 is correct
- B. Only Program 2 is correct
- C. Only Programs 1 and 2 are correct.
- D. Both Programs 2 and 3 are correct
- E. All the three programs are wrong

tifr2012 | algorithms | searching

Answer

Answers: Searching

1.15.1 Searching: GATE1996-18 [top](#)

<https://gateoverflow.in/2770>



The code is wrong here

```
k=(i+j) / 2;
if (a[k] < x) then i = k;
else j = k;
```

The (correct) code should be:

```
k=(i+j) / 2;
if (a[k] < x) then i = k + 1;
else j = k - 1;
```

We can try an example with the given code in question

Let the array be $a[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

Index numbers $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

Let $x = 10$; now run the code;

Initially $i = 1, j = 10$;

first time $k = (i + j)/2 = 11/2 = 5.5 = 5$ (because of integer type) $= i$

second time $k = (i + j)/2 = 15/2 = 7.5 = 7 = i$

third time $k = (i + j)/2 = 17/2 = 8.5 = 8 = i$

fourth time $k = (i + j)/2 = 18/2 = 9 = i$

fifth time $k = (i + j)/2 = 19/2 = 9.5 = 9 = i$

sixth time $k = (i + j)/2 = 19/2 = 9.5 = 9 = i$

seventh time $k = (i + j)/2 = 19/2 = 9.5 = 9 = i$

Going to infinite loop (run time error)

For terminating the loop, it should be $i = k + 1$ instead of $i = k$ and $j = k - 1$ instead of $j = k$;

18 votes

-- Mithlesh Upadhyay (5.8k points)

1.15.2 Searching: GATE1996-2.13, ISRO2016-28 top

<https://gateoverflow.in/2742>



Selected Answer

Expected number of comparisons

$= 1 \times \text{Probability of first element being } x + 2 \times \text{Probability of second element being } x + \dots + n \times \text{Probability of last element being } x.$

$$= \frac{1}{n} + \frac{2}{n} + \frac{3}{n} + \dots + \frac{n}{n}$$

$$= \frac{\left(\frac{n(n+1)}{2}\right)}{n}$$

$$= \frac{n+1}{2}$$

56 votes

-- Arjun Suresh (350k points)

1.15.3 Searching: GATE2002-2.10 top

<https://gateoverflow.in/840>



Selected Answer

Expected number of comparisons (E) = $1 \times$ Probability of find on first comparison + $2 \times$ Probability of find on second comparison + ... + $i \times$ Probability of find on i^{th} comparison + ...

$$= 1 \times \frac{1}{n} + 2 \times \frac{n-1}{n^2} + 3 \times \frac{(n-1)^2}{n^3} + \dots$$

$$= \frac{1/n}{1 - \frac{n-1}{n}} + \frac{(n-1)/n^2}{\left(1 - \frac{n-1}{n}\right)^2} \left(\text{Sum to infinity of aritmetico-geometric series with } a = \frac{1}{n}, r = \frac{n-1}{n} \text{ and } n > 1\right)$$

Reference: https://en.wikipedia.org/wiki/Arithmetico-geometric_sequence

Or we can also do,

$$E = 1 \times \frac{1}{n} + 2 \times \frac{n-1}{n^2} + 3 \times \frac{(n-1)^2}{n^3} + \dots$$

$$E \frac{n-1}{n} = \frac{n-1}{n^2} + 2 \times \frac{(n-1)^2}{n^3} + 3 \times \frac{(n-1)^3}{n^4} + \dots$$

$$E - E \frac{n-1}{n} = \frac{1}{n} + \frac{n-1}{n^2} + \frac{(n-1)^2}{n^3} + \dots$$

$$E \cdot \frac{1}{n} = \frac{(1/n)}{1 - \frac{n-1}{n}} = 1 \left(\text{Sum to infinity of GP with } a = \frac{1}{n} \text{ and } r = \frac{n-1}{n}\right) \implies E = n$$

41 votes

-- Arjun Suresh (350k points)

1.15.4 Searching: GATE2008-84 [top](#)

<https://gateoverflow.in/394>



Selected Answer

for Q.84

when it is **option C** the control will continue to iterate as $i = 8$ and $j = 9$; again and again i will be assigned k which itself equals 8 as $\frac{8+9}{2}$ being stored in an integer type variable, will evaluate to 8.

For option A, with $x = 9$, k will take the following values:

- 4
- 6
- 7
- $8 - y[8] = 9$, x found

For option D, with $x = 10$, k will take the following values:

- $4, y[4] = 10$, x found

26 votes

-- Amar Vashishth (30.5k points)

1.15.5 Searching: GATE2008-85 [top](#)

<https://gateoverflow.in/43508>



Selected Answer

Answer should be **A**.

```
if( Y[k] < x) then i = k + 1;
```

if given element that we are searching is greater then searching will be continued upper half of array

otherwise
j = k - 1;

lower half.

Take few case in consideration i.e.

1. All elements are same
2. Increasing order with no repetition
3. Increasing order with repetition.

20 votes

-- Manoj Kumar (38.6k points)

1.15.6 Searching: GATE2017-1-48 top

<https://gateoverflow.in/118331>



Selected Answer

Here, since 0s are followed by 1s so we have a sorted sequence and we can apply binary search.

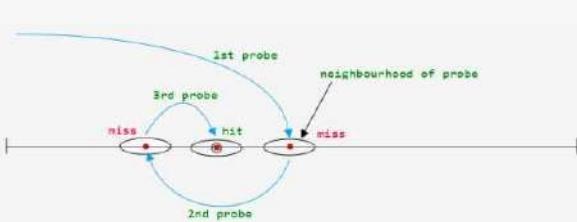
At each stage we compare with $\frac{(low+high)}{2}$ th element index and if it is 1 we check left and if it is 0 we check right.

Total worst case no. of probes is $\lceil \log_2 31 \rceil = 5$.

So, answer is 5.

36 votes

-- sriv_shubham (3.3k points)



it should not take more than **5 probes**

```
#include <bits/stdc++.h>
using namespace std;
#define N 100
void display(bool a[]) {
    int i=0;
    while(i<31) printf("%2d ",a[i++]);
}
void assign(bool a[],int zeros) {
    for(int i=0;i<31;i++)
        (i<zeros)?a[i] = 0:a[i] = 1;
}
int main() {
    srand(time(NULL));
    bool a[31];
    // header
    for(int i=0;i<31;i++) printf("%2d ",i);
```

```

printf("\n\n");
int max_probes = 0;
for(int iteration = 1; iteration <= N; iteration++) {
    int zeros = rand() % 32;
    assign(a, zeros);
    sort(a, a+31);
    int low, high, mid, ans;
    std::vector<int> seq;

    low = 0;
    high = 31;
    while(low < high) {
        mid = low + floor((high - low) / 2);
        seq.push_back(mid);
        if(a[mid] == 0) {
            low = mid + 1;
            ans = low;
            if(a[mid + 1] == 1) break;
        } else {
            high = mid;
            ans = high;
            if(mid > 0)
                if(a[mid - 1] == 0) break;
        }
    }

    display(a);
    printf(" | probes=%d ", seq.size());
    for(auto e:seq) printf("%d ", e);
    printf(" | at = %d\n", ans);
    //if(ans == 15) printf("\nHHH=====\\n");
    max_probes = max(max_probes, (int)(seq.size()));
    seq.clear();
}
printf("%d\n", max_probes);
}

```

20 votes

-- Debashish Deka (56.7k points)

1.15.7 Searching: TIFR2010-B-29 [top](#)

<https://gateoverflow.in/18752>



Selected Answer

Option D is the correct answer.

We can simply use clever indexing to binary search the element in the odd positions, and in the even positions separately.

This will take $O(\log n)$ time and $O(1)$ space in the worst case.

- A: Sorting using Quicksort will take $O(n^2)$ time.
- B: Merging will take $O(n)$ time and $O(n)$ space.
- C: Binary search only works on a sorted array.
- E: Sequential search will take $O(n)$ time.**

20 votes

-- Pragy Agarwal (20.6k points)

1.15.8 Searching: TIFR2012-B-11 [top](#)

<https://gateoverflow.in/25140>



First program wont work if array has elements same..it may go into infinite loop .To make it work it properly we have to do following changes $j = k - 1$ and $i = k + 1$

For second program $a[k] == x$ condition is missing so it is wrong

Third program is also wrong as $j! = k - 1$ and condition $a[k] == x$ is missing

So, answer is **E**.

12 votes

-- Pooja Palod (31.3k points)

1.16

Sorting(55) top

1.16.1 Sorting: CMI2011-B-06a top

<https://gateoverflow.in/46212>

Consider a plate stacked with several disks, each of a different diameter (they could all be, for instance, *dosas* or *chapatis* of different sizes). We want to sort these disks in decreasing order according to their diameter so that the widest disk is at the bottom of the pile. The only operation available for manipulating the disks is to pick up a stack of them from the top of the pile and invert that stack. (This corresponds to lifting up a stack *dosas* or *chapatis* between two big spoons and flipping the stack.)

- Give an algorithm for sorting the disks using this operation.

[cmi2011](#) [descriptive](#) [algorithms](#) [sorting](#)

Answer

1.16.2 Sorting: CMI2013-A-05 top

<https://gateoverflow.in/46595>

You have n lists, each consisting of m integers sorted in ascending order. Merging these lists into a single sorted list will take time:

- A. $O(nm \log m)$
- B. $O(mn \log n)$
- C. $O(m + n)$
- D. $O(mn)$

[cmi2013](#) [algorithms](#) [sorting](#)

Answer

1.16.3 Sorting: CMI2017-A-08 top

<https://gateoverflow.in/203281>

A stable sort preserves the order of values that are equal with respect to the comparison function. We have a list of three dimensional points

$[(7, 1, 8), (3, 5, 7), (6, 1, 4), (6, 5, 9), (0, 2, 5), (9, 0, 9)]$.

We sort these in ascending order by the second coordinate. Which of the following corresponds to a stable sort of this input?

- A. $[(9, 0, 9), (7, 1, 8), (6, 1, 4), (0, 2, 5), (6, 5, 9), (3, 5, 7)]$

- B. $[(0, 2, 5), (3, 5, 7), (6, 1, 4), (6, 5, 9), (7, 1, 8), (9, 0, 9)]$
 C. $[(9, 0, 9), (7, 1, 8), (6, 1, 4), (0, 2, 5), (3, 5, 7), (6, 5, 9)]$
 D. $[(9, 0, 9), (6, 1, 4), (7, 1, 8), (0, 2, 5), (3, 5, 7), (6, 5, 9)]$

cmi2017 | algorithms | sorting

Answer

1.16.4 Sorting: GATE1987-1-xviii top

<https://gateoverflow.in/80366>

Let P be a quicksort program to sort numbers in ascending order. Let t_1 and t_2 be the time taken by the program for the inputs $[1 \ 2 \ 3 \ 4]$ and $[5 \ 4 \ 3 \ 2 \ 1]$, respectively. Which of the following holds?

- A. $t_1 = t_2$
 B. $t_1 > t_2$
 C. $t_1 < t_2$
 D. $t_1 = t_2 + 5 \log 5$

gate1987 | algorithms | sorting

Answer

1.16.5 Sorting: GATE1988-1iii top

<https://gateoverflow.in/91338>

Quicksort is _____ efficient than heapsort in the worst case.

gate1988 | algorithms | sorting

Answer

1.16.6 Sorting: GATE1991-01,vii top

<https://gateoverflow.in/505>

The minimum number of comparisons required to sort 5 elements is _____

gate1991 | normal | algorithms | sorting

Answer

1.16.7 Sorting: GATE1991-13 top

<https://gateoverflow.in/540>

Give an optimal algorithm in pseudo-code for sorting a sequence of n numbers which has only k distinct numbers (k is not known a Priori). Give a brief analysis for the time-complexity of your algorithm.

gate1991 | sorting | time-complexity | algorithms | difficult

Answer

1.16.8 Sorting: GATE1992-02,ix top

<https://gateoverflow.in/559>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Following algorithm(s) can be used to sort n in the range $[1 \dots n^3]$ in $O(n)$ time

- a. Heap sort
- b. Quick sort
- c. Merge sort
- d. Radix sort

gate1992 | easy | algorithms | sorting

Answer

1.16.9 Sorting: GATE1992-03,iv [top](#)<https://gateoverflow.in/581>

Assume that the last element of the set is used as partition element in Quicksort. If n distinct elements from the set $[1 \dots n]$ are to be sorted, give an input for which Quicksort takes maximum time.

gate1992 algorithms sorting easy

Answer**1.16.10 Sorting: GATE1994-1.19, ISRO2016-31** [top](#)<https://gateoverflow.in/2462>

Algorithm design technique used in quicksort algorithm is?

- A. Dynamic programming
- B. Backtracking
- C. Divide and conquer
- D. Greedy method

gate1994 algorithms sorting easy isro2016

Answer**1.16.11 Sorting: GATE1995-1.16** [top](#)<https://gateoverflow.in/2603>

For merging two sorted lists of sizes m and n into a sorted list of size $m + n$, we require comparisons of

- A. $O(m)$
- B. $O(n)$
- C. $O(m + n)$
- D. $O(\log m + \log n)$

gate1995 algorithms sorting normal

Answer**1.16.12 Sorting: GATE1995-1.5** [top](#)<https://gateoverflow.in/2592>

Merge sort uses:

- A. Divide and conquer strategy
- B. Backtracking approach
- C. Heuristic search
- D. Greedy approach

gate1995 algorithms sorting easy

Answer**1.16.13 Sorting: GATE1995-12** [top](#)<https://gateoverflow.in/2648>

Consider the following sequence of numbers:

92, 37, 52, 12, 11, 25

Use bubblesort to arrange the sequence in ascending order. Give the sequence at the end of each of the first five passes.

gate1995 algorithms sorting easy

Answer

1.16.14 Sorting: GATE1996-14 top

<https://gateoverflow.in/2766>

A two dimensional array $A[1..n][1..n]$ of integers is partially sorted if $\forall i, j \in [1..n-1], A[i][j] < A[i][j+1]$ and $A[i][j] < A[i+1][j]$

Fill in the blanks:

- The smallest item in the array is at $A[i][j]$ where $i = \underline{\hspace{2cm}}$ and $j = \underline{\hspace{2cm}}$.
- The smallest item is deleted. Complete the following $O(n)$ procedure to insert item x (which is guaranteed to be smaller than any item in the last row or column) still keeping A partially sorted.

```
procedure insert (x: integer);
var i,j: integer;
begin
  i:=1; j:=1, A[i][j]:=x;
  while (x > A[i+1][j] or x > A[i][j+1]) do
    if A[i+1][j] < A[i][j] then begin
      A[i][j]:=A[i+1][j]; i:=i+1;
    end
    else begin
      end
    A[i][j]:=x;
end
```

gate1996 algorithms sorting normal

Answer

1.16.15 Sorting: GATE1996-2.15 top

<https://gateoverflow.in/2744>

Quick-sort is run on two inputs shown below to sort in ascending order taking first element as pivot

- $1, 2, 3, \dots, n$
- $n, n-1, n-2, \dots, 2, 1$

Let C_1 and C_2 be the number of comparisons made for the inputs (i) and (ii) respectively. Then,

- A. $C_1 < C_2$
- B. $C_1 > C_2$
- C. $C_1 = C_2$
- D. we cannot say anything for arbitrary n

gate1996 algorithms sorting normal

Answer

1.16.16 Sorting: GATE1998-1.22 [top](#)<https://gateoverflow.in/1659>

Give the correct matching for the following pairs:

- | | |
|-------------------|--------------------|
| (A) $O(\log n)$ | (P) Selection sort |
| (B) $O(n)$ | (Q) Insertion sort |
| (C) $O(n \log n)$ | (R) Binary search |
| (D) $O(n^2)$ | (S) Merge sort |

- A. A-R B-P C-Q D-S
- B. A-R B-P C-S D-Q
- C. A-P B-R C-S D-Q
- D. A-P B-S C-R D-Q

[gate1998](#) [algorithms](#) [sorting](#) [easy](#)

[Answer](#)

1.16.17 Sorting: GATE1999-1.12 [top](#)<https://gateoverflow.in/1465>

A sorting technique is called stable if

- A. it takes $O(n \log n)$ time
- B. it maintains the relative order of occurrence of non-distinct elements
- C. it uses divide and conquer paradigm
- D. it takes $O(n)$ space

[gate1999](#) [algorithms](#) [sorting](#) [easy](#)

[Answer](#)

1.16.18 Sorting: GATE1999-1.14, ISRO2015-42 [top](#)<https://gateoverflow.in/1467>

If one uses straight two-way merge sort algorithm to sort the following elements in ascending order:

20, 47, 15, 8, 9, 4, 40, 30, 12, 17

then the order of these elements after second pass of the algorithm is:

- A. 8, 9, 15, 20, 47, 4, 12, 17, 30, 40
- B. 8, 15, 20, 47, 4, 9, 30, 40, 12, 17
- C. 15, 20, 47, 4, 8, 9, 12, 30, 40, 17
- D. 4, 8, 9, 15, 20, 47, 12, 17, 30, 40

[gate1999](#) [algorithms](#) [sorting](#) [normal](#) [isro2015](#)
Answer

1.16.19 Sorting: GATE1999-8 [top](#)

<https://gateoverflow.in/1507>

Let A be an $n \times n$ matrix such that the elements in each row and each column are arranged in ascending order. Draw a decision tree, which finds 1st, 2nd and 3rd smallest elements in minimum number of comparisons.

[gate1999](#) [algorithms](#) [sorting](#) [normal](#) [descriptive](#)
Answer

1.16.20 Sorting: GATE2000-17 [top](#)

<https://gateoverflow.in/688>

An array contains four occurrences of 0, five occurrences of 1, and three occurrences of 2 in any order. The array is to be sorted using swap operations (elements that are swapped need to be adjacent).

- a. What is the minimum number of swaps needed to sort such an array in the worst case?
- b. Give an ordering of elements in the above array so that the minimum number of swaps needed to sort the array is maximum.

[gate2000](#) [algorithms](#) [sorting](#) [normal](#) [descriptive](#)
Answer

1.16.21 Sorting: GATE2001-1.14 [top](#)

<https://gateoverflow.in/707>

Randomized quicksort is an extension of quicksort where the pivot is chosen randomly. What is the worst case complexity of sorting n numbers using Randomized quicksort?

- A. $O(n)$
- B. $O(n \log n)$
- C. $O(n^2)$
- D. $O(n!)$

[gate2001](#) [algorithms](#) [sorting](#) [time-complexity](#) [easy](#)
Answer

1.16.22 Sorting: GATE2003-22 [top](#)

<https://gateoverflow.in/912>

The unusual $\Theta(n^2)$ implementation of Insertion Sort to sort an array uses linear search to identify the position where an element is to be inserted into the already sorted part of the array. If, instead, we use binary search to identify the position, the worst case running time will

- A. remain $\Theta(n^2)$
- B. become $\Theta(n(\log n)^2)$
- C. become $\Theta(n \log n)$
- D. become $\Theta(n)$

[gate2003](#) [algorithms](#) [sorting](#) [time-complexity](#) [normal](#)
Answer

1.16.23 Sorting: GATE2003-61 [top](#)<https://gateoverflow.in/949>

In a permutation $a_1 \dots a_n$, of n distinct integers, an inversion is a pair (a_i, a_j) such that $i < j$ and $a_i > a_j$.

If all permutations are equally likely, what is the expected number of inversions in a randomly chosen permutation of $1 \dots n$?

- A. $\frac{n(n-1)}{2}$
- B. $\frac{n(n-1)}{4}$
- C. $\frac{n(n+1)}{4}$
- D. $2n[\log_2 n]$

[gate2003](#) [algorithms](#) [sorting](#) [normal](#)
Answer**1.16.24 Sorting: GATE2003-62** [top](#)<https://gateoverflow.in/43576>

In a permutation $a_1 \dots a_n$, of n distinct integers, an inversion is a pair (a_i, a_j) such that $i < j$ and $a_i > a_j$.

What would be the worst case time complexity of the Insertion Sort algorithm, if the inputs are restricted to permutations of $1 \dots n$ with at most n inversions?

- A. $\Theta(n^2)$
- B. $\Theta(n \log n)$
- C. $\Theta(n^{1.5})$
- D. $\Theta(n)$

[gate2003](#) [algorithms](#) [sorting](#) [normal](#)
Answer**1.16.25 Sorting: GATE2004-29** [top](#)<https://gateoverflow.in/1026>

The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of

- A. n
- B. n^2
- C. $n \log n$
- D. $n \log^2 n$

[gate2004](#) [algorithms](#) [sorting](#) [asymptotic-notations](#) [easy](#)
Answer**1.16.26 Sorting: GATE2005-39** [top](#)<https://gateoverflow.in/784>

Suppose there are $\lceil \log n \rceil$ sorted lists of $\lfloor n / \log n \rfloor$ elements each. The time complexity of producing a sorted list of all these elements is: (Hint: Use a heap data structure)

- A. $O(n \log \log n)$
- B. $\Theta(n \log n)$

- C. $\Omega(n \log n)$
 D. $\Omega(n^{3/2})$

gate2005 algorithms sorting normal

Answer

1.16.27 Sorting: GATE2005-IT-59 top

<https://gateoverflow.in/3820>

Let a and b be two sorted arrays containing n integers each, in non-decreasing order. Let c be a sorted array containing $2n$ integers obtained by merging the two arrays a and b . Assuming the arrays are indexed starting from 0, consider the following four statements

- I. $a[i] \geq b[i] \Rightarrow c[2i] \geq a[i]$
- II. $a[i] \geq b[i] \Rightarrow c[2i] \geq b[i]$
- III. $a[i] \geq b[i] \Rightarrow c[2i] \leq a[i]$
- IV. $a[i] \geq b[i] \Rightarrow c[2i] \leq b[i]$

Which of the following is TRUE?

- A. only I and II
- B. only I and IV
- C. only II and III
- D. only III and IV

gate2005-it algorithms sorting normal

Answer

1.16.28 Sorting: GATE2006-14, ISRO2011-14 top

<https://gateoverflow.in/975>

Which one of the following in place sorting algorithms needs the minimum number of swaps?

- A. Quick sort
- B. Insertion sort
- C. Selection sort
- D. Heap sort

gate2006 algorithms sorting easy isro2011

Answer

1.16.29 Sorting: GATE2006-52 top

<https://gateoverflow.in/1830>

The median of n elements can be found in $O(n)$ time. Which one of the following is correct about the complexity of quick sort, in which median is selected as pivot?

- A. $\Theta(n)$
- B. $\Theta(n \log n)$
- C. $\Theta(n^2)$
- D. $\Theta(n^3)$

gate2006 algorithms sorting easy

Answer

1.16.30 Sorting: GATE2007-14 top

<https://gateoverflow.in/1212>

Which of the following sorting algorithms has the lowest worse-case complexity?

- A. Merge sort
- B. Bubble sort
- C. Quick sort
- D. Selection sort

gate2007 | algorithms | sorting | time-complexity | easy

[Answer](#)

1.16.31 Sorting: GATE2008-43 [top](#)

<https://gateoverflow.in/455>

Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let $T(n)$ be the number of comparisons required to sort n elements. Then

- A. $T(n) \leq 2T(n/5) + n$
- B. $T(n) \leq T(n/5) + T(4n/5) + n$
- C. $T(n) \leq 2T(4n/5) + n$
- D. $T(n) \leq 2T(n/2) + n$

gate2008 | algorithms | sorting | easy

[Answer](#)

1.16.32 Sorting: GATE2008-IT-43 [top](#)

<https://gateoverflow.in/3353>

If we use Radix Sort to sort n integers in the range $(n^{k/2}, n^k]$, for some $k > 0$ which is independent of n , the time taken would be?

- A. $\Theta(n)$
- B. $\Theta(kn)$
- C. $\Theta(n \log n)$
- D. $\Theta(n^2)$

gate2008-it | algorithms | sorting | normal

[Answer](#)

1.16.33 Sorting: GATE2009-11 [top](#)

<https://gateoverflow.in/1303>

What is the number of swaps required to sort n elements using selection sort, in the worst case?

- A. $\Theta(n)$
- B. $\Theta(n \log n)$
- C. $\Theta(n^2)$
- D. $\Theta(n^2 \log n)$

gate2009 | algorithms | sorting | easy

Answer**1.16.34 Sorting: GATE2009-39** top<https://gateoverflow.in/1325>

In quick-sort, for sorting n elements, the $(n/4)^{th}$ smallest element is selected as pivot using an $O(n)$ time algorithm. What is the worst case time complexity of the quick sort?

- A. $\Theta(n)$
- B. $\Theta(n \log n)$
- C. $\Theta(n^2)$
- D. $\Theta(n^2 \log n)$

gate2009 algorithms sorting normal

Answer**1.16.35 Sorting: GATE2012-39** top<https://gateoverflow.in/1762>

A list of n strings, each of length n , is sorted into lexicographic order using the merge-sort algorithm. The worst case running time of this computation is

- A. $O(n \log n)$
- B. $O(n^2 \log n)$
- C. $O(n^2 + \log n)$
- D. $O(n^2)$

gate2012 algorithms sorting normal

Answer**1.16.36 Sorting: GATE2013-30** top<https://gateoverflow.in/1541>

The number of elements that can be sorted in $\Theta(\log n)$ time using heap sort is

- A. $\Theta(1)$
- B. $\Theta(\sqrt{\log n})$
- C. $\Theta\left(\frac{\log n}{\log \log n}\right)$
- D. $\Theta(\log n)$

gate2013 algorithms sorting normal

Answer**1.16.37 Sorting: GATE2013-6** top<https://gateoverflow.in/1415>

Which one of the following is the tightest upper bound that represents the number of swaps required to sort n numbers using selection sort?

- A. $O(\log n)$
- B. $O(n)$
- C. $O(n \log n)$
- D. $O(n^2)$

[gate2013](#)
[algorithms](#)
[sorting](#)
[easy](#)
Answer**1.16.38 Sorting: GATE2014-1-14** [top](#)<https://gateoverflow.in/1780>

Let P be quicksort program to sort numbers in ascending order using the first element as the pivot. Let t_1 and t_2 be the number of comparisons made by P for the inputs $[1\ 2\ 3\ 4\ 5]$ and $[4\ 1\ 5\ 3\ 2]$ respectively. Which one of the following holds?

- A. $t_1 = 5$
- B. $t_1 < t_2$
- C. $t_1 > t_2$
- D. $t_1 = t_2$

[gate2014-1](#)
[algorithms](#)
[sorting](#)
[easy](#)
Answer**1.16.39 Sorting: GATE2014-2-38** [top](#)<https://gateoverflow.in/1997>

Suppose P, Q, R, S, T are sorted sequences having lengths 20, 24, 30, 35, 50 respectively. They are to be merged into a single sequence by merging together two sequences at a time. The number of comparisons that will be needed in the worst case by the optimal algorithm for doing this is ____.

[gate2014-2](#)
[algorithms](#)
[sorting](#)
[normal](#)
[numerical-answers](#)
Answer**1.16.40 Sorting: GATE2014-3-14** [top](#)<https://gateoverflow.in/2048>

You have an array of n elements. Suppose you implement quicksort by always choosing the central element of the array as the pivot. Then the tightest upper bound for the worst case performance is

- A. $O(n^2)$
- B. $O(n \log n)$
- C. $\Theta(n \log n)$
- D. $O(n^3)$

[gate2014-3](#)
[algorithms](#)
[sorting](#)
[easy](#)
Answer**1.16.41 Sorting: GATE2015-1-2** [top](#)<https://gateoverflow.in/8017>

Which one of the following is the recurrence equation for the worst case time complexity of the quick sort algorithm for sorting n (≥ 2) numbers? In the recurrence equations given in the options below, c is a constant.

- A. $T(n) = 2T(n/2) + cn$
- B. $T(n) = T(n - 1) + T(1) + cn$
- C. $T(n) = 2T(n - 1) + cn$
- D. $T(n) = T(n/2) + cn$

[gate2015-1](#)
[algorithms](#)
[recurrence](#)
[sorting](#)
[easy](#)
Answer**1.16.42 Sorting: GATE2015-2-45** [top](#)<https://gateoverflow.in/8243>

Suppose you are provided with the following function declaration in the C programming language.

```
int partition(int a[], int n);
```

The function treats the first element of $a[]$ as a pivot and rearranges the array so that all elements less than or equal to the pivot is in the left part of the array, and all elements greater than the pivot is in the right part. In addition, it moves the pivot so that the pivot is the last element of the left part. The return value is the number of elements in the left part.

The following partially given function in the C programming language is used to find the k^{th} smallest element in an array $a[]$ of size n using the partition function. We assume $k \leq n$.

```
int kth_smallest (int a[], int n, int k)
{
    int left_end = partition (a, n);
    if (left_end+1==k) {
        return a[left_end];
    }
    if (left_end+1 > k) {
        return kth_smallest (_____);
    } else {
        return kth_smallest (_____);
    }
}
```

The missing arguments lists are respectively

- A. $(a, \text{left_end}, k)$ and $(a+\text{left_end}+1, n-\text{left_end}-1, k-\text{left_end}-1)$
- B. $(a, \text{left_end}, k)$ and $(a, n-\text{left_end}-1, k-\text{left_end}-1)$
- C. $(a, \text{left_end}+1, n-\text{left_end}-1, k-\text{left_end}-1)$ and $(a, \text{left_end}, k)$
- D. $(a, n-\text{left_end}-1, k-\text{left_end}-1)$ and $(a, \text{left_end}, k)$

[gate2015-2](#) [algorithms](#) [normal](#) [sorting](#)

Answer

1.16.43 Sorting: GATE2015-3-27 [top](#)

<https://gateoverflow.in/8480>

Assume that a mergesort algorithm in the worst case takes 30 seconds for an input of size 64. Which of the following most closely approximates the maximum input size of a problem that can be solved in 6 minutes?

- A. 256
- B. 512
- C. 1024
- D. 2018

[gate2015-3](#) [algorithms](#) [sorting](#)

Answer

1.16.44 Sorting: GATE2016-1-13 [top](#)

<https://gateoverflow.in/39660>

The worst case running times of *Insertion sort*, *Merge sort* and *Quick sort*, respectively are:

- A. $\Theta(n \log n)$, $\Theta(n \log n)$ and $\Theta(n^2)$
- B. $\Theta(n^2)$, $\Theta(n^2)$ and $\Theta(n \log n)$
- C. $\Theta(n^2)$, $\Theta(n \log n)$ and $\Theta(n \log n)$

- D. $\Theta(n^2)$, $\Theta(n \log n)$ and $\Theta(n^2)$

gate2016-1 algorithms sorting easy

Answer

1.16.45 Sorting: GATE2016-2-13 [top](#)

<https://gateoverflow.in/39561>

Assume that the algorithms considered here sort the input sequences in ascending order. If the input is already in the ascending order, which of the following are **TRUE**?

- I. Quicksort runs in $\Theta(n^2)$ time
 - II. Bubblesort runs in $\Theta(n^2)$ time
 - III. Mergesort runs in $\Theta(n)$ time
 - IV. Insertion sort runs in $\Theta(n)$ time
-
- A. I and II only
 - B. I and III only
 - C. II and IV only
 - D. I and IV only

gate2016-2 algorithms sorting time-complexity normal ambiguous

Answer

1.16.46 Sorting: ISI2011-A-2a [top](#)

<https://gateoverflow.in/48028>

Give a strategy to sort four given distinct integers a, b, c, d in increasing order that minimizes the number of pairwise comparisons needed to sort any permutation of a, b, c, d .

descriptive isi2011 algorithms sorting

Answer

1.16.47 Sorting: TIFR2010-B-23 [top](#)

<https://gateoverflow.in/18623>

Suppose you are given n numbers and you sort them in descending order as follows:

First find the maximum. Remove this element from the list and find the maximum of the remaining elements, remove this element, and so on, until all elements are exhausted. How many comparisons does this method require in the worst case?

- A. Linear in n .
- B. $O(n^2)$ but not better.
- C. $O(n \log n)$
- D. Same as heap sort.
- E. $O(n^{1.5})$ but not better.

tifr2010 algorithms time-complexity sorting

Answer

1.16.48 Sorting: TIFR2010-B-27 [top](#)

<https://gateoverflow.in/19036>

Consider the Insertion Sort procedure given below, which sorts an array L of size n (≥ 2) in ascending order:

```
begin
    for xindex:= 2 to n do
```

```

x := L[xindex];
j := xindex - 1;
while j > 0 and L[j] > x do
    L[j + 1] := L[j];
    j := j - 1;
end {while}
L[j + 1] := X;
end{for}
end

```

It is known that insertion sort makes at most $n(n - 1)/2$ comparisons. Which of the following is true?

- A. There is no input on which insertion Sort makes $n(n - 1)/2$ comparisons.
- B. Insertion Sort makes $n(n - 1)/2$ comparisons when the input is already sorted in ascending order.
- C. Insertion Sort makes $n(n - 1)/2$ comparisons only when the input is sorted in descending order.
- D. There are more than one input orderings where insertion sort makes $n(n - 1)/2$ comparisons.
- E. Insertion Sort makes $n(n - 1)/2$ comparisons whenever all the elements of L are not distinct.

[tifr2010](#) [algorithms](#) [sorting](#)

[Answer](#)

1.16.49 Sorting: TIFR2011-B-21 [top](#)

<https://gateoverflow.in/20324>

Let $S = \{x_1, \dots, x_n\}$ be a set of n numbers. Consider the problem of storing the elements of S in an array $A[1\dots n]$ such that the following min-heap property is maintained for all $2 \leq i \leq n : A[\lfloor i/2 \rfloor] \leq A[i]$. (Note that $\lfloor x \rfloor$ is the largest integer that is at most x). Which of the following statements is TRUE?

- A. This problem can be solved in $O(\log n)$ time.
- B. This problem can be solved in $O(n)$ time but not in $O(\log n)$ time.
- C. This problem can be solved in $O(n \log n)$ time but not in $O(n)$ time.
- D. This problem can be solved in $O(n^2)$ time but not in $O(n \log n)$ time.
- E. None of the above.

[tifr2011](#) [algorithms](#) [sorting](#)

[Answer](#)

1.16.50 Sorting: TIFR2011-B-31 [top](#)

<https://gateoverflow.in/20617>

Given a set of $n = 2^k$ distinct numbers, we would like to determine the smallest and the second smallest using comparisons. Which of the following statements is TRUE?

- A. Both these elements can be determined using $2k$ comparisons.
- B. Both these elements can be determined using $n - 2$ comparisons.
- C. Both these elements can be determined using $n + k - 2$ comparisons.
- D. $2n - 3$ comparisons are necessary to determine these two elements.
- E. nk comparisons are necessary to determine these two elements.

[tifr2011](#) [algorithms](#) [sorting](#)

[Answer](#)

1.16.51 Sorting: TIFR2011-B-39 [top](#)

<https://gateoverflow.in/20935>

The first n cells of an array L contain positive integers sorted in decreasing order, and the remaining $m - n$ cells all contain 0. Then, given an integer x , in how many comparisons can one find the position of x in L ?

- A. At least n comparisons are necessary in the worst case.
- B. At least $\log m$ comparisons are necessary in the worst case.
- C. $O(\log(m - n))$ comparisons suffice.
- D. $O(\log n)$ comparisons suffice.
- E. $O(\log(m/n))$ comparisons suffice.

[tifr2011](#) [algorithms](#) [sorting](#)

[Answer](#)

1.16.52 Sorting: TIFR2012-B-13 [top](#)

<https://gateoverflow.in/25207>

An array A contains n integers. We wish to sort A in ascending order. We are told that initially no element of A is more than a distance k away from its final position in the sorted list. Assume that n and k are large and k is much smaller than n . Which of the following is true for the worst case complexity of sorting A ?

- A. A can be sorted with constant $.kn$ comparison but not with fewer comparisons.
- B. A cannot be sorted with less than constant $.n \log n$ comparisons.
- C. A can be sorted with constant $.n$ comparisons.
- D. A can be sorted with constant $.n \log k$ comparisons but not with fewer comparisons.
- E. A can be sorted with constant $.k^2n$ comparisons but not fewer.

[tifr2012](#) [algorithms](#) [sorting](#)

[Answer](#)

1.16.53 Sorting: TIFR2012-B-14 [top](#)

<https://gateoverflow.in/25209>

Consider the quick sort algorithm on a set of n numbers, where in every recursive subroutine of the algorithm, the algorithm chooses the median of that set as the pivot. Then which of the following statements is TRUE?

- A. The running time of the algorithm is $\Theta(n)$.
- B. The running time of the algorithm is $\Theta(n \log n)$.
- C. The running time of the algorithm is $\Theta(n^{1.5})$.
- D. The running time of the algorithm is $\Theta(n^2)$.
- E. None of the above.

[tifr2012](#) [algorithms](#) [sorting](#)

[Answer](#)

1.16.54 Sorting: TIFR2013-B-20 [top](#)

<https://gateoverflow.in/25878>

Suppose n processors are connected in a linear array as shown below. Each processor has a number. The processors need to exchange numbers so that the numbers eventually appear in ascending order (the processor P_1 should have the minimum value and the processor P_n should have the maximum value).



The algorithm to be employed is the following. Odd numbered processors and even numbered processors are activated alternate steps; assume that in the first step all the even numbered processors are activated. When a processor is activated, the number it holds is compared with the number held by its right-hand neighbour (if one exists) and the smaller of the two numbers is retained by the activated processor and the bigger stored in its right hand neighbour.

How long does it take for the processors to sort the values?

- A. $n \log n$ steps
- B. n^2 steps
- C. n steps
- D. $n^{1.5}$ steps
- E. The algorithm is not guaranteed to sort

[tifr2013](#) [algorithms](#) [sorting](#)

[Answer](#)

1.16.55 Sorting: TIFR2017-B-7 [top](#)

<https://gateoverflow.in/95699>

An array of n distinct elements is said to be un-sorted if for every index i such that $2 \leq i \leq n - 1$, either $A[i] > \max\{A[i - 1], A[i + 1]\}$, or $A[i] < \min\{A[i - 1], A[i + 1]\}$. What is the time-complexity of the fastest algorithm that takes as input a sorted array A with n distinct elements, and un-sorts A ?

- A. $O(n \log n)$ but not $O(n)$
- B. $O(n)$ but not $O(\sqrt{n})$
- C. $O(\sqrt{n})$ but not $O(\log n)$
- D. $O(\log n)$ but not $O(1)$
- E. $O(1)$

[tifr2017](#) [algorithms](#) [sorting](#)

[Answer](#)

Answers: Sorting

1.16.1 Sorting: CMI2011-B-06a [top](#)

<https://gateoverflow.in/46212>



Selected Answer

Let's say we have disk of radius 0,5,1,4,3,2

We have many disks, one above other, our task is to arrange them from largest disk to smaller disk.

The only operations available is, take a couple of disks from the top, hold it in your hand, rotate it & put it back.

Just imagine in a real scenario.

How we will do it.

0,5,1,4,3,2

What I will do is, I will take all plates from 5 to 2 (5, 1, 4, 3, 2) I hold them in my hand, rotated my hand 180 degree & put it back.

0, 2, 3, 4, 1, 5

Now I will rotate all.

5, 1, 4, 3, 2, 0

Now 5 is at its right place. We will not consider it.

Now find the second largest plate... 4.. rotate from 4 to 0
 $5 | , 1, 0, 2, 3, 4$
 rotate from 1 to 4.

$5, 4 | , 3, 2, 0, 1$

Now 4 is at its right place.

Next maximum is 3, it is at its right place, so we may add one more line in the code to reduce steps...

Otherwise, the same algo will also keep 3 at its right place.
 The same algo will sort them at last.

Hence, our algo is:

Step 1:

Find the current largest element.

Step 2:

Rotate the elements from largest element to end of the array. (Hold the disks from disk of max size to disk at top & rotate)

Step 3:

Now rotate the array from current start+1 to top.

Continue this process till it is sorted.

Note that initially current_start = -1 it will be incremented by one every time we get our largest disk placed at its correct place.

7 votes

-- Ahwan Mishra (10.5k points)

1.16.2 Sorting: CMI2013-A-05 top

<https://gateoverflow.in/46595>



Selected Answer

Since, n lists of each size m .

Since, each list is sorted in ascending order use directly merge procedure of merge sort algo.

Take two list and merge..so one pair will take
2m time.

So, total pairs in first level will be $n/2$. So total cost for one level is
 $(n/2) * 2m = nm$.

In next level cost for one pair is $4m$ and no of pairs will be $n/4$. so next level cost will be nm .

So, like this each level will have cost nm .

No of levels will be when we have one complete list..

$$n/2^k = 1..$$

$$k = \log_2 n..$$

So, total cost will be $\log n * (nm)$

17 votes

-- SONU (2.3k points)

1.16.3 Sorting: CMI2017-A-08 [top](#)

<https://gateoverflow.in/203281>



Selected Answer

A stable sort preserves the order of values that are equal with respect to the comparison function.

What does that mean?

That is whenever we compare two keys(values) for sorting and if they are equal, the order in which they will appear in the sorted list is same as they appeared in the original list. That is, if we have x_1 and x_2 in our list, such that $x_1 = x_2$ (they are equal when compared) and x_1 appeared before x_2 in the list (our list is something like: $\dots, x_1, \dots, x_2, \dots$).

Then the sorted list will be like: \dots, x_1, x_2, \dots and NOT like: \dots, x_2, x_1, \dots . That is, in the sorted list x_1 comes before x_2 . Then such a sort is known to be **stable**.

<https://stackoverflow.com/questions/1517793/what-is-stability-in-sorting-algorithms-and-why-is-it-important>

Now the given list:

$[(7, 1, 8), (3, 5, 7), (6, 1, 4), (6, 5, 9), (0, 2, 5), (9, 0, 9)]$

is sorted in ascending order by the **second coordinate**. We have to compare second coordinate to get the ordering.

Answer should be:

(c)

$[(9, 0, 9), (7, 1, 8), (6, 1, 4), (0, 2, 5), (3, 5, 7), (6, 5, 9)]$

Notice here that when compared, $(7, 1, 8)$ is equal to $(6, 1, 4)$ (compare second coordinates). So, in the sorted list $(7, 1, 8)$ comes before $(6, 1, 4)$, same order as in original list.

Similarly for $(3, 5, 7)$ and $(6, 5, 9)$.

1 votes

-- Rishabh Gupta (13.9k points)

1.16.4 Sorting: GATE1987-1-xviii [top](#)

<https://gateoverflow.in/80366>



Selected Answer

ACTUALLY IN BOTH THE CASES IT WILL TAKE $O(n^2)$ TIME ($O(n)$ TIME FOR PARTITION ALGORITHM AND $T(n - 1)$ TIME FOR SUB PROBLEM. AS n IS THE NUMBER OF INPUTS AND IN THE 2ND CASE INPUTS ARE 5(GREATER THAN 1ST ONE THAT IS 4) I THINK $t_1 < t_2$

18 votes

-- Rohan Ghosh (2k points)

1.16.5 Sorting: GATE1988-1iii [top](#)

<https://gateoverflow.in/91338>



Selected Answer

Answer is **LESS**.

As worst case time for Quicksort is $O(n^2)$ and worst case for heap sort is $O(n \log n)$.

1.16.9 Sorting: GATE1992-03,iv top

<https://gateoverflow.in/581>



The algorithm will take maximum time when:

1. The array is already sorted in same order.
2. The array is already sorted in reverse order.
3. All elements are same in the array.

1 25 votes

-- Rajarshi Sarkar (34.1k points)

1.16.10 Sorting: GATE1994-1.19, ISRO2016-31 top

<https://gateoverflow.in/2462>



Answer: Option C.

It is one of the efficient algorithms in **Divide and Conquer** strategy.

1 22 votes

-- Gate Keeda (19.6k points)

1.16.11 Sorting: GATE1995-1.16 top

<https://gateoverflow.in/2603>



Answer: Option C.

The number of moves are however always $m + n$ so that we can term it as $\Theta(m + n)$. But the number of comparisons vary as per the input. In the best case the comparisons are $\text{Min}(m, n)$ and in worst case they are $m + n - 1$.

1 26 votes

-- Gate Keeda (19.6k points)

1.16.12 Sorting: GATE1995-1.5 top

<https://gateoverflow.in/2592>



Answer: Option C.

One of the best examples of **Divide and Conquer** strategy.

1 16 votes

-- Gate Keeda (19.6k points)

1.16.13 Sorting: GATE1995-12 top

<https://gateoverflow.in/2648>



1st Pass: 37 52 12 11 25 92

2nd Pass: 37 12 11 25 52 92

3rd Pass: 12 11 25 37 52 92

4th Pass: 11 12 25 37 52 92

5th Pass: 11 12 25 37 52 92

17 votes

-- Gate Keeda (19.6k points)

1.16.14 Sorting: GATE1996-14 top

<https://gateoverflow.in/2766>



Selected Answer

- a. The smallest element is at index 1,1.
- b. So we have to give an array which is partially sorted. Definition of partially sorted is given in the question.
We will give the value of x which is less than last row & column value.
At last, 1,1 should be deleted & x should be at its correct place.

```
i=1; j=1; a[i][j]=x;
while ((x>a[i+1][j]) || (x>a[i][j+1]))
{
    if((a[i+1][j] < x) && (a[i+1][j] < a[i][j+1]))
    {
        a[i][j]=a[i+1][j];
        i=i+1;
    }
    else
    {
        a[i][j]=a[i][j+1];
        j=j+1;
    }
}
a[i][j]=x;
```

Enter the dimension of $n \times n$ array. Give the value of n

3

Enter the array elements in partially sorted order

2 3 9

5 6 10

8 11 15

Enter the value of x

7

The final output.

3 6 9

5 7 10

8 11 15

12 votes

-- Ahwan Mishra (10.5k points)

1.16.15 Sorting: GATE1996-2.15 top

<https://gateoverflow.in/2744>



Selected Answer

C.

both are the worst cases of quick sort. (assuming pivot is either first or last element)

- is sorted in ascending order.
- is sorted in descending order.

20 votes

-- Gate Keeda (19.6k points)

1.16.16 Sorting: GATE1998-1.22 top

<https://gateoverflow.in/1659>



Selected Answer

- Selection sort : $O(n^2)$
- Merge sort : $O(n \log n)$
- Binary search : $(\log n)$
- Insertion sort : $O(n)$

Note: if we use $O(n^2)$ for Insertion sort, we will not be having any suitable choice to fill selection sort. So, we can assume that the question is asking for best case time complexities.

13 votes

-- Bhagirathi Nayak (14.1k points)

1.16.17 Sorting: GATE1999-1.12 top

<https://gateoverflow.in/1465>



Selected Answer

(B) If it maintains the relative order of occurrence of non-distinct elements.

(from definition of stable sorting)

16 votes

-- Arjun Suresh (350k points)

1.16.18 Sorting: GATE1999-1.14, ISRO2015-42 top

<https://gateoverflow.in/1467>



Selected Answer

```

20  47  15  8  9  4  40  30  12  17
 \   /   \   /   \   /   \   /   \   /
20 47  8 15  49  30 40  12 17    after 1st pass
 \   /   \   /   \   /   \   /
8,15,20,47  4,9,30,40  12,17    after 2nd pass

```

Answer is B.

54 votes

-- Vikrant Singh (13.5k points)

1.16.19 Sorting: GATE1999-8 top

<https://gateoverflow.in/1507>



Selected Answer

This is a two dimensional array of size 4×4 .

2	5	8	9
4	11	19	21
6	13	24	27
8	15	25	29

You can see the elements in each row and each column are arranged in ascending order.

Smallest element: $A[0][0] = 2$

2nd Smallest element: $\min(A[0][1], A[1][0]) = \min(5, 4) = 4$

3rd smallest element: Just exclude the element you got as 2nd smallest(4). Here, we can compare $A[2][0], A[0][1]$ no need to compare with $A[0][2]$. So it depends upon from where you got 2nd element. You can draw a decision tree. If you got 2nd best from $A[0][1]$ then what to do & if you get from $A[1][0]$ then what to do.

Any way, time complexity is simply $O(1)$.

The elements are in ascending order. Not in non decreasing order. Clearly they are all distinct in a particular row or column.

14 votes

-- Ahwan Mishra (10.5k points)

1.16.20 Sorting: GATE2000-17 top

<https://gateoverflow.in/688>



Selected Answer

Since swaps are needed to be of adjacent elements only, the algorithm is actually Bubble sort.

In bubble sort, all smaller elements to right of an element are required to be swapped. So, if have ordering

[2, 2, 2, 1, 1, 1, 1, 0, 0, 0, 0], then we need total 47 swaps, and this will be the worst case.

So, it answers actually both parts.

23 votes

-- Happy Mittal (11.2k points)

1.16.21 Sorting: GATE2001-1.14 top

<https://gateoverflow.in/707>



Selected Answer

Answer will be (C).

There are following two cases, when Randomized Quick Sort will result into worstcase of time complexity $O(n^2)$

- When all elements are same in the input array, Partition algo will divide input array in two subarray, one with $n - 1$ elements and second with 0 element. There is an assumption here that, we are using the same partition algorithm without any modification.
- If the randomised pivot selector happens to select e.g. the smallest element N times in a row, we will get the worst possible performance. Though the probability of this particular case is about $\frac{1}{n!}$

PS:- If the partitioning is **unbalanced**, Quick Sort algorithm runs asymptotically as slow as **Insertion Sort** i.e $O(n^2)$

12 votes

-- Manu Thakur (39.6k points)

1.16.22 Sorting: GATE2003-22 top

<https://gateoverflow.in/912>

Selected Answer

In insertion sort, with linear search, it takes

(worst case) n comparisons for searching the right position, and n swaps to make room to place the element.

Hence for n elements, a total of $n \times (n + n)$; n for search and n for swaps.

$$= \Theta(2n^2) = \Theta(n^2)$$

If we replace it with binary search, it takes

(worst case) $\log n$ comparisons for searching the right position, and n swaps to make room to place the element.

Hence for n elements, a total of $n \times (\log n + n)$; n for search and n for swaps.

$$= \Theta(n \times \log n + n^2) = \Theta(n^2)$$

Hence, **answer is A.**

38 votes

-- ryan sequeira (3.4k points)

1.16.23 Sorting: GATE2003-61 top

<https://gateoverflow.in/949>

Selected Answer

They are asking the average number of inversion. basically what i learned about averages from dbms indexing is.

Average apart from the standard definition can be calculated as $(best\ case + worst\ case)/2$
and inversion is like 9, 5.

So, best case will be sorted array $-1, 2, 3, 4, 5$ no inversion $= zero$

worst case $= 9, 5, 4, 3, 2, 1$. here total number of inversion will be $n(n - 1)/2$ as . 9 can be paired with any 5 elements $(5, 4, 3, 2, 1)$ will form a inversion pair. similarly 5 with 4.elements .

So, we can say if we have n elements then. it will be $(n - 1) + (n - 2) + (n - 3) \dots + 2 + 1$
which is the sum of first $n - 1$ natural numbers. So, it is $n(n - 1)/2$

So, expected average number of inversion $= (n(n - 1)/2 + zero(best\ case))/2 = n(n - 1)/4$

So, option is **B.**

Second question.

we all know that insertion sort has complexity due to swapping and movements, if we have n n inversion pair then the movements and comparison will be restricted to n only . like if inversion is 1 , then array must be sorted and only the inversion should exist at the end, like 1, 2, 3, 5, 4.

otherwise more than one inversion pair will form. so to sort this. for two it will be 1, 2, 3, 7, 5, 4. so to sort this type of array using insertion sort atmost N swaps will be required, so D,

36 votes

-- Ravi Singh (15.7k points)

1.16.24 Sorting: GATE2003-62 top

<https://gateoverflow.in/43576>



Selected Answer

ANSWER: D. $\Theta(n)$

REASON:

Count of number of times the inner loop of insertion sort executes is actually equal to number of inversions in input permutation a_1, a_2, \dots, a_n . Since for each value of $i = 1 \dots n$, j take the value $1 \dots i - 1$, which means for every $j < i$ it checks if $a[j] > a[i]$.

In any given permutation, maximum number of inversions possible is $n(n - 1)/2$ which is $O(n^2)$. It is the case where the array is sorted in reverse order. Therefore, to resolve all inversions i.e., worst case time complexity of insertion sort is $\Theta(n^2)$.

However, as per the question the number of inversion in input array is restricted to n . The worst case time complexity of insertion sort reduces to $\Theta(n)$.

[INSERTION SORT ALGORITHM](#) (for reference)

25 votes

-- Prateek Dwivedi (4.4k points)

1.16.25 Sorting: GATE2004-29 top

<https://gateoverflow.in/1026>



Selected Answer

For comparison-based sorting the asymptotically tight bound for worst case is given by $\Theta(n \log n)$, which means it is the tightest upper bound (big O) as well as the tightest lower bound (big omega). So, answer is $n \log n$.

Tightest lower bound of sorting (say $S(n)$) is $n \log n$ means there is no function f which has an order of growth larger than $n \log n$ and $f(n) = \Omega(S(n))$ holds.

A usual mistake is to think worst case changes with lower and upper bounds, but that is not the case. Worst case is defined for the algorithm and it is always the input which causes the algorithm the maximum complexity.

37 votes

-- Arjun Suresh (350k points)

1.16.26 Sorting: GATE2005-39 top

<https://gateoverflow.in/784>



Selected Answer

Since we have $\log n$ lists we can make a min-heap of $\log n$ elements by taking the first element from each of the $\log n$ sorted lists. Now, we start deleting the min-element from the heap and put the next element from the sorted list from which that element was added to the heap. (This identity can be done by making a structure of two values, one for the number and one for identifying the origin sorted list of that number and storing this structure in the heap). In this way

each delete and the corresponding insert will take $O(\log \log n)$ time as delete in heap of size n is $O(\log n)$ and inserting an element on a heap of size n is also $O(\log n)$. (here, heap size is $\log n$). Now, we have a total of $\log n \times \frac{n}{\log n} = n$ elements. So, total time will be $O(n \log \log n)$.

42 votes

-- gatecse (18k points)

1.16.27 Sorting: GATE2005-IT-59 top

<https://gateoverflow.in/3820>



Selected Answer

$$a[i] \geq b[i]$$

Since both a and b are sorted in the beginning, there are i elements smaller than or equal to $a[i]$ (i starts from 0), and similarly i elements smaller than or equal to $b[i]$. So, $a[i] \geq b[i]$ means there are $2i$ elements smaller than or equal to $a[i]$, and hence in the merged array $a[i]$ will come after these $2i$ elements (its index will be $> 2i$). So, $c[2i] \leq a[i]$ (equality takes care of the "equal to" case which comes when array contains repeated elements).

Similarly, $a[i] \geq b[i]$ says for b that, there are not more than $2i$ elements smaller than $b[i]$ in the sorted array (i elements from b , and maximum another i elements from a). So, $b[i] \leq c[2i]$

So, II and III are correct is option (C).

26 votes

-- Arjun Suresh (350k points)

1.16.28 Sorting: GATE2006-14, ISRO2011-14 top

<https://gateoverflow.in/975>



Selected Answer

Selection sort.

Because in selection the maximum swaps which can take place are $O(n)$

Because we pick up an element and find the minimum (in case of forward sorting) from the next index till the end of array and then perform the swap

Hence, $O(n)$ whereas in all other algos the swaps are greater (considering Worst-Case scenario)

19 votes

-- ANKUR MAHIWAL (411 points)

1.16.29 Sorting: GATE2006-52 top

<https://gateoverflow.in/1830>



Selected Answer

As we choose the pivot a median element ... so, every time we are going to have good splits guaranteed so the best case $O(n \log n)$.

23 votes

-- Bhagirathi Nayak (14.1k points)

1.16.30 Sorting: GATE2007-14 top

<https://gateoverflow.in/1212>



Selected Answer

A.

Irrespective of the input, merge sort always have a time complexity of $\Theta(n \log n)$.

17 votes

-- Gate Keeda (19.6k points)

1.16.31 Sorting: GATE2008-43 top

<https://gateoverflow.in/455>



Selected Answer

$$T(n) \leq T(n/5) + T(4n/5) + n$$

One part contains $n/5$ elements
and the other part contains $4n/5$ elements
 $+n$ is common to all options, so we need not to worry about it.

Hence, answer is **option B.**

22 votes

-- Amar Vashishth (30.5k points)

1.16.32 Sorting: GATE2008-IT-43 top

<https://gateoverflow.in/3353>



Selected Answer

Answer: C

The complexity of Radix Sort is $O(wn)$, for n keys which are integers of word size w .

$$\text{Here, } w = \log_2(n^k) = k \times \log_2(n)$$

So, the complexity is $O(wn) = O(k \times \log_2(n) \times n)$, which leads to option **C**.

34 votes

-- Rajarshi Sarkar (34.1k points)

1.16.33 Sorting: GATE2009-11 top

<https://gateoverflow.in/1303>



Selected Answer

The answer is **A**.

we have 1 swap in each loop and hence n swaps at max for 1 to n . Therefore the worst case number of swaps is $\Theta(n)$

21 votes

-- Gate Keeda (19.6k points)

1.16.34 Sorting: GATE2009-39 top

<https://gateoverflow.in/1325>



Selected Answer

Answer is B.

$T(n) = O(n)$ pivot selection time + $T(n/4 - 1) + T(3n/4)$

which'll give $\Theta(n \log n)$.

Pivot selection complexity is given in questions. Pivot being the $(n/4)$ th smallest element, once it is found, we have two sub arrays- one of size $(n/4 - 1)$ and other of size $(3n/4)$ and for both of these we solve recursively.

29 votes

-- Gate Keeda (19.6k points)

1.16.35 Sorting: GATE2012-39 top

<https://gateoverflow.in/1762>



Selected Answer

I thought like this:

you are given the first character of each n strings to sort it will take $O(n \log n)$ time..in the worst case we may have to do the above process 2 times, 3 times,....., n times so $n * O(n \log n) = O(n^2 \log n)$ please correct me if my approach is wrong...

39 votes

-- Bhagirathi Nayak (14.1k points)

1.16.36 Sorting: GATE2013-30 top

<https://gateoverflow.in/1541>



Selected Answer

To sort k elements in a heap, complexity is $\Theta(k \log k)$. Lets assume there are $\frac{\log n}{\log \log n}$ elements in the heap.

$$\begin{aligned} \text{Complexity} &= \Theta\left(\frac{\log n}{\log \log n} \log\left(\frac{\log n}{\log \log n}\right)\right) \\ &= \Theta\left(\frac{\log n}{\log \log n} (\log \log n - \log \log \log n)\right) \\ &= \Theta\left(\log n - \frac{\log n \log \log \log n}{\log \log n}\right) \\ &= \Theta(\log n) \text{ (as shown below)} \end{aligned}$$

So, (C) is the answer.

$\log \log n > \log \log \log n$

$$\begin{aligned} &\implies \frac{\log \log \log n}{\log \log n} < 1 \\ &\implies \frac{\log n \log \log \log n}{\log \log n} < \log n \\ &\implies \Theta\left(\log n - \frac{\log n \log \log \log n}{\log \log n}\right) = \Theta(\log n) \end{aligned}$$

93 votes

-- Arjun Suresh (350k points)

1.16.37 Sorting: GATE2013-6 top<https://gateoverflow.in/1415>

Selected Answer

In selection max you can do is n swaps..selecting the smallest element from all the elements and replacing it correct position so $O(n)$

23 votes

-- Bhagirathi Nayak (14.1k points)

1.16.38 Sorting: GATE2014-1-14 top<https://gateoverflow.in/1780>

Selected Answer

it would be $t_1 > t_2$, because the first case is the worst case of quicksort i.e. minimum number is chosen as pivot. Hence in the worst case the comparisons are high.

The splitting occurs as

[1]	[2345]
[2]	[345]
[3]	[45]
[4]	[5]

and

[123]	[45]		
[1]	[23]	[4]	[5]
[2]	[3]		

Number of recursive calls remain the same, but in second case the number of elements passed for the recursive call is less and hence the number of comparisons also less.

29 votes

-- Parul Agarwal (809 points)

1.16.39 Sorting: GATE2014-2-38 top<https://gateoverflow.in/1997>

Selected Answer

The optimal algorithm always chooses the smallest sequences for merging.

20 24 – 44, 43 comparisons

30 35 – 65, 64 comparisons

44 50 – 94, 93 comparisons

65 94 – 159, 158 comparisons

so, totally $43 + 64 + 93 + 158 = 358$ comparisons.

PS: In merge operation we do a comparison of two elements and put one element in the sorted output array. So, every comparison produces one output element. But for the last element we won't need a comparison and we simply insert it to the output array. So for n output elements we need $(n - 1)$ comparisons.

73 votes

-- Arjun Suresh (350k points)

1.16.40 Sorting: GATE2014-3-14 top<https://gateoverflow.in/2048>

Selected Answer

(A) $O(n^2)$ is the answer. When we choose the first element as the pivot, the worst case of quick sort comes if the input is sorted- either in ascending or descending order. Now, when we choose the middle element as pivot, sorted input no longer gives worst case behavior. But, there will be some permutation of the input numbers which will be giving the same worst case behavior. For example,

1 2 3 4 5 6 7

This array gives worst case behavior for quick sort when the first element is pivot.

6 4 2 1 3 5 7

This array gives the worst case behavior of $O(n^2)$ if we take middle element as the pivot- each split will be 1 element on one side and $n - 1$ elements on other side. Similarly, for any input, we can have a permutation where the behavior is like this. So, whichever element we take as pivot it gives worst case complexity of $O(n^2)$ as long as pivot is from a fixed position (not random position as in randomized quick sort).

45 votes

-- Arjun Suresh (350k points)

1.16.41 Sorting: GATE2015-1-2 top

<https://gateoverflow.in/8017>



B.

Worst case for quick sort happens when 1 element is on one list and $n - 1$ elements on another list.

26 votes

-- Arjun Suresh (350k points)

1.16.42 Sorting: GATE2015-2-45 top

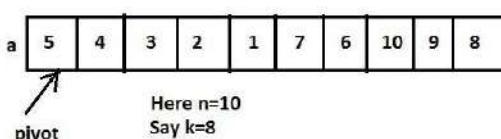
<https://gateoverflow.in/8243>



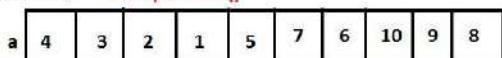
First of all, here the return value is the number of elements less than the pivot

Pivot is just to minimize searching

So, now we are assuming our array has 10 elements, $N = 10, k = 8$



STEP 1: after partition()



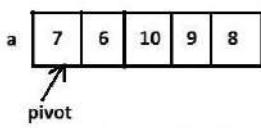
left_end=4

pivot

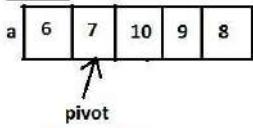
left_end+1<k

So, (a+left_end+1, n-left_end-1, k-left_end-1)

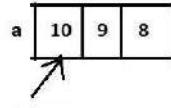
(a+5, 5, 3)



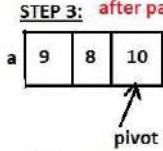
STEP 2: after partition()



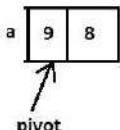
left_end=1
left_end+1<k
So, (a+2, 3, 1)



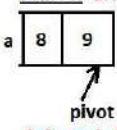
STEP 3: after partition()



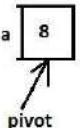
left_end=2
left_end+1>k
So, (a, 2, 1)



STEP 4: after partition()



left_end=1
left_end+1>k
So, (a,1,1)



STEP 5:



left_end=0
left_end+1==k
a[left_end]=8

So, in STEP 1 and STEP 2 'else' condition satisfying, and STEP 3 and STEP 4 'if' condition satisfying
Here, partition is calling and returning left_end value

Answer will be (A).

21 votes

-- srestha (87k points)

1.16.43 Sorting: GATE2015-3-27 top

<https://gateoverflow.in/8480>



The worst case time complexity of Mergesort is $k \times n \log n$ for an input of size n .

For an input of size 64, the algorithm takes 30s. Therefore,

$$k \times 64 \log_2 64 = 30s$$

$$k \times 384 = 30s$$

$$\Rightarrow k = 0.078125s$$

Let the size of the problem that can be solved in 6 minutes be x . Then,

$$k \times x \log_2 x = 360s$$

From this, we get:

$$\begin{aligned} x \log_2 x &= \frac{360s}{0.078125s} \\ \Rightarrow x &= 512 \end{aligned}$$

45 votes

-- Pragy Agarwal (20.6k points)

1.16.44 Sorting: GATE2016-1-13 top

<https://gateoverflow.in/39660>



Answer is D.

Insertion sort: $= \Theta(n^2)$

Merge sort: $= \Theta(n \log n)$

Quick sort: $= \Theta(n^2)$

Note : here Θ is not average case since question asked worst case so Θ represent worst case only

25 votes

-- Abhilash Panicker (9.5k points)

1.16.45 Sorting: GATE2016-2-13 top

<https://gateoverflow.in/39561>



$Q = Q$ can be used instead of Theta

- I. Quicksort takes $Q(N^2)$ in case of already sorted input. This is true
- II. This is false. If no swap happens then bubble sort can stop in single loop. $Q(N)$ is best case.
This is false !

- III. Mergesort never takes more than $O(N \log N)$ This is false
IV. This is true. Insertion sort will finish in $O(N)$ time in case of sorted input.

Answer **D**. I and IV.

Proof Bubble sort has best case $O(N) \Rightarrow$

Ref: https://en.wikipedia.org/wiki/Bubble_sort,_Aduni_lecture

Now quicksort taking

$O(N)$ can happen in some cases but not all cases, so that is why I) should be considered true. Whereas Bubble sort time complexity in best case is always $O(N)$. So D is any time stronger answer than C .

1 29 votes

-- Akash Kanase (42.5k points)

1.16.46 Sorting: ISI2011-A-2a top

<https://gateoverflow.in/48028>

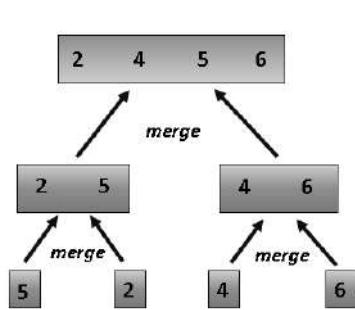
It will be merge sort.

a, b, c, d

1 + 1 comparisons for lowest step

+3 comparisons for upper one

So, total 5 comparisons



1 6 votes

-- Tanushree (385 points)

1.16.47 Sorting: TIFR2010-B-23 top

<https://gateoverflow.in/18623>



The given procedure resembles something like Bubble sort or Selection Sort. Every time, for every input it will take $O(n^2)$ so, **B** is the answer.

If you are thinking about heap sort, where in the procedure, it is talked about building a heap? You can do extract_max only on a heap. Right? The procedure is nearly same as bubble sort or selection sort.

Read qsn again: "First find the maximum. Remove this element from the list and find the

maximum of the remaining elements, remove this element, and so on, until all elements are exhausted."

Just follow the steps. Find max, take it to right side of array. Shift.

So $n - 1$ comparisons & $n - 1$ shifts in the worst case.

Total $n - 1 + n - 2 + n - 3 + \dots + 1 + n - 1 + n - 2 + n - 3 + \dots + 1$ which is $O(n^2)$

Only number of comparisons is asked... which is also $O(n^2)$

7 votes

-- Ahwan Mishra (10.5k points)

1.16.48 Sorting: TIFR2010-B-27 top

<https://gateoverflow.in/19036>



Selected Answer

In worst case Insertion sort will have $n(n - 1)/2$ comparisons i.e. when input is sorted in descending order.

50 40 30 20 10 ... n

pass 1 : 50 40 30 20 10 ... n 0 comparison

pass 2 : 40 50 30 20 10 ... n 1 comparison

.

.

.

pass n : n ... 10 20 30 40 50 $n - 1$ comparisons

Total $1 + 2 + 3 + \dots + n - 1 = n(n - 1)/2$ comparisons

7 votes

-- Umang Raman (15.7k points)

1.16.49 Sorting: TIFR2011-B-21 top

<https://gateoverflow.in/20324>



Selected Answer

store the elements in an array and then call build_heap(A). the build_heap takes $O(n)$ time.

so, option 'b' is correct.

but, if we try building heap by inserting each element one by one, the total complexity will be then $O(n \log n)$. cause insertion takes $O(\log n)$ and inserting ' n ' elements will take $O(n \log n)$.

17 votes

-- Sujit Kumar Muduli (209 points)

1.16.50 Sorting: TIFR2011-B-31 top

<https://gateoverflow.in/20617>

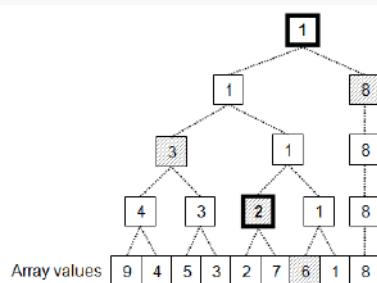


Selected Answer

Option **(c)** $n + k - 2$

Here is a nice explanation of the algorithm: <http://www.codinghelmet.com/?path=exercises/two-smallest>

it is solution to the problem known for ages, and it has to do with tennis tournaments. The question was, knowing the outcome of the tennis tournament, how can we tell which player was the second best? The defeated finalist is a good candidate, but there are other players that were defeated directly by the tournament winner and any of them could also be a good candidate for the second best. So the solution to the problem is quite simple: Once the tournament finishes, pick up the $\log N$ competitors that were beaten by the tournament winner and hold a mini-tournament to find which one is the best among them. If we imagine that better players correspond with smaller numbers, the algorithm now goes like this. Hold the tournament to find the smallest number (requires $N - 1$ comparisons). During this step, for each number construct the list of numbers it was smaller than. Finally, pick the list of numbers associated with the smallest number and find their minimum in $\log N - 1$ steps. This algorithm requires $N + \log N - 2$ comparisons to complete, but unfortunately it requires additional space proportional to N (each element except the winner will ultimately be added to someone's list); it also requires more time per step because of the relatively complex enlisting logic involved in each comparison. When this optimized algorithm is applied to example array, we get the following figure.



Tournament held among numbers promotes value 1 as the smallest number. That operation, performed on an array with nine numbers, requires exactly eight comparisons. While promoting the smallest number, this operation has also flagged four numbers that were removed from competition by direct comparison with the future winner: 6, 2, 3 and 8 in that order. Another sequence of three comparisons is required to promote number 2 as the second-smallest number in the array. This totals 11 comparisons, while naive algorithm requires 17 comparisons to come up with the same result.

All in all, this algorithm that minimizes number of comparisons looks to be good only for real tournaments, while number cracking algorithms should keep with the simple logic explained above. Implementation of simple algorithm may look like this:

```
a = array containing n elements
min1 = a[0] - candidate for the smallest value
min2 = a[1] - candidate for the second smallest value
if min2 < min1
    min1 = a[1]
    min2 = a[0]
for i = 2 to n - 1
    if a[i] < min1
        min2 = min1
        min1 = a[i]
    else if a[i] < min2
        min2 = a[i]
```

by Zoran Horvat [@zoranh75](#)

24 votes

-- Pragy Agarwal (20.6k points)

1.16.51 Sorting: TIFR2011-B-39 top

<https://gateoverflow.in/20935>



Selected Answer

(d)
 $O(\log n)$ comparisons suffice.

Since it is possible that $m \gg n$, we need to restrict ourselves to the first $O(n)$ elements to perform the binary search.

We start with the first element (index $i = 1$), and check if it is equal to 0. If not, we double the value of i , and check again. We repeat this process until we hit a 0.

```
i = 1;
while (arr[i] != 0)
    i *= 2;
```

Once we hit a 0, the largest possible value (worst case) of i can be $2n - 2$. This will happen if $n = 2^k + 1$ for some k . Then, our 2nd last value of i will be 2^k , and then we get 2^{k+1} , which is equal to $2n - 2$.

Now that we've hit a 0, and the array contains positive numbers in decreasing order, if x is present in L , it must be in the first i elements.

We can binary search the first i elements in $O(\log i)$ comparisons.

Since the largest possible value of $i = 2n - 2$, our algorithm takes $O(\log(2n - 2)) = O(\log n)$ comparisons.

15 votes

-- Pragy Agarwal (20.6k points)

1.16.52 Sorting: TIFR2012-B-13 top

<https://gateoverflow.in/25207>



Selected Answer

Let Array element be $\{4, 3, 2, 1, 7, 5, 6, 9, 10, 8\}$ and K be 3 here no element is more than 3 distance away from its final position

So if we take

$arr(1 \text{ to } 6)$ and sort then surely first three element will be sorted in its final position
 $\{12345769108\} O(6 \log 6)$

then sort $arr(3 \text{ to } 9)$ then 3 to 6 will be sorted $\{12345679108\} O(6 \log 6)$
 then at last $arr(6 \text{ to } 9)$ less than $O(6 \log 6)$ $\{12345678910\}$

in general

Sort $arr(0 \text{ to } 2k)$

Now we know that $arr[0 \text{ to } k]$ are in their final sorted positions
 and $arr(k \text{ to } 2k)$ may be not sorted.

Sort $arr(k \text{ to } 3k)$

Now we know that $arr[k \text{ to } 2k]$ are in their final sorted positions
 and $arr(2k \text{ to } 3k)$ may be not sorted.

.

sort till $arr(ik..N)$

in final sorting there will be less than $2k$ element.

in each step it will take $O(2k \log 2k)$
 and there will $\frac{n}{k}$ steps so $O(n \log k)$
 option D.

10 votes

-- Umang Raman (15.7k points)

1.16.53 Sorting: TIFR2012-B-14 top

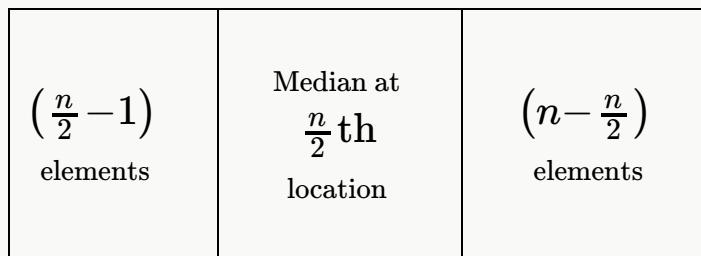
<https://gateoverflow.in/25209>



Selected Answer

Algorithm is choosing median = $n/2$ smallest element as pivot.

Hence, the array is divided as:



Therefore quick sort recurrence relation is given by:

$$\begin{aligned} T(n) &= T\left(\frac{n}{2} - 1\right) + T\left(n - \frac{n}{2}\right) + \Theta(n) \\ &= \Theta(n \log n) \end{aligned}$$

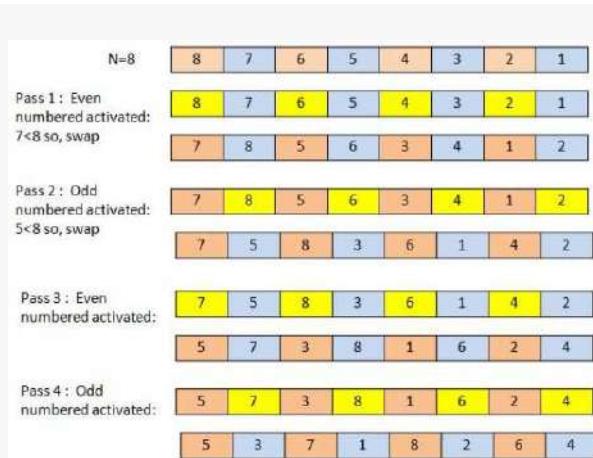
Hence, Option **B** is the correct answer.

15 votes

-- Umang Raman (15.7k points)

1.16.54 Sorting: TIFR2013-B-20 top

<https://gateoverflow.in/25878>



Answer: C.

10 votes

-- Vidhi Sethi (8.7k points)

1.16.55 Sorting: TIFR2017-B-7 [top](#)

<https://gateoverflow.in/95699>

A pairwise swap will make the sorted array unsorted. Hence, the option (B) is correct.

For eg - if an array is 1 2 3 4 5 6 7 8

The array will become after a pair wise swap to 2 1 4 3 6 5 8 7. For all i between 2 and $n - 1$, $a[i]$ is either lower, or either greater than their adjacent elements.

Since, each element is being swapped exactly once. The operation has $O(n)$ time complexity.

9 votes

-- tarun_svbk (1.5k points)

1.17

Spanning Tree(31) [top](#)

1.17.1 Spanning Tree: GATE1991-03,vi [top](#)

<https://gateoverflow.in/521>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Kruskal's algorithm for finding a minimum spanning tree of a weighted graph G with n vertices and m edges has the time complexity of:

- A. $O(n^2)$
- B. $O(mn)$
- C. $O(m + n)$
- D. $O(m \log n)$
- E. $O(m^2)$

[gate1991](#) [algorithms](#) [spanning-tree](#)**Answer**

1.17.2 Spanning Tree: GATE1992-01,ix [top](#)

<https://gateoverflow.in/549>

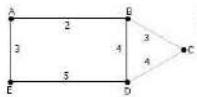
Complexity of Kruskal's algorithm for finding the minimum spanning tree of an undirected graph containing n vertices and m edges if the edges are sorted is _____

[gate1992](#) [spanning-tree](#) [algorithms](#) [time-complexity](#) [easy](#)**Answer**

1.17.3 Spanning Tree: GATE1995-22 [top](#)

<https://gateoverflow.in/2660>

How many minimum spanning trees does the following graph have? Draw them. (Weights are assigned to edges).



gate1995 algorithms graph-algorithms spanning-tree easy

Answer

1.17.4 Spanning Tree: GATE1996-16 [top](#)

<https://gateoverflow.in/2768>

A complete, undirected, weighted graph G is given on the vertex $\{0, 1, \dots, n-1\}$ for any fixed 'n'. Draw the minimum spanning tree of G if

- A. the weight of the edge (u, v) is $|u - v|$
- B. the weight of the edge (u, v) is $u + v$

gate1996 algorithms graph-algorithms spanning-tree normal

Answer

1.17.5 Spanning Tree: GATE1997-9 [top](#)

<https://gateoverflow.in/2269>

Consider a graph whose vertices are points in the plane with integer co-ordinates (x, y) such that $1 \leq x \leq n$ and $1 \leq y \leq n$, where $n \geq 2$ is an integer. Two vertices (x_1, y_1) and (x_2, y_2) are adjacent iff $|x_1 - x_2| \leq 1$ and $|y_1 - y_2| \leq 1$. The weight of an edge $\{(x_1, y_1), (x_2, y_2)\}$ is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

- A. What is the weight of a minimum weight-spanning tree in this graph? Write only the answer without any explanations.
- B. What is the weight of a maximum weight-spanning tree in this graph? Write only the answer without any explanations.

gate1997 algorithms spanning-tree normal

Answer

1.17.6 Spanning Tree: GATE2000-2.18 [top](#)

<https://gateoverflow.in/665>

Let G be an undirected connected graph with distinct edge weights. Let e_{max} be the edge with maximum weight and e_{min} the edge with minimum weight. Which of the following statements is false?

- A. Every minimum spanning tree of G must contain e_{min}
- B. If e_{max} is in a minimum spanning tree, then its removal must disconnect G
- C. No minimum spanning tree contains e_{max}
- D. G has a unique minimum spanning tree

gate2000 algorithms spanning-tree normal

Answer

1.17.7 Spanning Tree: GATE2001-15 [top](#)

<https://gateoverflow.in/756>

Consider a weighted undirected graph with vertex set $V = \{n1, n2, n3, n4, n5, n6\}$ and edge set

$E = \{(n1, n2, 2), (n1, n3, 8), (n1, n6, 3), (n2, n4, 4), (n2, n5, 12), (n3, n4, 7), (n4, n5, 9), (n4, n1, 1)\}$
The third value in each tuple represents the weight of the edge specified in the tuple.

- List the edges of a minimum spanning tree of the graph.
- How many distinct minimum spanning trees does this graph have?
- Is the minimum among the edge weights of a minimum spanning tree unique over all possible minimum spanning trees of a graph?
- Is the maximum among the edge weights of a minimum spanning tree unique over all possible minimum spanning tree of a graph?

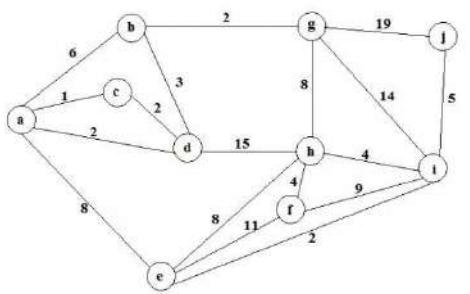
gate2001 algorithms spanning-tree normal descriptive

Answer

1.17.8 Spanning Tree: GATE2003-68 top

<https://gateoverflow.in/955>

What is the weight of a minimum spanning tree of the following graph?



- 29
- 31
- 38
- 41

gate2003 algorithms spanning-tree normal

Answer

1.17.9 Spanning Tree: GATE2005-6 top

<https://gateoverflow.in/1348>

An undirected graph G has n nodes. its adjacency matrix is given by an $n \times n$ square matrix whose (i) diagonal elements are 0's and (ii) non-diagonal elements are 1's. Which one of the following is TRUE?

- Graph G has no minimum spanning tree (MST)
- Graph G has unique MST of cost $n - 1$
- Graph G has multiple distinct MSTs, each of cost $n - 1$
- Graph G has multiple spanning trees of different costs

gate2005 algorithms spanning-tree normal

Answer

1.17.10 Spanning Tree: GATE2005-IT-52 top

<https://gateoverflow.in/3813>

Let G be a weighted undirected graph and e be an edge with maximum weight in G . Suppose there is a minimum weight spanning tree in G containing the edge e . Which of the following statements is always TRUE?

- A. There exists a cutset in G having all edges of maximum weight.
- B. There exists a cycle in G having all edges of maximum weight.
- C. Edge e cannot be contained in a cycle.
- D. All edges in G have the same weight.

gate2005-it algorithms spanning-tree normal

[Answer](#)

1.17.11 Spanning Tree: GATE2006-11 [top](#)

<https://gateoverflow.in/890>

Consider a weighted complete graph G on the vertex set $\{v_1, v_2, \dots, v_n\}$ such that the weight of the edge (v_i, v_j) is $2|i - j|$. The weight of a minimum spanning tree of G is:

- A. $n - 1$
- B. $2n - 2$
- C. $\binom{n}{2}$
- D. n^2

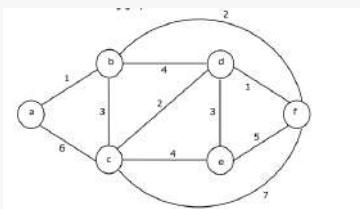
gate2006 algorithms spanning-tree normal

[Answer](#)

1.17.12 Spanning Tree: GATE2006-47 [top](#)

<https://gateoverflow.in/1823>

Consider the following graph:



Which one of the following cannot be the sequence of edges added, **in that order**, to a minimum spanning tree using Kruskal's algorithm?

- A. $(a - b), (d - f), (b - f), (d - c), (d - e)$
- B. $(a - b), (d - f), (d - c), (b - f), (d - e)$
- C. $(d - f), (a - b), (d - c), (b - f), (d - e)$
- D. $(d - f), (a - b), (b - f), (d - e), (d - c)$

gate2006 algorithms graph-algorithms spanning-tree normal

[Answer](#)

1.17.13 Spanning Tree: GATE2007-49 [top](#)

<https://gateoverflow.in/1247>

Let w be the minimum weight among all edge weights in an undirected connected graph. Let e be a specific edge of weight w . Which of the following is FALSE?

- A. There is a minimum spanning tree containing e
- B. If e is not in a minimum spanning tree T , then in the cycle formed by adding e to T , all edges have the same weight.
- C. Every minimum spanning tree has an edge of weight w
- D. e is present in every minimum spanning tree

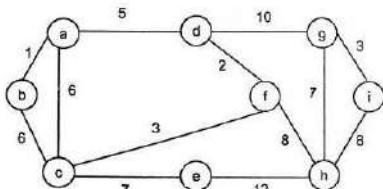
gate2007 algorithms spanning-tree normal

Answer

1.17.14 Spanning Tree: GATE2008-IT-45 [top](#)

<https://gateoverflow.in/3355>

For the undirected, weighted graph given below, which of the following sequences of edges represents a correct execution of Prim's algorithm to construct a Minimum Spanning Tree?



(Ctrl) +

- A. (a, b), (d, f), (f, c), (g, i), (d, a), (g, h), (c, e), (f, h)
- B. (c, e), (c, f), (f, d), (d, a), (a, b), (g, h), (h, f), (g, i)
- C. (d, f), (f, c), (d, a), (a, b), (c, e), (f, h), (g, h), (g, i)
- D. (h, g), (g, i), (h, f), (f, c), (f, d), (d, a), (a, b), (c, e)

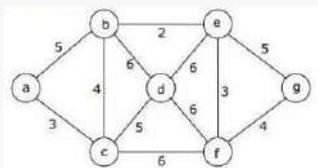
gate2008-it algorithms graph-algorithms spanning-tree normal

Answer

1.17.15 Spanning Tree: GATE2009-38 [top](#)

<https://gateoverflow.in/1324>

Consider the following graph:



Which one of the following is NOT the sequence of edges added to the minimum spanning tree using Kruskal's algorithm?

- A. (b, e) (e, f) (a, c) (b, c) (f, g) (c, d)
- B. (b, e) (e, f) (a, c) (f, g) (b, c) (c, d)
- C. (b, e) (a, c) (e, f) (b, c) (f, g) (c, d)
- D. (b, e) (e, f) (b, c) (a, c) (f, g) (c, d)

gate2009 algorithms spanning-tree normal

Answer**1.17.16 Spanning Tree: GATE2010-50** [top](#)<https://gateoverflow.in/2355>

Consider a complete undirected graph with vertex set $\{0, 1, 2, 3, 4\}$. Entry W_{ij} in the matrix W below is the weight of the edge $\{i, j\}$

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T ?

- A. 7
- B. 8
- C. 9
- D. 10

[gate2010](#) [algorithms](#) [spanning-tree](#) [normal](#)**Answer****1.17.17 Spanning Tree: GATE2010-51** [top](#)<https://gateoverflow.in/43328>

Consider a complete undirected graph with vertex set $\{0, 1, 2, 3, 4\}$. Entry W_{ij} in the matrix W below is the weight of the edge $\{i, j\}$

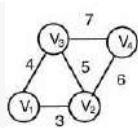
$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the minimum possible weight of a path P from vertex 1 to vertex 2 in this graph such that P contains at most 3 edges?

- A. 7
- B. 8
- C. 9
- D. 10

[gate2010](#) [normal](#) [algorithms](#) [spanning-tree](#)**Answer****1.17.18 Spanning Tree: GATE2011-54** [top](#)<https://gateoverflow.in/2162>

An undirected graph $G(V, E)$ contains n ($n > 2$) nodes named v_1, v_2, \dots, v_n . Two nodes v_i, v_j are connected if and only if $0 < |i - j| \leq 2$. Each edge (v_i, v_j) is assigned a weight $i + j$. A sample graph with $n = 4$ is shown below.



What will be the cost of the minimum spanning tree (MST) of such a graph with n nodes?

- A. $\frac{1}{12}(11n^2 - 5n)$
- B. $n^2 - n + 1$
- C. $6n - 11$
- D. $2n + 1$

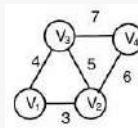
[gate2011](#) [algorithms](#) [graph-algorithms](#) [spanning-tree](#) [normal](#)

Answer

1.17.19 Spanning Tree: GATE2011-55 [top](#)

<https://gateoverflow.in/43325>

An undirected graph $G(V, E)$ contains n ($n > 2$) nodes named v_1, v_2, \dots, v_n . Two nodes v_i, v_j are connected if and only if $0 < |i - j| \leq 2$. Each edge (v_i, v_j) is assigned a weight $i + j$. A sample graph with $n = 4$ is shown below.



The length of the path from v_5 to v_6 in the MST of previous question with $n = 10$ is

- A. 11
- B. 25
- C. 31
- D. 41

[gate2011](#) [algorithms](#) [graph-algorithms](#) [spanning-tree](#) [normal](#)

Answer

1.17.20 Spanning Tree: GATE2012-29 [top](#)

<https://gateoverflow.in/786>

Let G be a weighted graph with edge weights greater than one and G' be the graph constructed by squaring the weights of edges in G . Let T and T' be the minimum spanning trees of G and G' , respectively, with total weights t and t' . Which of the following statements is **TRUE**?

- A. $T' = T$ with total weight $t' = t^2$
- B. $T' = T$ with total weight $t' < t^2$
- C. $T' \neq T$ but total weight $t' = t^2$
- D. None of the above

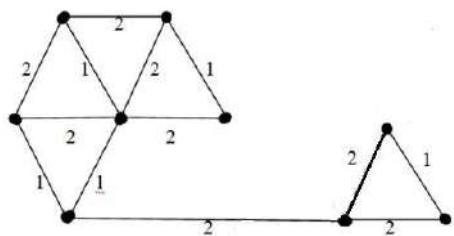
[gate2012](#) [algorithms](#) [spanning-tree](#) [normal](#) [marks-to-all](#)

Answer

1.17.21 Spanning Tree: GATE2014-2-52 [top](#)

<https://gateoverflow.in/2019>

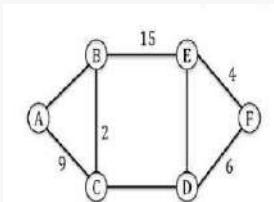
The number of distinct minimum spanning trees for the weighted graph below is _____


[gate2014-2](#) [algorithms](#) [spanning-tree](#) [numerical-answers](#) [normal](#)
Answer

1.17.22 Spanning Tree: GATE2015-1-43 [top](#)

<https://gateoverflow.in/8313>

The graph shown below has 8 edges with distinct integer edge weights. The minimum spanning tree (**MST**) is of weight 36 and contains the edges: $\{(A,C), (B,C), (B,E), (E,F), (D,F)\}$. The edge weights of only those edges which are in the **MST** are given in the figure shown below. The minimum possible sum of weights of all 8 edges of this graph is _____.


[gate2015-1](#) [algorithms](#) [spanning-tree](#) [normal](#) [numerical-answers](#)
Answer

1.17.23 Spanning Tree: GATE2015-3-40 [top](#)

<https://gateoverflow.in/8499>

Let G be a connected undirected graph of 100 vertices and 300 edges. The weight of a minimum spanning tree of G is 500. When the weight of each edge of G is increased by five, the weight of a minimum spanning tree becomes _____.

[gate2015-3](#) [algorithms](#) [spanning-tree](#) [easy](#) [numerical-answers](#)
Answer

1.17.24 Spanning Tree: GATE2016-1-14 [top](#)

<https://gateoverflow.in/39673>

Let G be a weighted connected undirected graph with distinct positive edge weights. If every edge weight is increased by the same value, then which of the following statements is/are TRUE?

- P : Minimum spanning tree of G does not change.
 - Q : Shortest path between any pair of vertices does not change.
- P only
 - Q only
 - Neither P nor Q
 - Both P and Q

[gate2016-1](#) [algorithms](#) [spanning-tree](#) [normal](#)
Answer

1.17.25 Spanning Tree: GATE2016-1-39 [top](#)

<https://gateoverflow.in/39725>

Let G be a complete undirected graph on 4 vertices, having 6 edges with weights being 1, 2, 3, 4, 5, and 6. The maximum possible weight that a minimum weight spanning tree of G can have is _____.

[gate2016-1](#) [algorithms](#) [spanning-tree](#) [normal](#) [numerical-answers](#)

Answer

1.17.26 Spanning Tree: GATE2016-1-40 [top](#)

<https://gateoverflow.in/39727>

$G = (V, E)$ is an undirected simple graph in which each edge has a distinct weight, and e is a particular edge of G . Which of the following statements about the minimum spanning trees ($MSTs$) of G is/are TRUE?

- I. If e is the lightest edge of some cycle in G , then every MST of G includes e .
 - II. If e is the heaviest edge of some cycle in G , then every MST of G excludes e .
- A. I only.
 B. II only.
 C. Both I and II.
 D. Neither I nor II.

[gate2016-1](#) [algorithms](#) [spanning-tree](#) [normal](#)

Answer

1.17.27 Spanning Tree: TIFR2011-B-35 [top](#)

<https://gateoverflow.in/20842>

Let G be a connected simple graph (no self-loops or parallel edges) on $n \geq 3$ vertices, with distinct edge weights. Let e_1, e_2, \dots, e_m be an ordering of the edges in decreasing order of weight. Which of the following statements is FALSE?

- A. The edge e_1 has to be present in every maximum weight spanning tree.
- B. Both e_1 and e_2 have to be present in every maximum weight spanning tree.
- C. The edge e_m has to be present in every minimum weight spanning tree.
- D. The edge e_m is never present in any maximum weight spanning tree.
- E. G has a unique maximum weight spanning tree.

[tifr2011](#) [algorithms](#) [graph-algorithms](#) [spanning-tree](#)

Answer

1.17.28 Spanning Tree: TIFR2013-B-17 [top](#)

<https://gateoverflow.in/25860>

In a connected weighted graph with n vertices, all the edges have distinct positive integer weights. Then, the maximum number of minimum weight spanning trees in the graph is

- a. 1
- b. n
- c. equal to number of edges in the graph.
- d. equal to maximum weight of an edge of the graph.
- e. n^{n-2}

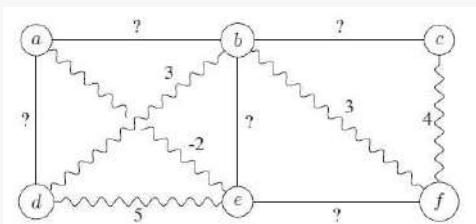
[tifr2013](#) [spanning-tree](#)

Answer

1.17.29 Spanning Tree: TIFR2014-B-4 [top](#)

<https://gateoverflow.in/27174>

Consider the following undirected graph with some edge costs missing.



Suppose the wavy edges form a Minimum Cost Spanning Tree for G . Then, which of the following inequalities NEED NOT hold?

- a. $\text{cost}(a, b) \geq 6$.
- b. $\text{cost}(b, e) \geq 5$.
- c. $\text{cost}(e, f) \geq 5$.
- d. $\text{cost}(a, d) \geq 4$.
- e. $\text{cost}(b, c) \geq 4$.

[tifr2014](#) [algorithms](#) [graph-algorithms](#) [spanning-tree](#)

Answer

1.17.30 Spanning Tree: TIFR2014-B-5 [top](#)

<https://gateoverflow.in/27180>

Let $G = (V, E)$ be an undirected connected simple (i.e., no parallel edges or self-loops) graph with the weight function $w : E \rightarrow \mathbb{R}$ on its edge set. Let $w(e_1) < w(e_2) < \dots < w(e_m)$, where $E = \{e_1, e_2, \dots, e_m\}$. Suppose T is a minimum spanning tree of G . Which of the following statements is FALSE?

- A. The tree T has to contain the edge e_1 .
- B. The tree T has to contain the edge e_2 .
- C. The minimum weight edge incident on each vertex has to be present in T .
- D. T is the unique minimum spanning tree in G .
- E. If we replace each edge weight $w_i = w(e_i)$ by its square w_i^2 , then T must still be a minimum spanning tree of this new instance.

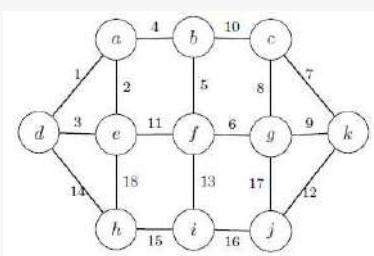
[tifr2014](#) [algorithms](#) [spanning-tree](#)

Answer

1.17.31 Spanning Tree: TIFR2015-B-2 [top](#)

<https://gateoverflow.in/29844>

Consider the following undirected connected graph G with weights on its edges as given in the figure below. A minimum spanning tree is a spanning tree of least weight and a maximum spanning tree is one with largest weight. A second best minimum spanning tree whose weight is the smallest among all spanning trees that are not minimum spanning trees in G .



Which of the following statements is TRUE in the above graph? (Note that all the edge weights are distinct in the above graph)

- A. There is more than one minimum spanning tree and similarly, there is more than one maximum spanning tree here.
- B. There is a unique minimum spanning tree, however there is more than one maximum spanning tree here.
- C. There is more than one minimum spanning tree, however there is a unique maximum spanning tree here.
- D. There is more than one minimum spanning tree and similarly, there is more than one second-best minimum spanning tree here.
- E. There is unique minimum spanning tree, however there is more than one second-best minimum spanning tree here.

[tifr2015](#) [spanning-tree](#) [algorithms](#) [graph-algorithms](#)

Answer

Answers: Spanning Tree

1.17.1 Spanning Tree: GATE1991-03,vi top

<https://gateoverflow.in/521>



Selected Answer

Answer: **D, B, E.**

When Union-Find algorithm is used to detect cycle while constructing the MST time complexity is $O(m \log n)$ where m is the number of edges, and n is the number of vertices. Since $n = O(m^2)$ in a graph, options B and E are also correct as big-O specifies asymptotic upper bound only.

Reference: <http://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/>

13 votes

-- Rajarshi Sarkar (34.1k points)

1.17.2 Spanning Tree: GATE1992-01,ix top

<https://gateoverflow.in/549>



Selected Answer

If all edges are already sorted then this problem will reduced to union-find problem on a graph with E edges and V vertices.

```
for each edge (u, v) in E
    if (FIND-SET(u) != FIND-SET(v))
        UNION(u, v)
```

$\text{FIND-SET}(v)$ and $\text{UNION}(u, v)$ runs in $\alpha(|V|)$

where $\alpha(n)$ is inverse ackermann function i.e $\log^*(n)$

So, overall complexity becomes $O(|E| \cdot \alpha(|V|))$

21 votes

-- Vikrant Singh (13.5k points)

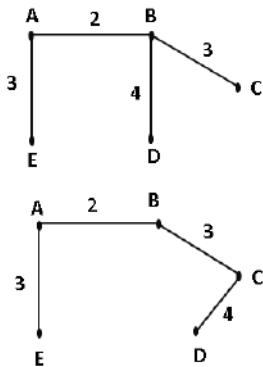
1.17.3 Spanning Tree: GATE1995-22 top

<https://gateoverflow.in/2660>



2 only.

$\{AB, BC, AE, BD\}$ and $\{AB, BC, AE, CD\}$.



1 21 votes

-- Gate Keeda (19.6k points)

1.17.4 Spanning Tree: GATE1996-16 top

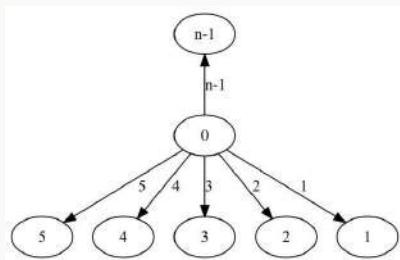
<https://gateoverflow.in/2768>



(A).



(B).



1 19 votes

-- Anu (5.9k points)

1.17.5 Spanning Tree: GATE1997-9 top

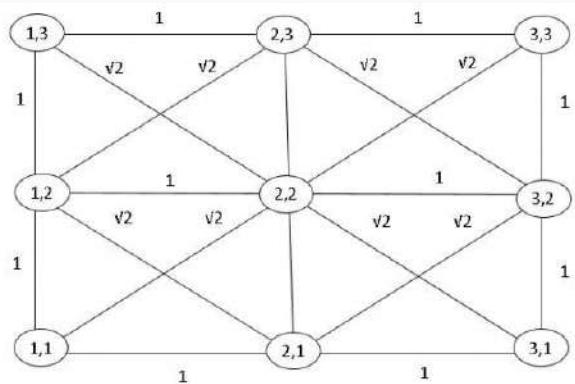
<https://gateoverflow.in/2269>



Consider $n = 3$, we have 9 vertices in the plane. The vertices are:

$(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)$

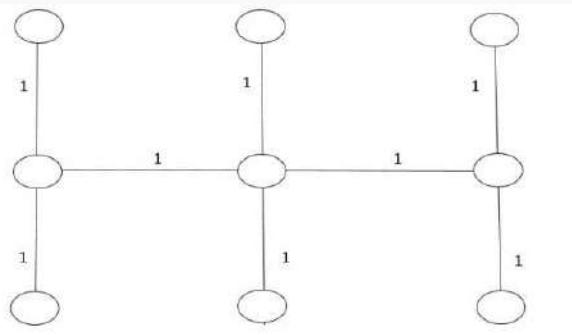
So, the corresponding graph will be:



The cost of the edge between the vertices $(2, 3)$ and $(2, 2) = 1$ and the cost of the edge between the vertices $(2, 2)$ and $(2, 1) = 1$.

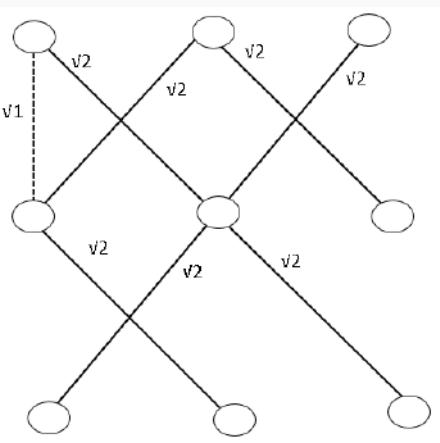
Minimum spanning tree:

(One of the possibilities)



There is no problem with minimum spanning tree. We have n^2 vertices so, the minimum spanning tree contains $n^2 - 1$ edges of cost 1. So, the cost of the minimum spanning tree is $n^2 - 1$.

Maximum spanning tree:



In this maximum spanning tree of $n^2 - 1$ edges $n^2 - 2$ edges is of cost $\sqrt{2}$ and 1 edge is of cost

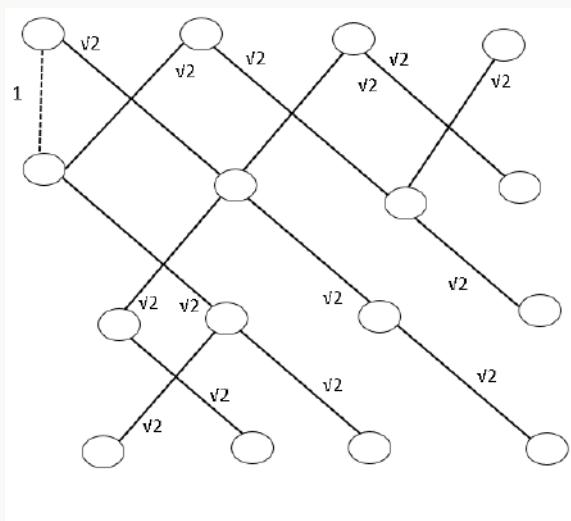
1.

So, the answer is $\sqrt{2} (n^2 - 2) + 1$.

This pattern continues for high values of n .

For $n = 4$, the maximum spanning tree is:

(one of possibilities)



For $n = 4, 16$ vertices are there and the maximum spanning tree requires 15 edges. Out of 15 edges, 14 edges are of cost $\sqrt{2}$ and 1 edge is of cost 1. Thus satisfying the formula $\sqrt{2} (n^2 - 2) + 1$.

The cost of Minimum spanning tree is: $n^2 - 1$.

The cost of Maximum spanning tree is: $(n^2 - 1) \sqrt{2} + 1$.

16 votes

-- balaeinstein (1.3k points)

1.17.6 Spanning Tree: GATE2000-2.18 top

<https://gateoverflow.in/665>



C the case should be written as "may or may not", to be true.

D will always be true as per the question saying that the graph has distinct weights.

20 votes

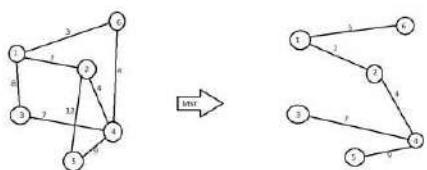
-- Gate Keeda (19.6k points)

1.17.7 Spanning Tree: GATE2001-15 top

<https://gateoverflow.in/756>



a) edges with weight: 2,3,4,7,9



- b) no of distinct minimum spanning tree: 2 (2nd with the different edge of weight 4)
 c) yes
 d) yes

Edit:- Combining both answers into 1.

14 votes

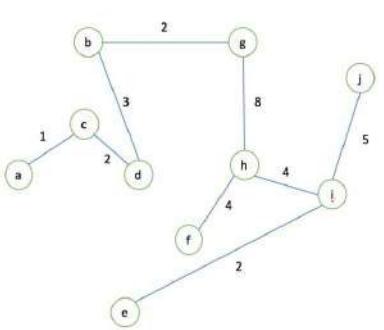
-- jayendra (8.2k points)

1.17.8 Spanning Tree: GATE2003-68 top

<https://gateoverflow.in/955>



Apply Prim's algorithm, start from A as shown in figure below.



add all the weights in the given figure which would be equal to 31.

8 votes

-- Monanshi Jain (9.3k points)

1.17.9 Spanning Tree: GATE2005-6 top

<https://gateoverflow.in/1348>



Graph G has multiple distinct MSTs, each of cost $n - 1$

From the given data given graph is a complete graph with all edge weights 1. A MST will contain $n - 1$ edges. Hence weight of MST is $n - 1$.

The graph will have multiple MST. In fact all spanning trees of the given graph will be MSTs also since all edge weights are equal.

17 votes

-- Sankaranarayanan P.N (11.5k points)

1.17.10 Spanning Tree: GATE2005-IT-52 top

<https://gateoverflow.in/3813>



Option A is correct.

Question says the MST of graph G contain an edge e which is a maximum weight edge in G . Need to choose the answer which is always true to follow the above constraint.

Case 1:

Option B says that if edge e is in MST then for sure there is a cycle having all edges of maximum weight. But it is not true always because when there is only $n-1$ edges(but no cycle) in graph then also maximum edge has to be taken for MST.

Case 2:

Option C says otherwise. That if e is in MST then it cannot be in any cycle that is wrong as if there is a cycle with all maximum edges then also e will be in MST

Option D says all edges should be of same weight same explanation if there are $n - 1$ distinct edges(but no cycle) in G then have to take all edges including maximum weight edge.

And at last option A says if e is in MST then for sure there is a cut-set (minimum edge set whose removal disconnects the graph) in G having all edges of maximum weight. And it is true.

Because then only we maximum weight edges has to be taken in MST.

For eg. If there are $n - 1$ edges (but no cycle) then if edge e is not taken in the MST then MST will not be connected.

12 votes

-- khush tak (7.6k points)

1.17.11 Spanning Tree: GATE2006-11 top

<https://gateoverflow.in/890>



$2(n - 1)$ the spanning tree will traverse adjacent edges since they contain the least weight.

22 votes

-- anshu (3.3k points)

1.17.12 Spanning Tree: GATE2006-47 top

<https://gateoverflow.in/1823>



In Kruskal's algo the edges are added in non decreasing order of their weight. But in Option D edge $d - e$ with weight 3 is added before edge $d - c$ with weight 2. Hence, option D is wrong option.

14 votes

-- Sankaranarayanan P.N (11.5k points)

1.17.13 Spanning Tree: GATE2007-49 top

<https://gateoverflow.in/1247>



D is the false statement.

A minimum spanning tree must have the edge with the smallest weight (In Kruskal's algorithm we start from the smallest weight edge). So, **C** is TRUE.

If e is not part of a minimum spanning tree, then all edges which are part of a cycle with e , must have weight $\leq e$, as otherwise we can interchange that edge with e and get another minimum spanning tree of lower weight. So, **B** and **A** are also TRUE.

16 votes

-- Arjun Suresh (350k points)

1.17.14 Spanning Tree: GATE2008-IT-45 top

<https://gateoverflow.in/3355>



Prim's algorithm starts with any vertex and expands the MST by adding one vertex in each step which is close to the Intermediate MST(made till previous step).

Therefore, correct answer would be (**C**).

(A): (d, f) is chosen but neither d nor f vertices are part of the previous MST(MST made till previous step).

(B): (g, h) is chosen but neither g or h vertices are part of the previous MST(MST made till previous step).

(D): (f, c) is chosen but at that point (f, d) is close to the intermediate MST.

16 votes

-- suraj (5.6k points)

1.17.15 Spanning Tree: GATE2009-38 top

<https://gateoverflow.in/1324>



In **Option D** $b-c$ with weight, 4 is added before $a-c$ with weight 3 is added. In Kruskal's algorithm, edges should be added in non-decreasing order of weight.

So, **Option D** may be correct.

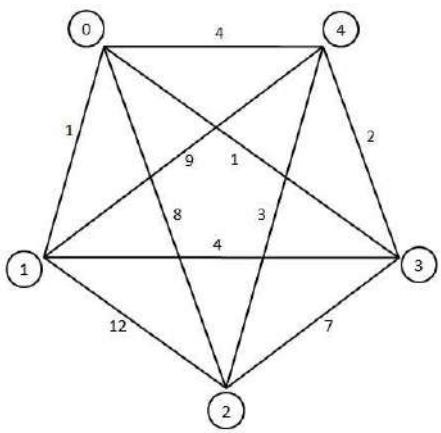
14 votes

-- Sankaranarayanan P.N (11.5k points)

1.17.16 Spanning Tree: GATE2010-50 top

<https://gateoverflow.in/2355>





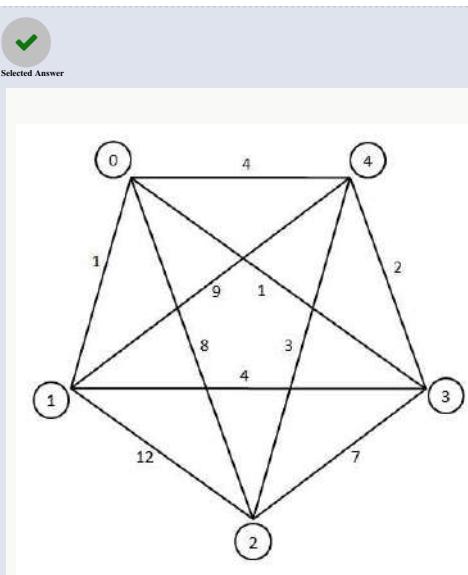
Answer is (D) 10. The edges of the spanning tree are: $0 - 1, 1 - 3, 3 - 4, 4 - 2$. Total Weight = 10

26 votes

-- Ashis Kumar Sahoo (865 points)

1.17.17 Spanning Tree: GATE2010-51 top

<https://gateoverflow.in/43328>



Answer is (B) 8. The possible path is: $1 - 0, 0 - 4, 4 - 2$.

20 votes

-- Ashis Kumar Sahoo (865 points)

1.17.18 Spanning Tree: GATE2011-54 top

<https://gateoverflow.in/2162>



Q 54. Answer is B.

We observe a pattern in the weight of MST being formed

For $n=3$ $(1 + 2 + 3) + (1)$

For $n=4 (1 + 2 + 3 + 4) + (1 + 2)$

For $n=5 (1 + 2 + 3 + 4 + 5) + (1 + 2 + 3)$

These can be obtained by drawing graphs for these graphs.

\therefore Total weight of MST is $\sum_{i=1}^n i + \sum_{i=1}^{n-2} i = n^2 - n + 1$

1 21 votes

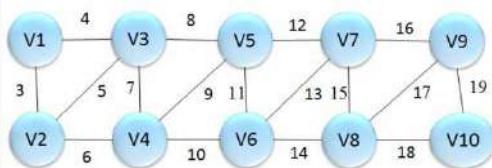
-- Sona Praneeth Akula (4.2k points)

1.17.19 Spanning Tree: GATE2011-55 top

<https://gateoverflow.in/43325>

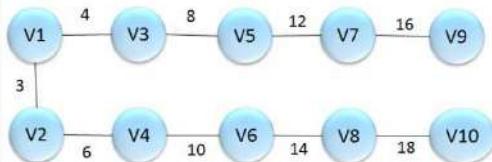


Selected Answer



Above is the graph....

Below is the MST.



Length of the path from v_5 to $v_6 = 8 + 4 + 3 + 6 + 10 = 31$ (Answer)

1 26 votes

-- Ahwan Mishra (10.5k points)

1.17.20 Spanning Tree: GATE2012-29 top

<https://gateoverflow.in/786>



Selected Answer

When the edge weights are squared the minimum spanning tree won't change.

$t' < t^2$, because sum of squares is always less than the square of the sums except for a single element case.

Hence, B is the general answer and A is also true for a single edge graph. Hence, in GATE 2012, marks were given to all.

1 37 votes

-- gatecse (18k points)

1.17.21 Spanning Tree: GATE2014-2-52 top

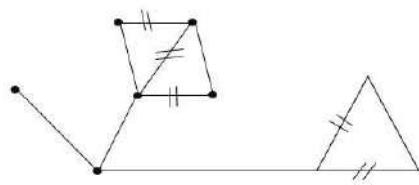
<https://gateoverflow.in/1919>



Selected Answer

6 is the answer.

$2 \times 3 = 6$ possibilities



1 31 votes

-- Arjun Suresh (350k points)

1.17.22 Spanning Tree: GATE2015-1-43 top

<https://gateoverflow.in/3313>



Selected Answer

Consider the cycle ABC . AC and AB are part of minimum spanning tree. So, AB should be greater than $\max(AC, BC)$ (greater and not equal as edge weights are given to be distinct), as otherwise we could add AB to the minimum spanning tree and removed the greater of AC, BC and we could have got another minimum spanning tree. So, $AB > 9$.

Similarly, for the cycle DEF , $ED > 6$.

And for the cycle $BCDE$, $CD > 15$.

So, minimum possible sum of these will be $10 + 7 + 16 = 33$. Adding the weight of spanning tree, we get the total sum of edge weights

$$= 33 + 36 = 69$$

1 51 votes

-- Arjun Suresh (350k points)

1.17.23 Spanning Tree: GATE2015-3-40 top

<https://gateoverflow.in/8499>



Selected Answer

First find no of edges in mst.

It has $n - 1$ edges where n is no of vertices. $100 - 1 = 99$ edges

Each 99 edges in mst increases by 5 so weight in mst increased $99 * 5 = 495$

Now total weight of mst = $500 + 495 = 995$

1 38 votes

-- Anoop Sonkar (5k points)

1.17.24 Spanning Tree: GATE2016-1-14 top

<https://gateoverflow.in/39673>



Selected Answer

Statement P is true.

For statement Q consider a simple graph with 3 nodes.

A	B	C	
A	0	1	100
B	1	0	2
C	100	2	0

Shortest path from A to C is $A-B-C = 1 + 2 = 3$

Now if the value of each edge is increased by 100,

	A	B	C
A	0	101	200
B	101	0	102
C	200	102	0

The shortest path from A to C is A-C = 200, (A-B-C = $101 + 102 = 203$)

Hence, option **A is correct.**

1 51 votes

-- ryan sequeira (3.4k points)

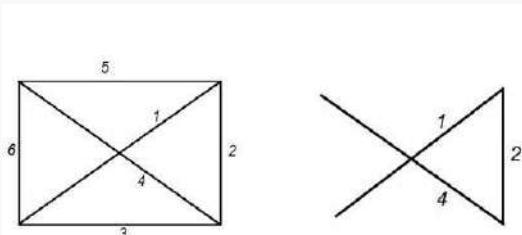
1.17.25 Spanning Tree: GATE2016-1-39 top

<https://gateoverflow.in/39725>



Selected Answer

Graph G can be like this:



Weight: $1 + 2 + 4 = 7$

1 45 votes

-- shaiklam09 (237 points)

1.17.26 Spanning Tree: GATE2016-1-40 top

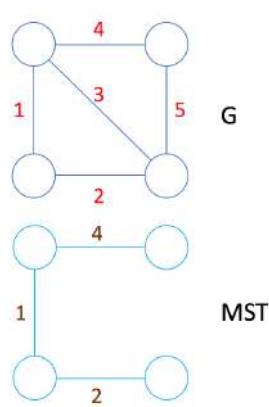
<https://gateoverflow.in/39727>



Selected Answer

Statement 1:- False by [Cut Property of MST]

See counter example :- Here in below Graph G in (cycle 3, 4, 5) 3 is the lightest edge and still it is not included in MST.



Statement 2: True by [Cycle property of MST] : (in above Graph G $1 - 2 - 3$ is a cycle and 3 is the heaviest edge) If heaviest edge is in cycle then we will always exclude that bcz Cycle is their means we can have other choice of low cost edges.

So, option **B** is answer.

Must visit Links:

<http://www.cs.princeton.edu/courses/archive/spr07/cos226/lectures/mst.pdf>

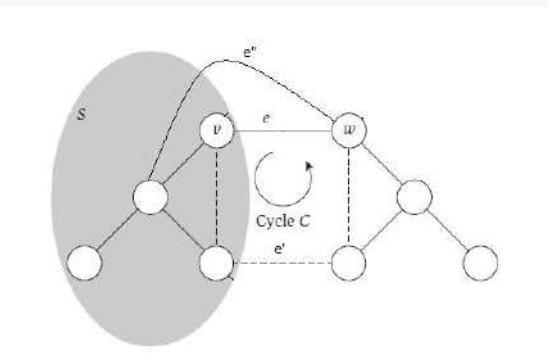
Up 52 votes

-- Rajesh Pradhan (22.6k points)

1. For an edge to be INCLUDED in the MST, it must satisfy the **CUT PROPERTY** which states that

Assume that all edge costs are distinct. Let S be any subset of nodes that is neither empty nor equal to all of V , and let edge $e = (v, w)$ be the minimum cost edge with one end in S and the other in $V - S$. Then every minimum spanning tree contains the edge e .

Now, in the given problem, e is the lightest edge of some cycle in G , but it may not be the lightest edge connecting S and $V - S$. See the below image for clarification.



Here, **e** is the lightest edge of cycle **C**. There are three edges e, e', e'' connecting S and $V - S$.

e is obviously minimal weighted than e' since e' is part of the cycle **C**, but **e MAY NOT** be minimal weighted than e'' .

So, every MST of G **MAY NOT** include e .

2. For an edge to be EXCLUDED from the MST, it must satisfy the **CYCLE PROPERTY** which states that

Assume that all edge costs are distinct. Let C be any cycle in G , and let edge $e = (v, w)$ be the most expensive edge belonging to C . Then e does not belong to the minimum spanning tree of G .

Now, in the given problem, e is the heaviest edge of some cycle C .

So, every MST of G **MUST** exclude e .

ANSWER is B.

NOTE: For proofs of the above two properties, one can refer to the source of this answer:
Algorithm Design by Jon Kleinberg and Eva Tardos

Up 29 votes

-- Tanaya Pradhan (625 points)

1.17.27 Spanning Tree: TIFR2011-B-35 top

<https://gateoverflow.in/20842>



a) & c) are trivially true. Edge with max value e_1 must be present in Maximum spanning tree & same with minimum.

e) This is true, because all edge weights are distinct. maximum spanning tree is unique.

b) e_1 & e_2 must be present in Maximum spanning tree. I'll prove it using Kruskal Algorithm.

We will first insert weight with biggest value, e_1 . Then we insert e_2 (second highest). 2 edges do not create **cycle**. Then we can go on from there inserting edges according to edge weights. As they have just asked for top 2 edges, using Kruskal Algo we can say that top 2 edges must be in Maximum spanning tree.

d) This is false. There are chances that this e_m weight edge is **cut edge(Bridge)** Then it must be inserted to from any spanning tree.

D is answer.

We can not say the same for Top 3 as they can create cycle & They we can not take a_3 to make spanning tree.

Kruskal Algo Reference: <http://stackoverflow.com/questions/4992664/how-to-find-maximum-spanning-tree>

7 votes

-- Akash Kanase (42.5k points)

1.17.28 Spanning Tree: TIFR2013-B-17 top

<https://gateoverflow.in/25860>



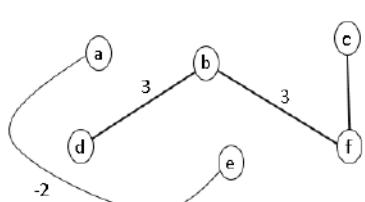
There will be unique min weight spanning tree since all weights are distinct.
Option is **A**.

10 votes

-- Umang Raman (15.7k points)

1.17.29 Spanning Tree: TIFR2014-B-4 top

<https://gateoverflow.in/27174>



Now check this diagram, this is forest obtained from above given graph using Kruskal's algorithm for MST.

So, according to the question edge $d - e$ has weight 5 and it is included in the formation of MST. Now if edges $b - e$ and $e - f$ has weight greater than 5 than it is not a problem for our MST because still we will get the given tree as Kruskal's algorithm takes the smallest weighted edge without forming a cycle.

Cost of edge $b - c \geq 4$ may also lead us to the same tree as above though Kruskal's algorithm will have choice between $c - f$ and $b - c$.

Now if the edge weight of $a - d$ becomes 4, it is guaranteed that Kruskal's algorithm will not select edge $d - e$ because its edge cost is 5, and hence the tree structure will change. But there can be the case where edge weight is greater than 4 and we still get the same tree (happens when $a - d \geq 5$). Because in the question they asked to point out an unnecessary condition this case is not the answer as we need $a - d \geq 5$ which implies $a - d \geq 4$.

Now notice option A. Put $a - b = 5$. The given MST would not change. So, this condition is not always necessary and hence is the answer.

Therefore, option **A** is the answer.

14 votes

-- Riya Roy(Arayana) (7.2k points)

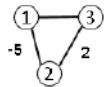
1.17.30 Spanning Tree: TIFR2014-B-5 top

<https://gateoverflow.in/27180>



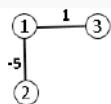
Selected Answer

Answer is **E**. The catch here is edge-weights belongs to real number. Therefore, edge weight can be negative. In that case the minimum spanning tree may be different.



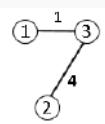
- $e_1 = -5$
- $e_2 = 1$
- $e_3 = 2$

T is:



New edge weights:

- $e_1 = 25$
- $e_2 = 1$
- $e_3 = 4$



EDIT:

(Here every edge weight is distinct, therefore MST is unique.)

Option A is True. If we apply Kruskal's algorithm, it will choose e_1

Option B is True. If we apply Kruskal's algorithm then it will also choose e_2 , and 2 edges can not form a cycle. (e_3 is not guaranteed in MST, as it may form a cycle)

Option C is also true. If we apply Prims algorithm also on any vertex (say u), it chooses minimum weight edge incident on vertex u.



Option D is true. Because every edge weight is distinct.

1 23 votes

-- Riya Roy(Arayana) (7.2k points)

1.17.31 Spanning Tree: TIFR2015-B-2 top

<https://gateoverflow.in/29844>



Selected Answer

In the graph we have all edge weights are distinct so we will get unique minimum and maximum spanning tree.

Each Cycle must exclude maximum weight edge in minimum spanning tree.

Here, we have two cycle of 3 edges, $a - d - e$ and $c - g - k$.

For second best minimum spanning tree, exclude $a - e$ edge and include $d - e$ edge

Other way for second best minimum spanning tree: exclude $c - g$ edge and include $g - k$ edge.

So, **e** should be the answer.

1 12 votes

-- papesh (25.7k points)

1.18

Time Complexity(34) top

1.18.1 Time Complexity: CMI2013-A-10 top

<https://gateoverflow.in/46602>

The below question is based on following program:

```
procedure mystery (A : array [1..100] of int)
    int i,j,position,tmp;
    begin
        for j := 1 to 100 do
            position := j;
            for i := j to 100 do
                if (A[i] > A[position]) then
                    position := i;
                endfor
                tmp := A[j];
                A[j] := A[position];
                A[position] := tmp;
            endfor
    end
```

The number of times the test $A[i] > A[position]$ is executed is:

- A. 100
- B. 5050
- C. 10000

- D. Depends on contents of A

[cmi2013](#) [algorithms](#) [time-complexity](#)

Answer

1.18.2 Time Complexity: CMI2015-A-08 [top](#)

<https://gateoverflow.in/47045>

How many times is the comparison $i \geq n$ performed in the following program?

```
int i=85, n=5;
main() {
    while (i >= n) {
        i=i-1;
        n=n+1;
    }
}
```

- A. 40
- B. 41
- C. 42
- D. 43

[cmi2015](#) [algorithms](#) [time-complexity](#)

Answer

1.18.3 Time Complexity: CMI2017-B-8 [top](#)

<https://gateoverflow.in/203320>

Consider the following function that takes as input a sequence A of integers with n elements, $A[A_1], [A_2], \dots, [A_n]$ and an integer k and returns an integer value. The function $\text{length}(S)$ returns the length of sequence S . Comments start with $//$.

```
function mystery(A, k) {
n = length(A);
if (k > n) return A[n];
v = A[1];
AL = [ A[j] : 1 <= j <= n, A[j] < v ]; // AL has elements < v in A
Av = [ A[j] : 1 <= j <= n, A[j] == v ]; // Av has elements = v in A
AR = [ A[j] : 1 <= j <= n, A[j] > v ]; // AR has elements > v in A
if (length(AL) >= k) return mystery(AL,k);
if (length(AL) + length(Av) >= k) return v;
return mystery(AR, k - (length(AL) + length(Av)));
}
```

- A. Explain what the function computes.
- B. What is the worst-case complexity of this algorithm in terms of the length of the input sequence A ?
- C. Give an example of a worst-case input for this algorithm.

[cmi2017](#) [algorithms](#) [time-complexity](#) [descriptive](#)

Answer

1.18.4 Time Complexity: GATE1989-2-iii [top](#)

<https://gateoverflow.in/87080>

Match the pairs in the following questions:

- (A) $O(\log n)$ (p) Heapsort

- (B) $O(n)$ (q) Depth-first search
 (C) $O(n \log n)$ (r) Binary search
 (D) $O(n^2)$ (s) Selection of the k^{th} smallest element in a set of n elements.

gate1989 match-the-following algorithms time-complexity

[Answer](#)

1.18.5 Time Complexity: GATE1993-8.7 [top](#)

<https://gateoverflow.in/2305>

$\sum_{1 \leq k \leq n} O(n)$, where $O(n)$ stands for order n is:

- A. $O(n)$
- B. $O(n^2)$
- C. $O(n^3)$
- D. $O(3n^2)$
- E. $O(1.5n^2)$

gate1993 algorithms time-complexity easy

[Answer](#)

1.18.6 Time Complexity: GATE1999-1.13 [top](#)

<https://gateoverflow.in/1466>

Suppose we want to arrange the n numbers stored in any array such that all negative values occur before all positive ones. Minimum number of exchanges required in the worst case is

- A. $n - 1$
- B. n
- C. $n + 1$
- D. None of the above

gate1999 algorithms time-complexity normal

[Answer](#)

1.18.7 Time Complexity: GATE1999-1.16 [top](#)

<https://gateoverflow.in/1469>

If n is a power of 2, then the minimum number of multiplications needed to compute a^n is

- A. $\log_2 n$
- B. \sqrt{n}
- C. $n - 1$
- D. n

gate1999 algorithms time-complexity normal

[Answer](#)

1.18.8 Time Complexity: GATE1999-11a [top](#)

<https://gateoverflow.in/1510>

Consider the following algorithms. Assume, procedure A and procedure B take $O(1)$ and $O(1/n)$ unit of time respectively. Derive the time complexity of the algorithm in O -notation.

```
algorithm what (n)
begin
  if n = 1 then call A
  else
    begin
      what (n-1);
      call B(n)
    end
end.
```

gate1999 | algorithms | time-complexity | normal

Answer

1.18.9 Time Complexity: GATE2000-1.15 [top](#)

<https://gateoverflow.in/638>

Let S be a sorted array of n integers. Let $T(n)$ denote the time taken for the most efficient algorithm to determine if there are two elements with sum less than 1000 in S . Which of the following statement is true?

- A. $T(n) = O(1)$
- B. $n \leq T(n) \leq n \log_2 n$
- C. $n \log_2 n \leq T(n) < \frac{n}{2}$
- D. $T(n) = \left(\frac{n}{2}\right)$

gate2000 | easy | algorithms | time-complexity

Answer

1.18.10 Time Complexity: GATE2003-66 [top](#)

<https://gateoverflow.in/258>

The cube root of a natural number n is defined as the largest natural number m such that $(m^3 \leq n)$. The complexity of computing the cube root of n (n is represented by binary notation) is

- A. $O(n)$ but not $O(n^{0.5})$
- B. $O(n^{0.5})$ but not $O((\log n)^k)$ for any constant $k > 0$
- C. $O((\log n)^k)$ for some constant $k > 0$, but not $O((\log \log n)^m)$ for any constant $m > 0$
- D. $O((\log \log n)^k)$ for some constant $k > 0.5$, but not $O((\log \log n)^{0.5})$

gate2003 | algorithms | time-complexity | normal

Answer

1.18.11 Time Complexity: GATE2004-39 [top](#)

<https://gateoverflow.in/1036>

Two matrices M_1 and M_2 are to be stored in arrays A and B respectively. Each array can be stored either in row-major or column-major order in contiguous memory locations. The time complexity of an algorithm to compute $M_1 \times M_2$ will be

- A. best if A is in row-major, and B is in column-major order
- B. best if both are in row-major order
- C. best if both are in column-major order
- D. independent of the storage scheme

gate2004 algorithms time-complexity easy

Answer

1.18.12 Time Complexity: GATE2004-82 [top](#)

<https://gateoverflow.in/1076>

Let $A[1, \dots, n]$ be an array storing a bit (1 or 0) at each location, and $f(m)$ is a function whose time complexity is $\Theta(m)$. Consider the following program fragment written in a C like language:

```
counter = 0;
for (i=1; i<=n; i++)
{ if a[i] == 1) counter++;
else {f (counter); counter = 0;}
}
```

The complexity of this program fragment is

- A. $\Omega(n^2)$
- B. $\Omega(n \log n)$ and $O(n^2)$
- C. $\Theta(n)$
- D. $o(n)$

gate2004 algorithms time-complexity normal

Answer

1.18.13 Time Complexity: GATE2006-15 [top](#)

<https://gateoverflow.in/976>

Consider the following C-program fragment in which i , j and n are integer variables.

```
for ( i = n, j = 0; i > 0; i /= 2, j +=i );
```

Let $val(j)$ denote the value stored in the variable j after termination of the for loop. Which one of the following is true?

- A. $val(j) = \Theta(\log n)$
- B. $val(j) = \Theta(\sqrt{n})$
- C. $val(j) = \Theta(n)$
- D. $val(j) = \Theta(n \log n)$

gate2006 algorithms normal time-complexity

Answer

1.18.14 Time Complexity: GATE2007-15,ISRO2016-26 [top](#)

<https://gateoverflow.in/56129>

Consider the following segment of C-code:

```
int j, n;
j = 1;
while (j <= n)
    j = j * 2;
```

The number of comparisons made in the execution of the loop for any $n > 0$ is:

- A. $\lceil \log_2 n \rceil + 1$

- B. n
- C. $\lceil \log_2 n \rceil$
- D. $\lfloor \log_2 n \rfloor + 1$

gate2007 algorithms time-complexity normal isro2016

[Answer](#)

1.18.15 Time Complexity: GATE2007-44 [top](#)

<https://gateoverflow.in/1242>

In the following C function, let $n \geq m$.

```
int gcd(n,m) {
    if (n%m == 0) return m;
    n = n%m;
    return gcd(m,n);
}
```

How many recursive calls are made by this function?

- A. $\Theta(\log_2 n)$
- B. $\Omega(n)$
- C. $\Theta(\log_2 \log_2 n)$
- D. $\Theta(\sqrt{n})$

gate2007 algorithms time-complexity normal

[Answer](#)

1.18.16 Time Complexity: GATE2007-45 [top](#)

<https://gateoverflow.in/1243>

What is the time complexity of the following recursive function?

```
int DoSomething (int n) {
    if (n <= 2)
        return 1;
    else
        return (DoSomething (floor (sqrt(n))) + n);
}
```

- A. $\Theta(n^2)$
- B. $\Theta(n \log_2 n)$
- C. $\Theta(\log_2 n)$
- D. $\Theta(\log_2 \log_2 n)$

gate2007 algorithms time-complexity normal

[Answer](#)

1.18.17 Time Complexity: GATE2007-50 [top](#)

<https://gateoverflow.in/1248>

An array of n numbers is given, where n is an even number. The maximum as well as the minimum

of these n numbers needs to be determined. Which of the following is **TRUE** about the number of comparisons needed?

- A. At least $2n - c$ comparisons, for some constant c are needed.
- B. At most $1.5n - 2$ comparisons are needed.
- C. At least $n \log_2 n$ comparisons are needed
- D. None of the above

[gate2007](#) [algorithms](#) [time-complexity](#) [easy](#)

Answer

1.18.18 Time Complexity: GATE2007-51 [top](#)

<https://gateoverflow.in/1249>

Consider the following C program segment:

```
int IsPrime (n)
{
    int i, n;
    for (i=2; i<=sqrt(n); i++)
        if (n%i == 0)
            {printf("Not Prime \n"); return 0;}
    return 1;
}
```

Let $T(n)$ denote number of times the *for* loop is executed by the program on input n . Which of the following is TRUE?

- A. $T(n) = O(\sqrt{n})$ and $T(n) = \Omega(\sqrt{n})$
- B. $T(n) = O(\sqrt{n})$ and $T(n) = \Omega(1)$
- C. $T(n) = O(n)$ and $T(n) = \Omega(\sqrt{n})$
- D. None of the above

[gate2007](#) [algorithms](#) [time-complexity](#) [normal](#)

Answer

1.18.19 Time Complexity: GATE2007-IT-17 [top](#)

<https://gateoverflow.in/3450>

Exponentiation is a heavily used operation in public key cryptography. Which of the following options is the tightest upper bound on the number of multiplications required to compute $b^n \bmod m, 0 \leq b, n \leq m$?

- A. $O(\log n)$
- B. $O(\sqrt{n})$
- C. $O\left(\frac{n}{\log n}\right)$
- D. $O(n)$

[gate2007-it](#) [algorithms](#) [time-complexity](#) [normal](#)

Answer

1.18.20 Time Complexity: GATE2007-IT-81 [top](#)

<https://gateoverflow.in/3533>

Let P_1, P_2, \dots, P_n be n points in the xy -plane such that no three of them are collinear. For every pair of points P_i and P_j , let L_{ij} be the line passing through them. Let L_{ab} be the line with the steepest gradient among all $n(n - 1)/2$ lines.

The time complexity of the best algorithm for finding P_a and P_b is

- A. $\Theta(n)$
- B. $\Theta(n \log n)$
- C. $\Theta(n \log^2 n)$
- D. $\Theta(n^2)$

[gate2007-it](#) [algorithms](#) [time-complexity](#) [normal](#)

[Answer](#)

1.18.21 Time Complexity: GATE2008-40 [top](#)

<https://gateoverflow.in/452>

The minimum number of comparisons required to determine if an integer appears more than $\frac{n}{2}$ times in a sorted array of n integers is

- A. $\Theta(n)$
- B. $\Theta(\log n)$
- C. $\Theta(\log^* n)$
- D. $\Theta(1)$

[gate2008](#) [normal](#) [algorithms](#) [time-complexity](#)

[Answer](#)

1.18.22 Time Complexity: GATE2008-47 [top](#)

<https://gateoverflow.in/459>

We have a binary heap on n elements and wish to insert n more elements (not necessarily one after another) into this heap. The total time required for this is

- A. $\Theta(\log n)$
- B. $\Theta(n)$
- C. $\Theta(n \log n)$
- D. $\Theta(n^2)$

[gate2008](#) [algorithms](#) [time-complexity](#) [normal](#)

[Answer](#)

1.18.23 Time Complexity: GATE2008-74 [top](#)

<https://gateoverflow.in/495>

Consider the following C functions:

```
int f1 (int n)
{
```

```

if(n == 0 || n == 1)
    return n;
else
    return (2 * f1(n-1) + 3 * f1(n-2));
}
int f2(int n)
{
    int i;
    int X[N], Y[N], Z[N];
    X[0] = Y[0] = Z[0] = 0;
    X[1] = 1; Y[1] = 2; Z[1] = 3;
    for(i = 2; i <= n; i++){
        X[i] = Y[i-1] + Z[i-2];
        Y[i] = 2 * X[i];
        Z[i] = 3 * X[i];
    }
    return X[n];
}

```

The running time of $f1(n)$ and $f2(n)$ are

- A. $\Theta(n)$ and $\Theta(n)$
- B. $\Theta(2^n)$ and $\Theta(n)$
- C. $\Theta(n)$ and $\Theta(2^n)$
- D. $\Theta(2^n)$ and $\Theta(2^n)$

gate2008 algorithms time-complexity normal

Answer

1.18.24 Time Complexity: GATE2008-75 top

<https://gateoverflow.in/43489>

Consider the following C functions:

```

int f1 (int n)
{
    if(n == 0 || n == 1)
        return n;
    else
        return (2 * f1(n-1) + 3 * f1(n-2));
}
int f2(int n)
{
    int i;
    int X[N], Y[N], Z[N];
    X[0] = Y[0] = Z[0] = 0;
    X[1] = 1; Y[1] = 2; Z[1] = 3;
    for(i = 2; i <= n; i++){
        X[i] = Y[i-1] + Z[i-2];
        Y[i] = 2 * X[i];
        Z[i] = 3 * X[i];
    }
    return X[n];
}

```

$f1(8)$ and $f2(8)$ return the values

- A. 1661 and 1640
- B. 59 and 59
- C. 1640 and 1640
- D. 1640 and 1661

[gate2008](#)
[normal](#)
[algorithms](#)
[time-complexity](#)
Answer**1.18.25 Time Complexity: GATE2010-12** [top](#)<https://gateoverflow.in/2185>

Two alternative packages A and B are available for processing a database having 10^k records. Package A requires $0.0001n^2$ time units and package B requires $10n \log_{10} n$ time units to process n records. What is the smallest value of k for which package B will be preferred over A ?

- A. 12
- B. 10
- C. 6
- D. 5

[gate2010](#)
[algorithms](#)
[time-complexity](#)
[easy](#)
Answer**1.18.26 Time Complexity: GATE2014-1-42** [top](#)<https://gateoverflow.in/1920>

Consider the following pseudo code. What is the total number of multiplications to be performed?

```
D = 2
for i = 1 to n do
    for j = i to n do
        for k = j + 1 to n do
            D = D * 3
```

- A. Half of the product of the 3 consecutive integers.
- B. One-third of the product of the 3 consecutive integers.
- C. One-sixth of the product of the 3 consecutive integers.
- D. None of the above.

[gate2014-1](#)
[algorithms](#)
[time-complexity](#)
[normal](#)
Answer**1.18.27 Time Complexity: GATE2015-1-40** [top](#)<https://gateoverflow.in/8299>

An algorithm performs $(\log N)^{\frac{1}{2}}$ find operations, N insert operations, $(\log N)^{\frac{1}{2}}$ delete operations, and $(\log N)^{\frac{1}{2}}$ decrease-key operations on a set of data items with keys drawn from a linearly ordered set. For a delete operation, a pointer is provided to the record that must be deleted. For the decrease-key operation, a pointer is provided to the record that has its key decreased. Which one of the following data structures is the most suited for the algorithm to use, if the goal is to achieve the best total asymptotic complexity considering all the operations?

- A. Unsorted array
- B. Min - heap
- C. Sorted array
- D. Sorted doubly linked list

[gate2015-1](#)
[algorithms](#)
[data-structure](#)
[normal](#)
[time-complexity](#)
Answer**1.18.28 Time Complexity: GATE2015-2-22** [top](#)<https://gateoverflow.in/8113>

An unordered list contains n distinct elements. The number of comparisons to find an element in this list that is neither maximum nor minimum is

- A. $\Theta(n \log n)$
- B. $\Theta(n)$
- C. $\Theta(\log n)$
- D. $\Theta(1)$

gate2015-2 algorithms time-complexity easy

Answer

1.18.29 Time Complexity: GATE2017-2-03 top

<https://gateoverflow.in/118156>

Match the algorithms with their time complexities:

Algorithms	Time Complexity
P. Tower of Hanoi with n disks	i. $\Theta(n^2)$
Q. Binary Search given n numbers n sorted numbers	ii. $\Theta(n \log n)$
R. Heap sort given n numbers at the worst case	iii. $\Theta(2^n)$
S. Addition of two $n \times n$ matrices	iv. $\Theta(\log n)$

- A. P → (iii) Q → (iv) R → (i) S → (ii)
- B. P → (iv) Q → (iii) R → (i) S → (ii)
- C. P → (iii) Q → (iv) R → (ii) S → (i)
- D. P → (iv) Q → (iii) R → (ii) S → (i)

gate2017-2 algorithms time-complexity

Answer

1.18.30 Time Complexity: GATE2017-2-38 top

<https://gateoverflow.in/118283>

Consider the following C function

```
int fun(int n) {
    int I, j;
    for(i=1; i<=n; i++) {
        for (j=1; j<n; j+=i) {
            printf("%d %d", I, j);
        }
    }
}
```

Time complexity of *fun* in terms of Θ notation is

- A. $\Theta(n\sqrt{n})$
- B. $\Theta(n^2)$
- C. $\Theta(n \log n)$
- D. $\Theta(n^2 \log n)$

gate2017-2 algorithms time-complexity

Answer

1.18.31 Time Complexity: TIFR2013-B-12 top

<https://gateoverflow.in/25774>

It takes $O(n)$ time to find the median in a list of n elements, which are not necessarily in sorted order while it takes only $O(1)$ time to find the median in a list of n sorted elements. How much time does it take to find the median of $2n$ elements which are given as two lists of n sorted elements each?

- A. $O(1)$
- B. $O(\log n)$ but not $O(1)$
- C. $O(\sqrt{n})$ but not $O(\log n)$
- D. $O(n)$ but not $O(\sqrt{n})$
- E. $O(n \log n)$ but not $O(n)$

[tifr2013](#) [algorithms](#) [time-complexity](#)

Answer

1.18.32 Time Complexity: TIFR2013-B-18 top

<https://gateoverflow.in/25865>

Let S be a set of numbers. For $x \in S$, the rank of x is the number of elements in S that are less than or equal to x . The procedure $\text{Select}(S, r)$ takes a set S of numbers and a rank r ($1 \leq r \leq |S|$) and returns the element in S of rank r . The procedure $\text{MultiSelect}(S, R)$ takes a set of numbers S and a list of ranks $R = \{r_1 < r_2 < \dots < r_k\}$, and returns the list $\{x_1 < x_2 < \dots < x_k\}$ of elements of S , such that the rank of x_i is r_i . Suppose there is an implementation for $\text{Select}(S, r)$ that uses at most “constant $\cdot |S|$ ” binary comparisons between elements of S . The minimum number of comparisons needed to implement $\text{MultiSelect}(S, R)$ is

- a. constant $\cdot |S| \log |S|$
- b. constant $\cdot |S|$
- c. constant $\cdot |S| |R|$
- d. constant $\cdot |R| \log |S|$
- e. constant $\cdot |S| (1 + \log |R|)$

[tifr2013](#) [algorithms](#) [time-complexity](#)

Answer

1.18.33 Time Complexity: TIFR2014-B-7 top

<https://gateoverflow.in/27189>

Which of the following statements is TRUE for all sufficiently large n ?

- A. $(\log n)^{\log \log n} < 2^{\sqrt{\log n}} < n^{1/4}$
- B. $2^{\sqrt{\log n}} < n^{1/4} < (\log n)^{\log \log n}$
- C. $n^{1/4} < (\log n)^{\log \log n} < 2^{\sqrt{\log n}}$
- D. $(\log n)^{\log \log n} < n^{1/4} < 2^{\sqrt{\log n}}$
- E. $2^{\sqrt{\log n}} < (\log n)^{\log \log n} < n^{1/4}$

[tifr2014](#) [algorithms](#) [time-complexity](#)

Answer

1.18.34 Time Complexity: TIFR2015-B-3 [top](#)

<https://gateoverflow.in/29846>

Consider the following code fragment in the *C* programming language when run on a non-negative integer n .

```
int f (int n)
{
    if (n==0 || n==1)
        return 1;
    else
        return f (n - 1) + f(n - 2);
}
```

Assuming a typical implementation of the language, what is the running time of this algorithm and how does it compare to the optimal running time for this problem?

- A. This algorithm runs in polynomial time in n but the optimal running time is exponential in n .
- B. This algorithm runs in exponential time in n and the optimal running time is exponential in n .
- C. This algorithm runs in exponential time in n but the optimal running time is polynomial in n .
- D. This algorithm runs in polynomial time in n and the optimal running time is polynomial in n .
- E. The algorithm does not terminate.

[tifr2015](#) [time-complexity](#)

[Answer](#)

Answers: Time Complexity

1.18.1 Time Complexity: CMI2013-A-10 [top](#)

<https://gateoverflow.in/46602>



Selected Answer

Answer: $5050 (100 + 99 + 98 + \dots + 1) = (100 * 101)/2$

7 votes

-- Dhananjay Kumar Sharma (25.7k points)

1.18.2 Time Complexity: CMI2015-A-08 [top](#)

<https://gateoverflow.in/47045>



Selected Answer

Correct answer is (C)

42

It will start comparison from $(85, 5), (84, 6), (83, 7), \dots, (45, 45), (44, 46)$

Hence, total number of comparison will be $(85 - 44) + 1 = 41 + 1 = 42$

16 votes

-- Muktinath Vishwakarma (35.4k points)

1.18.3 Time Complexity: CMI2017-B-8 [top](#)

<https://gateoverflow.in/203320>



Selected Answer

a. Function computes k th smallest element (if $k <$ number of elements in the array, else it returns max element).

- b. Time complexity = $O(k*n)$, where n is the number of elements in the array.
 c. Worst case input: Array consists of a distinct increasing sequence of numbers and from this sequence, if we need to find nth smallest (i.e. max element) then it will take $O(n*n)$ time.

1 votes

-- Digvijay (54.9k points)

1.18.4 Time Complexity: GATE1989-2-iii top

<https://gateoverflow.in/87080>

	Selected Answer
(A) $O(\log n)$	(r) Binary Search
(B) $O(n)$	(s) Selection of the k^{th} smallest element in a set of n elements.(Worst case)
(C) $O(n \log n)$	(p) Heap sort
(D) $O(n^2)$	(q) Depth-first search (It will be $O(n + m)$ if the graph is given in the form of adjacency list but if the graph is in the form of adjacency matrix then the complexity is $O(n \times n)$) have to traverse through the whole row until we find an edge.)

PS: k^{th} smallest element can be found in $O(n)$ time using partition algorithm.

$$T(n) = T(n/2) + n \text{ (For finding } k^{th} \text{ smallest element)}$$

By Masters Theorem it will be done in $O(n)$.

11 votes

-- Prajwal Bhat (11.3k points)

1.18.5 Time Complexity: GATE1993-8.7 top

<https://gateoverflow.in/2305>

Selected Answer

This is N added itself N times. So it is N^2 . Even if you consider as sum of $O(1) + O(2) + \dots + O(N)$, it will add up to N^2

So answer is

A) $O(N)$ this is false.

B,C,D,E) All of this are true. We have N^2 here, so all options apart from A are correct.

In fact B = D = E these three options are same. and N^3 is always upper bound of N^2 . So $O(N^3)$ is

also true.

PS: $\sum(K = 1 \text{ to } n) O(K)$ is never equal to n functions. It is always equal to one single function.

It can be written as $c.1 + c.2 + c.3$ and so on which results in $O(N^2)$ (source Cormen)

24 votes

-- Akash Kanase (42.5k points)

1.18.6 Time Complexity: GATE1999-1.13 top

<https://gateoverflow.in/1466>



Selected Answer

Answer is (D) None of these.

We just require $n/2$ swaps in the worst case. The algorithm is as given below:

Find positive number from left side and negative number from right side and do exchange. Since, at least one of them must be less than or equal to $n/2$, there cannot be more than $n/2$ exchanges. An implementation is given below:

http://gatcse.in/wiki/Moving_Negative_Numbers_to_the_Beginning_of_Array

36 votes

-- Arjun Suresh (350k points)

1.18.7 Time Complexity: GATE1999-1.16 top

<https://gateoverflow.in/1469>



Selected Answer

a. $\log n$

$$a^n = (a^2)^{\frac{n}{2}}.$$

One multiplication and recurrence on $\frac{n}{2}$. So, we get the recurrence relation for the number of multiplications as

$$T(n) = T(n/2) + 1.$$

This gives $T(n) = \log_2 n$

For $n = 8$, we can do

$$b = a \times a$$

$$b = b \times b$$

$$b = b \times b \text{ and we get } b = a^8$$

24 votes

-- Arjun Suresh (350k points)

1.18.8 Time Complexity: GATE1999-11a top

<https://gateoverflow.in/1510>



The recurrence relation for time complexity is

$T(n) = T(n - 1) + \frac{1}{n} + c$ ($O(1/n)$ replaced with $1/n$ and so our answer will also be in O only and not Θ)

$$\begin{aligned} T(n) &= T(n - 2) + \frac{1}{n-1} + \frac{1}{n} + 2c \\ &= T(1) + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} + (n - 1)c \\ &= A(1) + \frac{1}{2} + \frac{1}{3} \dots + \frac{1}{n} + nc \\ &= 1 + \frac{1}{2} + \frac{1}{3} \dots + \frac{1}{n} + nc \\ &= \log n + nc \end{aligned}$$

(Sum of the first n terms in harmonic series is $\Theta(\log n)$)

So, our time complexity will be $O(n)$.

Upvote 24 votes

-- Arjun Suresh (350k points)

1.18.9 Time Complexity: GATE2000-1.15 top

<https://gateoverflow.in/638>



Answer: Option A. Because array is always sorted just check the 1st two elements.

Upvote 29 votes

-- anshu (3.3k points)

1.18.10 Time Complexity: GATE2003-66 top

<https://gateoverflow.in/258>



We can simply do a binary search in the array of natural numbers from $1..n$ and check if the cube of the number matches n (i.e., check if $a[i] * a[i] * a[i] == n$). This check takes $O(\log n)$ time and in the worst case we need to do the search $O(\log n)$ times. So, in this way we can find the cube root in $O(\log^2 n)$. So, options (A) and (B) are wrong.

Now, a number is represented in binary using $\log n$ bit. Since each bit is important in finding the cube root, any cube root finding algorithm must examine each bit at least once. This ensures that complexity of cube root finding algorithm cannot be lower than $\log n$. (It must be $\Omega(\log n)$). So, (D) is also false and (C) is the correct answer.

Upvote 52 votes

-- gatecse (18k points)

1.18.11 Time Complexity: GATE2004-39 top

<https://gateoverflow.in/1036>



D is correct

Here time complexity is asked, for each access of array element it will be constant,

So the time complexity will not depend upon storage. If at all program execution time is asked

a is true

34 votes

-- Anurag Semwal (8k points)

1.18.12 Time Complexity: GATE2004-82 top

<https://gateoverflow.in/1076>



Selected Answer

The key part in the code is "*counter* = 0" in the else part as we can see below.

Lets take the best case. This happens when $a[i] = 1$ for all i , and then the loop executes with time complexity $\Theta(1)$ for each iteration and hence overall time complexity of $\Theta(n)$ and we can say time complexity of the code fragment is $\Omega(n)$ and hence options A and B are false.

Now, consider the worst case. This happens when $a[i] = 0$ or when else part is executed. Here, the time complexity of each iteration will be $\Theta(counter)$ and after each else, counter is reset to 0. Let k iterations go to the else part during the worst case. Then the worst case time complexity will be $\Theta(x_1) + \Theta(x_2) + \dots + \Theta(x_k) + \Theta(n - k)$, where x_i is the value of the counter when, $A[i] = 0$ and $f(counter)$ is called. But due to $counter = 0$ after each call to $f()$, we have, $x_1 + x_2 + \dots + x_k = n$. So,
 $\Theta(x_1) + \Theta(x_2) + \dots + \Theta(x_k) + \Theta(n - k) = \Theta(n) + \Theta(n - k) = \Theta(n)$.

Since the time complexity is $\Omega(n)$ and $\Theta(n)$ we can say it is $\Theta(n)$ - Option (C). (Option D is false because the small o needs the growth rate to be **STRICTLY lower** and not equal to or lower as the case for big O)

If $counter = 0$ was not there in else part, then time complexity would be $\Omega(n)$ and $O(n^2)$ as in worst case we can have equal number of 0's and 1's in array a giving time complexity $\Theta(1) + \Theta(2) + \dots + \Theta(n/2) + \Theta(n/2)$ would give $O(n^2)$.

30 votes

-- Arjun Suresh (350k points)

1.18.13 Time Complexity: GATE2006-15 top

<https://gateoverflow.in/976>



Selected Answer

Answer will be $\Theta(n)$

$$j = n/2 + n/4 + n/8 + \dots + 1$$

$$= n [1/2^1 + 1/2^2 + 1/2^3 + \dots + 1/2^{\lg n}]$$

(Sum of first n terms of GP is $[a \cdot \frac{1-r^n}{1-r}]$, where a is the first term, r is the common ratio < 1 , and n is the number of terms)

$$= n \left[1/2 \frac{1-(1/2)^{\lg n}}{1-1/2} \right]$$

$$= n \left[\frac{n-1}{n} \right]$$

$$= n - 1 = \Theta(n)$$

26 votes

-- rahulkrr (805 points)

1.18.14 Time Complexity: GATE2007-15,ISRO2016-26 [top](#)

n	no. of comparisons	$\text{Ceil}(\log_2 n) + 1$
1	$2(j = 1, 2)$	1
2	$3(j = 1, 2, 4)$	2
3	$3(j = 1, 2, 4)$	2
4	$4(j = 1, 2, 4, 8)$	3
5	$4(j = 1, 2, 4, 8)$	4

may be we have to count those comparisons which results in the execution of loop.

Answer should be $\text{Ceil}(\log_2 n) + 1$

EDIT: but answer could be: $\text{floor}(\log_2 n) + 2$

👍 40 votes

-- Vikrant Singh (13.5k points)

1.18.15 Time Complexity: GATE2007-44 [top](#)

<https://gateoverflow.in/1242>



Selected Answer

Worst case will arise when both n and m are consecutive Fibonacci numbers.

$$\gcd(F_n, F_{n-1}) = \gcd(F_{n-1}, F_{n-2}) = \dots = \gcd(F_1, F_0) = 1$$

and n^{th} Fibonacci number is 1.618^n , where 1.618 is the [Golden ratio](#).

So, to find $\gcd(n, m)$, number of recursive calls will be $\Theta(\log n)$.

👍 32 votes

-- Vikrant Singh (13.5k points)

1.18.16 Time Complexity: GATE2007-45 [top](#)

<https://gateoverflow.in/1243>



Selected Answer

We are asked the time complexity which will be the number of recursive calls in the function as in each call we perform a constant no. of operations and a recursive call. The recurrence relation for this is (considering constant time "c" as 1)

$$T(n) = T(\sqrt{n}) + 1$$

$$\begin{aligned} &= T(n^{1/4}) + 2 \\ &= T(n^{1/8}) + 3 \end{aligned}$$

Going like this we will eventually reach $T(3)$ or $T(2)$. For asymptotic case this doesn't matter and

we can assume we reach $T(2)$ and in next step reach $T(1)$. So, all we want to know is how many steps it takes to reach $T(1)$ which will be 1+no. of steps to reach $T(2)$.

From the recurrence relation we know that $T(2)$ happens when $n^{\left(\frac{1}{2^k}\right)} = 2$.

Taking log and equating,

$$\begin{aligned} \frac{1}{2^k} \log n &= 1 \\ \implies 2^k &= \log n \\ \implies k &= \log \log n. \end{aligned}$$

So, $T(1)$ happens in $\log \log n + 1$ calls, but for asymptotic complexity we can write as $\Theta(\log \log n)$

Alternatively,

Substituting values

$$\begin{aligned} T(1) &= 1 \\ T(2) &= 1 \\ T(3) &= T(1) + 1 = 2 \\ &\vdots \\ T(8) &= T(2) + 1 = 2 \\ T(9) &= T(3) + 1 = 3 \\ &\vdots \end{aligned}$$

$$\begin{aligned} T\left(\left((2^2)^2\right)^2\right) &= T\left((2^2)^2\right) + 1 \\ &= T(2^2) + 2 \\ &= T(2) + 3 = 1 + 3 = 4, \end{aligned}$$

$\log \log n = 3$ as $n = 256$.

$$\begin{aligned} T\left(\left(\left((2^2)^2\right)^2\right)^2\right) &= 6, \\ \log \log n &= 5 \text{ as } n = 65536 \times 65536 = 2^{32} \end{aligned}$$

$$\begin{aligned} T(2^{(2^{10})}) &= T(2^{512}) + 1 \\ &= T(2^{256}) + 2 \\ &= T(2^{128}) + 3 \\ &= T(2^{64}) + 4 \\ &= T(2^{32}) + 5 \\ &= T(2^{16}) + 6 \\ &= T(2^8) + 7 \\ &= T(2^4) + 8 \\ &= T(2^2) + 9 \\ &= T(2) + 10 = 11, \\ \log \log n &= 10 \end{aligned}$$

So, answer is **D**.

<http://stackoverflow.com/questions/16472012/what-would-cause-an-algorithm-to-have-o-log-log-n-complexity>

36 votes

-- Arjun Suresh (350k points)

1.18.17 Time Complexity: GATE2007-50 top

<https://gateoverflow.in/1248>



Selected Answer

An easier way to find it is by using **Tournament Method Technique** -

1. To find the smallest element in the array will take **$n - 1$ comparisions**.
2. To find the largest element -
 - a. After the first round of Tournament , there will be exactly **$n/2$ numbers** that will loose the round.
 - b. So the biggest looser (the largest number) should be among these 50 losers.To find the largest number will take **$n/2 - 1$** .

$$\text{Total Comparisons} = (n - 1) + (n/2 - 1) = 1.5n - 2.$$

41 votes

-- Harsh181996 (4k points)

1.18.18 Time Complexity: GATE2007-51 top

<https://gateoverflow.in/1249>



Selected Answer

Answer is **option B**.

Worst Case : $T(n) = \mathcal{O}(\sqrt{n})$

Best Case : When n is an even number body of *for* loop is executed only 1 time (due to "return 0" inside if) which is irrespective of n . $\therefore T(n) = \Omega(1)$

24 votes

-- Gate Keeda (19.6k points)

1.18.19 Time Complexity: GATE2007-IT-17 top

<https://gateoverflow.in/3450>



Selected Answer

Answer is (A)

We need to divide

n recursively and compute like following:

$C_1 = b^{\frac{n}{2}} \times b^{\frac{n}{2}}$. In this, we need to calculate $b^{\frac{n}{2}}$ only once.

$$C_2 = b^{\frac{n}{4}} \times b^{\frac{n}{4}}$$

:

$$C_k = b^2 \times b^2 \quad \left\{ \begin{array}{l} k = \log n \end{array} \right.$$

Recurrence relation: $T(n) = T\left(\frac{n}{2}\right) + O(1)$

$$T(n) = O(\log n)$$

Upvote 34 votes

-- Sandeep_Uniyal (7.5k points)

1.18.20 Time Complexity: GATE2007-IT-81 top

<https://gateoverflow.in/3533>



Selected Answer

Answer: **B**

$$\text{Gradient} = y_2 - y_1 / x_2 - x_1$$

For gradient to be maximum $x_2 - x_1$ should be minimum. So, sort the points (in $\Theta(n \log n)$ time) according to x coordinate and find the minimum difference between them (in $\Theta(n)$ time).

Best complexity: $\Theta(n \log n + n)$ which leads to B.

<https://www.careercup.com/question?id=4787065684230144>

<https://stackoverflow.com/questions/8222108/max-slope-from-set-of-points>

Upvote 17 votes

-- Rajarshi Sarkar (34.1k points)

1.18.21 Time Complexity: GATE2008-40 top

<https://gateoverflow.in/452>



Selected Answer

Answer is **option B.**

whenever there exists an element which is present in the array : more than $\frac{n}{2}$ times, then definitely it will be present at the middle index position; in addition to that it will also be present at anyone of the neighbourhood indices namely $i - 1$ and $i + 1$

No matter how we push that stream of **More than** $\frac{n}{2}$ times of elements of same value around the Sorted Array, it is bound to be present at the middle index + atleast anyone of its neighbourhood

once we got the element which should have occurred more than $n/2$ times we [count its total occurrences](#) in $\mathcal{O}(\log n)$ time.

Upvote 35 votes

-- Amar Vashishth (30.5k points)

1.18.22 Time Complexity: GATE2008-47 top

<https://gateoverflow.in/459>



Selected Answer

An insert operation on a binary heap takes $\mathcal{O}(\log n)$ time, but an alternative approach we can use. which requires us to insert n elements in heap without any computation i.e. in constant time. after which we can apply Heapify operation(this operation creates heap in linear time) on the array of those element and Hence obtain a Heap in $\mathcal{O}(n)$ time.

Here "not necessarily one after another" should mean that we can insert n elements at once and

not necessarily have to wait for first insert to be completed before doing second.

31 votes

-- Amar Vashishth (30.5k points)

1.18.23 Time Complexity: GATE2008-74 top

<https://gateoverflow.in/495>



Q.74 = option **B**
Q.75 = option **C**

Time complexity of f_1 is given by

$T(n) = T(n-1) + T(n-2)$, (multiplication by 2 and 3 won't affect complexity as it is a constant time operation)

$$T(0) = T(1) = 1$$

The solution to this (fibonacci series) is given by Golden ratio. https://en.wikipedia.org/wiki/Golden_ratio which is $O(2^n)$. (Using theta in question must be a mistake)

Time complexity of f_2 is $\Theta(n)$ as here all recursive calls are avoided by saving the results in an array (dynamic programming).

So, answer to 74 is (**B**).

75. Both f_1 and f_2 are calculating the same function. So,

$$\begin{aligned}f_1(2) &= 2f_1(1) + 3f_1(0) = 2 \\f_1(3) &= 2f_1(2) + 3f_1(1) = 7 \\f_1(4) &= 20 \\f_1(5) &= 61 \\f_1(6) &= 182 \\f_1(7) &= 547 \\f_1(8) &= 1640 = f_2(8)\end{aligned}$$

31 votes

-- Arjun Suresh (350k points)

1.18.24 Time Complexity: GATE2008-75 top

<https://gateoverflow.in/43489>



Both f_1 and f_2 are calculating the same function in recursive and iterative fashion respectively.

So, lets solve the recurrence relation.

$$F(n) = 2F(n-1) + 3F(n-2)$$

Its characteristic equation will be

$$\begin{aligned}r^2 - 2r - 3 &= 0 \\ \Rightarrow (r-3)(r+1) &= 0 \\ \Rightarrow r &= 3, -1\end{aligned}$$

Now, we have two roots. So the equation will be

$$a_n = C_1(-1^n) + C_2(3^n) \rightarrow (1)$$

Now from the function $f1$,

$$F(0) = 0 \text{ and } F(1) = 1$$

$$\begin{aligned} \text{So, } C_1 + C_2 &= 0 \\ -C_1 + 3C_2 &= 1 \end{aligned}$$

$$\Rightarrow C_1 = -1/4 \text{ and } C_2 = 1/4$$

After putting the values in (i) the equation will become

$$a_n = (-1/4)(-1^n) + (1/4)(3^n)$$

Putting $n = 8$ it will become

$$F(8) = a_8 = 1640.$$

Or we can do it manually

$$\begin{aligned} f1(2) &= 2f1(1) + 3f1(0) = 2 \\ f1(3) &= 2f1(2) + 3f1(1) = 7 \\ f1(4) &= 20 \\ f1(5) &= 61 \\ f1(6) &= 182 \\ f1(7) &= 547 \\ f1(8) &= 1640 = f2(8) \end{aligned}$$

1 votes

-- Puja Mishra (10.6k points)

1.18.25 Time Complexity: GATE2010-12 top

<https://gateoverflow.in/2185>



Selected Answer

$$10n \log_{10} n \leq 0.0001n^2$$

$$\Rightarrow 10 \times 10^k \log_{10} 10^k \leq 0.0001(10^k)^2$$

$$\Rightarrow 10^{k+1} k \leq 0.0001 \times 10^{2k}$$

$$\Rightarrow k \leq 10^{2k-k-1-4}$$

$$\Rightarrow k \leq 10^{k-5}$$

Trying the values, 5 doesn't satisfy this but 6 satisfies.

47 votes

-- Arjun Suresh (350k points)

1.18.26 Time Complexity: GATE2014-1-42 top

<https://gateoverflow.in/1920>



Selected Answer

Total number of multiplications

$$\begin{aligned}
&= \sum_{i=1}^n \sum_{j=i}^n \sum_{k=j+1}^n 1 \\
&= \sum_{i=1}^n \sum_{j=i}^n (n - j) \\
&= \sum_{i=1}^n (n - i) + (n - (i + 1)) + \dots + (n - n) \\
&= \frac{1}{2} \sum_{i=1}^n (n - i)(n - i + 1) \\
&= \frac{1}{2} \sum_{i=1}^n (n^2 + n + i^2 - (2n + 1)i) \\
&= \frac{1}{2} (n^3 + n^2 + \sum_{i=1}^n i^2 - (2n + 1) \cdot \sum_{i=1}^n i) \\
&= \frac{1}{2} \left(n^3 + n^2 + \frac{n(n+1)(2n+1)}{6} - (2n + 1) \cdot \frac{n(n+1)}{2} \right) \\
&= \frac{1}{2} \left(n^3 + n^2 - \frac{n(n+1)(2n+1)}{3} \right) \\
&= \frac{n}{6} (3n(n+1) - (n+1) \cdot (2n+1)) \\
&= \frac{n(n+1)}{6} (n-1) \\
&= \frac{(n-1)n(n+1)}{6}
\end{aligned}$$

Therefore, correct answer would be (C).

54 votes

-- suraj (5.6k points)

1.18.27 Time Complexity: GATE2015-1-40 top

<https://gateoverflow.in/8299>

	(log find N) ^{1/2}	N insert	(log delete N) ^{1/2}	(log decrease-key N) ^{1/2}
Unsorted Array	$O(N(\log N)^{1/2})$	$O(N)$	$O(\log N)^{1/2}$	$O(\log N)^{1/2})$
Min-heap	$O(N(\log N)^{1/2})$	$O(N \log N)$	$O(\log N)^{3/2}$	$O((\log N)^{3/2})$
Sorted Array	$O(\log N)^{3/2}$	$O(N^2)$	$O(N(\log N)^{1/2})$	$O(N(\log N)^{1/2})$

Sorted doubly linked-list	$O(N \log N)$	$O(N^2)$	$O((\log N)^{\frac{1}{2}})$	$O(N(\log N)^{\frac{1}{2}})$
----------------------------------	---------------	----------	-----------------------------	------------------------------

So, Unsorted array is the answer.

The operations given can be performed in any order. So, for Min-heap we cannot do the usual BuildHeap method.

Delete in unsorted array is $O(1)$ as we can just swap the deleted element with the last element in the array and delete the last element.

For sorted-doubly linked-list we cannot do binary search as this would require another array to maintain the pointers to the nodes.

61 votes

-- Arjun Suresh (350k points)

1.18.28 Time Complexity: GATE2015-2-22 top

<https://gateoverflow.in/8113>



Selected Answer

Ans: $\Theta(1)$, because all elements are distinct, select any three numbers and output 2nd largest from them.

60 votes

-- Vikrant Singh (13.5k points)

1.18.29 Time Complexity: GATE2017-2-03 top

<https://gateoverflow.in/118156>



Selected Answer

According to the recurrence relation

$$T(n) = 2T(n - 1) + 1$$

Tower of hanoi we get it is $\Theta(2^n)$

Now, heap sort worst case $\Theta(n \log n)$

Binary Search given n numbers n sorted numbers $\Theta(\log n)$

Addition of two nxn matrices $\Theta(n^2)$

So, C is correct answer here.

22 votes

-- Aboveallplayer (18.4k points)

1.18.30 Time Complexity: GATE2017-2-38 top

<https://gateoverflow.in/118283>



Selected Answer

Inner for loop is dependent on i , so for each i we have to check no of times inner loop operating..

It'll be something like

$$\frac{n-1}{1} + \frac{n-1}{2} + \frac{n-1}{3} + \dots + \frac{n-1}{n-1} + 1$$

$$\frac{n}{1} + \frac{n}{2} + \frac{n}{3} + \dots + \frac{n}{n-1} - \log(n-1)$$

$$n\left\{\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-1}\right\} - \log(n-1)$$

$$n \log(n-1) - \log(n-1)$$

$$n \log(n-1)$$

$$n \log n$$

14 votes

-- 2018 (6.7k points)

1.18.31 Time Complexity: TIFR2013-B-12 top

<https://gateoverflow.in/25774>



Selected Answer

1. Calculate the medians $m1$ and $m2$ of the input arrays $ar1[]$ and $ar2[]$ respectively.
2. If $m1$ and $m2$ both are equal.
return $m1$ (or $m2$)
3. If $m1$ is greater than $m2$, then median is present in one of the below two subarrays.
 - a) From first element of $ar1$ to $m1$ ($ar1[0 \text{ to } n/2]$)
 - b) From $m2$ to last element of $ar2$ ($ar2[n/2 \text{ to } n-1]$)
4. If $m2$ is greater than $m1$, then median is present in one of the below two subarrays.
 - a) From $m1$ to last element of $ar1$ ($ar1[n/2 \text{ to } n-1]$)
 - b) From first element of $ar2$ to $m2$ ($ar2[0 \text{ to } n/2]$)
5. Repeat the above process until size of both the subarrays becomes 2.
6. If size of the two arrays is 2 then
the median.
 $\text{Median} = (\max(ar1[0], ar2[0]) + \min(ar1[1], ar2[1]))/2$
Time complexity $O(\log n)$

<http://www.geeksforgeeks.org/median-of-two-sorted-arrays/>

16 votes

-- Umang Raman (15.7k points)

1.18.32 Time Complexity: TIFR2013-B-18 top

<https://gateoverflow.in/25865>

- a) should be the answer

Reason: Sort the set. Initialize a counter to 1. Keep a parallel pointer to the sorted list of needed ranks. Keep incrementing the counter as you traverse the sorted set. As soon as counter matches the first needed rank, add this to the required output. Increment the parallel pointer.

2 votes

-- Akshay Arora (2.5k points)

1.18.33 Time Complexity: TIFR2014-B-7 top

<https://gateoverflow.in/27189>



Let us take \log for each function.

1. $\log(\log n)^{\log \log n} = (\log \log n)^2$
2. $\log 2^{\sqrt{\log n}} = \sqrt{\log n} = (\log n)^{0.5}$
3. $\log n^{1/4} = 1/4 \log n$

Here, If we consider $\log n$ as a term (which is common in all 3), first 1 is a log function, second one is sqrt function and third one is linear function of $\log n$. Order of growth of these functions are [well known](#) and \log is the slowest growing followed by sqrt and then linear. So, option A is the correct answer here.

PS: After taking \log we arrive at functions distinguished by some constant terms only, then we can not conclude the order of growth of the original functions using the \log function. Examples are $f(n) = 2^n, g(n) = 3^n$.

10 votes

-- srestha (87k points)

1.18.34 Time Complexity: TIFR2015-B-3 top

<https://gateoverflow.in/29846>



Answer: C.

It is fibonacci series generation. it takes exponential time if we won't use dynamic programming.

If we use dynamic programming then it takes $O(n)$

12 votes

-- pramod (3.3k points)

2**Compiler Design (175)** top**2.1****Abstract Syntax Tree(1)** top**2.1.1 Abstract Syntax Tree: GATE2015-2-14** top<https://gateoverflow.in/8084>

In the context of abstract-syntax-tree (AST) and control-flow-graph (CFG), which one of the following is TRUE?

- A. In both AST and CFG, let node N_2 be the successor of node N_1 . In the input program, the code corresponding to N_2 is present after the code corresponding to N_1
- B. For any input program, neither AST nor CFG will contain a cycle
- C. The maximum number of successors of a node in an AST and a CFG depends on the input program
- D. Each node in AST and CFG corresponds to at most one statement in the input program

[gate2015-2](#) [compiler-design](#) [easy](#) [abstract-syntax-tree](#)
Answer**Answers: Abstract Syntax Tree****2.1.1 Abstract Syntax Tree: GATE2015-2-14** top<https://gateoverflow.in/8084>

Selected Answer

Option (C) is Correct

- A. is false , In CFG , code of N_2 may be present before N_1 when there is a loop or Goto.
- B. is false , CFG contains cycle when input program has loop.
- C. is true ,successors in AST and CFG depend on Input program.
- D. is false, In CFG a single node may belong to a block of statements.

 26 votes
-- Himanshu Agarwal (15.3k points)
2.2**Assembler(6)** top**2.2.1 Assembler: GATE1992-01,viii** top<https://gateoverflow.in/553>

The purpose of instruction location counter in an assembler is _____

[gate1992](#) [compiler-design](#) [assembler](#) [normal](#)
Answer**2.2.2 Assembler: GATE1992-03,ii** top<https://gateoverflow.in/579>

Mention the pass number for each of the following activities that occur in a two pass assembler:

- A. object code generation
- B. literals added to literal table
- C. listing printed

- D. address resolution of local symbols

gate1992 compiler-design assembler easy

Answer

2.2.3 Assembler: GATE1993-7.6 [top](#)

<https://gateoverflow.in/2294>

A simple two-pass assembler does the following in the first pass:

- A. It allocates space for the literals.
- B. It computes the total length of the program.
- C. It builds the symbol table for the symbols and their values.
- D. It generates code for all the load and store register instructions.
- E. None of the above.

gate1993 compiler-design assembler easy

Answer

2.2.4 Assembler: GATE1994-17 [top](#)

<https://gateoverflow.in/2513>

State whether the following statements are True or False with reasons for your answer:

- A. Coroutine is just another name for a subroutine.
- B. A two pass assembler uses its machine opcode table in the first pass of assembly.

gate1994 compiler-design normal assembler

Answer

2.2.5 Assembler: GATE1994-18 [top](#)

<https://gateoverflow.in/2514>

State whether the following statements are True or False with reasons for your answer

- A. A subroutine cannot always be used to replace a macro in an assembly language program.
- B. A symbol declared as 'external' in an assembly language program is assigned an address outside the program by the assembler itself.

gate1994 compiler-design normal assembler true-false

Answer

2.2.6 Assembler: GATE1996-1.17 [top](#)

<https://gateoverflow.in/2721>

The pass numbers for each of the following activities

- i. object code generation
- ii. literals added to literal table
- iii. listing printed
- iv. address resolution of local symbols that occur in a two pass assembler

respectively are

- A. 1,2,1,2
- B. 2,1,2,1
- C. 2,1,1,2
- D. 1,2,2,2

[gate1996](#) [compiler-design](#) [normal](#) [assembler](#)

[Answer](#)

Answers: Assembler

2.2.1 Assembler: GATE1992-01,viii [top](#)

<https://gateoverflow.in/553>

Each section of an assembler language program has a location counter used to assign storage addresses to your program's statements. As the instructions of a source module are being assembled, the location counter keeps track of the current location in storage.

👍 6 votes

-- Rohan Ghosh (2k points)

2.2.2 Assembler: GATE1992-03,ii [top](#)

<https://gateoverflow.in/579>



Selected Answer

- a. 2
- b. 1
- c. 2
- d. 1

P.S. : In first pass, symbol table is created and In second pass, machine code is generated. Listing of final machine code is done after 2nd pass only.

👍 13 votes

-- Aditya Gaurav (3k points)

2.2.3 Assembler: GATE1993-7.6 [top](#)

<https://gateoverflow.in/2294>



Selected Answer

A, B, C are TRUE.

https://gateoverflow.in/?qa=blob&qa_blobid=2337905098612945492

👍 23 votes

-- Arjun Suresh (350k points)

2.2.4 Assembler: GATE1994-17 [top](#)

<https://gateoverflow.in/2513>

1. True

Subroutine are special cases of coroutine. Coroutine are generalized form of subroutine which is

nonpreemptive multitasking

<https://en.wikipedia.org/wiki/Croutine>

2. True. Basically machine opcode table is the symbol table which requires in 1st pass of two pass assembler

<http://users.cis.fiu.edu/~downeyt/cop3402/two-pass.htm>

6 votes

-- srestha (87k points)

2.2.5 Assembler: GATE1994-18 [top](#)

<https://gateoverflow.in/2514>

-  Selected Answer
- This is true. We can not replace macro entirely using subroutine. Ex -> Macro constant used for renaming.
 - This is false. This is job of Linker.

16 votes

-- Akash Kanase (42.5k points)

2.2.6 Assembler: GATE1996-1.17 [top](#)

<https://gateoverflow.in/2721>

 Selected Answer

Ans should be (B)

the functions performed in pass 1 and pass 2 in 2 pass assembler are

Pass 1

- Assign addresses to all statements in the program.
- Save the values assigned to all labels for use in pass 2
- Perform some processing of assembler directives.

Pass 2

- Assemble instructions.
- Generate data values defined by BYTE, WORD etc.
- Perform processing of assembler directives not done during pass 1.
- Write the program and the assembling listing

23 votes

-- sonam vyas (14.8k points)

2.3

Code Optimization(4) [top](#)

2.3.1 Code Optimization: GATE2008-12 [top](#)

<https://gateoverflow.in/410>

Some code optimizations are carried out on the intermediate code because

- They enhance the portability of the compiler to the target processor
- Program analysis is more accurate on intermediate code than on machine code
- The information from dataflow analysis cannot otherwise be used for optimization

- D. The information from the front end cannot otherwise be used for optimization

gate2008 | normal | code-optimization | compiler-design

Answer

2.3.2 Code Optimization: GATE2014-1-17 [top](#)

<https://gateoverflow.in/1784>

Which one of the following is **FALSE**?

- A. A basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end.
- B. Available expression analysis can be used for common subexpression elimination.
- C. Live variable analysis can be used for dead code elimination.
- D. $x = 4 * 5 \Rightarrow x = 20$ is an example of common subexpression elimination.

gate2014-1 | compiler-design | code-optimization | normal

Answer

2.3.3 Code Optimization: GATE2014-3-11 [top](#)

<https://gateoverflow.in/2045>

The minimum number of arithmetic operations required to evaluate the polynomial $P(X) = X^5 + 4X^3 + 6X + 5$ for a given value of X , using only one temporary variable is _____.

gate2014-3 | compiler-design | numerical-answers | normal | code-optimization

Answer

2.3.4 Code Optimization: GATE2014-3-34 [top](#)

<https://gateoverflow.in/2068>

Consider the basic block given below.

```
a = b + c
c = a + d
d = b + c
e = d - b
a = e + b
```

The minimum number of nodes and edges present in the DAG representation of the above basic block respectively are

- A. 6 and 6
- B. 8 and 10
- C. 9 and 12
- D. 4 and 4

gate2014-3 | compiler-design | code-optimization | normal

Answer

Answers: Code Optimization

2.3.1 Code Optimization: GATE2008-12 [top](#)

<https://gateoverflow.in/410>



Selected Answer

Ans is (A)

Intermediate codes are machine independent codes. So, intermediate code can be used for code optimization since a given source code can be converted to target machine code.

24 votes

-- Keith Kr (6.1k points)

2.3.2 Code Optimization: GATE2014-1-17 top

<https://gateoverflow.in/1784>



Selected Answer

- A. A basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end is TRUE.
- B. Available expression analysis can be used for common subexpression elimination is TRUE. Available expressions is an analysis algorithm that determines for each point in the program the set of expressions that need not be recomputed. Available expression analysis is used to do global common subexpression elimination (CSE). If an expression is available at a point, there is no need to re-evaluate it.
- C. Live variable analysis can be used for dead code elimination is TRUE.
- D. $x = 4 * 5 \Rightarrow x = 20$ is an example of common subexpression elimination is FALSE. Common subexpression elimination (CSE) refers to compiler optimization replaces identical expressions (i.e., they all evaluate to the same value) with a single variable holding the computed value when it is worthwhile to do so Source: Geeksforgeeks

15 votes

-- Pyuri sahu (2.1k points)

2.3.3 Code Optimization: GATE2014-3-11 top

<https://gateoverflow.in/2045>



Selected Answer

$$\begin{aligned}
 P(X) &= x^5 + 4x^3 + 6x + 5 \\
 &= x(x^4 + 4x^2 + 6) + 5 \\
 &= x(x(x^3 + 4x) + 6) + 5 \\
 &= x(x(x(x^2 + 4)) + 6) + 5 \\
 &= x(x(x(x(x) + 4)) + 6) + 5
 \end{aligned}$$

mul = pair of brackets 4
add = num of signs 3
total 7

54 votes

-- jayendra (8.2k points)

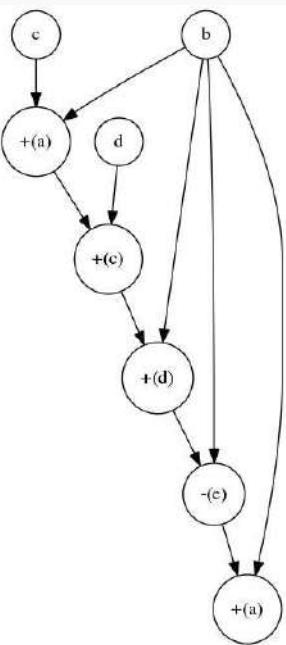
2.3.4 Code Optimization: GATE2014-3-34 top

<https://gateoverflow.in/2068>

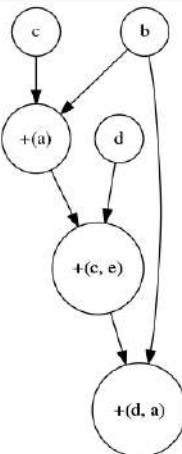


Selected Answer

A normal DAG construction will give 8 nodes and 10 edges as shown below.



Since, this question asks for minimum possible, we can assume algebraic simplification is allowed. So, $d = b + c, e = d - b$; can be simplified to $d = b + c; e = c$; Similarly, $e = d - b$; $a = e + b$; can be simplified to $a = d$. This gives the following DAG with 6 nodes and 6 edges.



<https://cs.nyu.edu/~gottlieb/courses/2000s/2006-07-fall/compilers/lectures/lecture-14.html> [working link]

35 votes

-- Arjun Suresh (350k points)

2.4

Compilation Phases(9) top

2.4.1 Compilation Phases: CMI2012-A-09 top

<https://gateoverflow.in/46538>

Consider the following programming errors:

- I. Type mismatch in an expression.
- II. Array index out of bounds.
- III. Use of an uninitialized variable in an expression.

Which of these errors will typically be caught at compile-time by a modern compiler.

- A. I, II and III
 B. I and II
 C. I and III
 D. None of them

cmi2012 | compiler-design | compilation-phases | normal

[Answer](#)

2.4.2 Compilation Phases: GATE1987-1-xi [top](#)

<https://gateoverflow.in/80282>

In a compiler the module which checks every character of the source text is called:

- A. The code generator.
 B. The code optimiser.
 C. The lexical analyser.
 D. The syntax analyser.

gate1987 | compiler-design | compilation-phases

[Answer](#)

2.4.3 Compilation Phases: GATE1990-2-ix [top](#)

<https://gateoverflow.in/84033>

Match the pairs in the following questions:

- | | |
|-----------------------|-------------------------|
| (a) Lexical analysis | (p) DAG's |
| (b) Code optimization | (q) Syntax trees |
| (c) Code generation | (r) Push down automaton |
| (d) Abelian groups | (s) Finite automaton |

gate1990 | match-the-following | compiler-design | compilation-phases

[Answer](#)

2.4.4 Compilation Phases: GATE2005-61 [top](#)

<https://gateoverflow.in/4066>

Consider line number 3 of the following C-program.

```
int main() {           /*Line 1 */  
int I, N;             /*Line 2 */  
fro (I=0, I<N, I++); /*Line 3 */  
}
```

Identify the compiler's response about this line while creating the object-module:

- A. No compilation error
 B. Only a lexical error
 C. Only syntactic errors
 D. Both lexical and syntactic errors

gate2005 | compiler-design | compilation-phases | normal

[Answer](#)

2.4.5 Compilation Phases: GATE2009-17 [top](#)

<https://gateoverflow.in/1309>

Match all items in Group 1 with the correct options from those given in Group 2.

Group 1		Group 2	
P.	Regular Expression	1.	Syntax analysis
Q.	Pushdown automata	2.	Code generation
R.	Dataflow analysis	3.	Lexical analysis
S.	Register allocation	4.	Code optimization

- A. P-4, Q-1, R-2, S-3
 - B. P-3, Q-1, R-4, S-2
 - C. P-3, Q-4, R-1, S-2
 - D. P-2, Q-1, R-4, S-3

gate2009 compiler-design easy compilation-phases

Answer

2.4.6 Compilation Phases: GATE2015-2-19 top

<https://gateoverflow.in/8098>

Match the following:

P. Lexical analysis 1. Graph coloring

Q. Parsing 2. DFA minimization

R. Register 3. Post-order allocation traversal

S. Expression 4. Production tree evaluation

- A. P-2, Q-3, R-1, S-4
 - B. P-2, Q-1, R-4, S-3
 - C. P-2, Q-4, R-1, S-3
 - D. P-2, Q-3, R-4, S-1

gate2015-2 compiler-design normal compilation-phases

Answer

2.4.7 Compilation Phases: GATE2016-2-19 top

<https://gateoverflow.in/39548>

Match the following:

(Q) Top down parsing (ii) Type checking

(R) Semantic analysis (iii) Regular expressions

(S) Runtime environment (iv) Activation records

- A. $P \leftrightarrow i, Q \leftrightarrow ii, R \leftrightarrow iv, S \leftrightarrow iii$
- B. $P \leftrightarrow iii, Q \leftrightarrow i, R \leftrightarrow ii, S \leftrightarrow iv$
- C. $P \leftrightarrow ii, Q \leftrightarrow iii, R \leftrightarrow i, S \leftrightarrow iv$
- D. $P \leftrightarrow iv, Q \leftrightarrow i, R \leftrightarrow ii, S \leftrightarrow iii$

gate2016-2 compiler-design easy compilation-phases

[Answer](#)

2.4.8 Compilation Phases: GATE2017-2-05 [top](#)

<https://gateoverflow.in/118592>

Match the following according to input (from the left column) to the compiler phase (in the right column) that processes it:

P. Syntax tree	i. Code generator
Q. Character stream	ii. Syntax analyser
R. Intermediate representation	iii. Semantic analyser
S. Token stream	iv. Lexical analyser

- A. P-ii; Q-iii; R-iv; S-i
- B. P-ii; Q-i; R-iii; S-iv
- C. P-iii; Q-iv; R-i; S-ii
- D. P-i; Q-iv; R-ii; S-iii

gate2017-2 compiler-design compilation-phases easy

[Answer](#)

2.4.9 Compilation Phases: GATE2018-8 [top](#)

<https://gateoverflow.in/204082>

Which one of the following statements is FALSE?

- A. Context-free grammar can be used to specify both lexical and syntax rules
- B. Type checking is done before parsing
- C. High-level language programs can be translated to different Intermediate Representations
- D. Arguments to a function can be passed using the program stack

gate2018 compiler-design easy compilation-phases

[Answer](#)

Answers: Compilation Phases

2.4.1 Compilation Phases: CMI2012-A-09 [top](#)

<https://gateoverflow.in/46538>



Selected Answer

Type mismatch gives compile time error for statically typed languages (type of variables are determined at compile time) like C.

Use of uninitialized variable can be detected by compiler? Not always- but possible in most cases. There can be cases where there are two branches and a variable is initialized in one but not in other. But, at compile time we can assume that we are only interested in those variables which are

uninitialized in all the branches,

Array index out of bounds- can be detected by compiler in some cases. Consider

```
int a[100];
a[200] = 4;
```

A decent C compiler should catch this. But if the index is a variable it becomes more difficult to catch at compile time.

So, answer should be **C**.

8 votes

-- Arjun Suresh (350k points)

2.4.2 Compilation Phases: GATE1987-1-xi [top](#)

<https://gateoverflow.in/80282>



Selected Answer

lexical analyser phase checks every character of text to identify tokens..

12 votes

-- kirti singh (3.8k points)

2.4.3 Compilation Phases: GATE1990-2-ix [top](#)

<https://gateoverflow.in/84033>



Selected Answer

- (a) Lexical analysis (s) Finite automaton (Dfa creation for finding token)
- (b) Code optimization (p) DAG's (common subtree minimization.)
- (c) Code generation can be
used for code generation)
- (d) Abelian groups (r) Push down automaton (left one)

22 votes

-- Prashant Singh (59.8k points)

2.4.4 Compilation Phases: GATE2005-61 [top](#)

<https://gateoverflow.in/4066>



Selected Answer

C language allows only certain words in it- these are called tokens. If we input any invalid tokens it causes lexical error.

eg:

44a44

causes lexical error as in C as an alphabet cannot come in between digits.

Syntactic error is caused by bad combination of tokens. For example, we cannot have a constant on the left hand side of an assignment statement, a for loop must have two expressions inside () separated by semi colon etc.

In the given question, line 3 won't cause a lexical error or syntactic error. The statement will be treated as a function call with three arguments. Function definition being absent will cause link time error, but the question asks only for compile-time errors. So, (a) must be the answer.

PS: Implicit function declaration was removed from C99 standard onwards. As per current standard, we should not use a function without declaration. Still, we cannot guarantee "compilation error"- just expect compiler warnings in C. In C++ this should produce a compilation (semantic)

error. The output of compiling the above code using different standards are given below:

```
arjun@linux:~$ gcc -c chk.c
chk.c: In function 'main':
chk.c:3:2: warning: implicit declaration of function 'fro' [-Wimplicit-function-declaration]
  fro (I=0, I<N, I++); /*Line 3 */
^
arjun@linux:~$ gcc -c -ansi chk.c
arjun@linux:~$ gcc -c -std=c99 chk.c
chk.c: In function 'main':
chk.c:3:2: warning: implicit declaration of function 'fro' [-Wimplicit-function-declaration]
  fro (I=0, I<N, I++); /*Line 3 */
^
arjun@linux:~$ gcc -c -std=c11 chk.c
chk.c: In function 'main':
chk.c:3:2: warning: implicit declaration of function 'fro' [-Wimplicit-function-declaration]
  fro (I=0, I<N, I++); /*Line 3 */
```

<http://stackoverflow.com/questions/15570553/lexical-and-semantic-errors-in-c>

47 votes

-- Arjun Suresh (350k points)

2.4.5 Compilation Phases: GATE2009-17 top

<https://gateoverflow.in/1309>



Selected Answer

Regular expressions are used in lexical analysis.

Pushdown automata is related to context free grammar which is related to syntax analysis.

Dataflow analysis is done in code optimization.

Register allocation is done in code generation.

Ans **B**

31 votes

-- Keith Kr (6.1k points)

2.4.6 Compilation Phases: GATE2015-2-19 top

<https://gateoverflow.in/8098>



Selected Answer

1. Regular expression uses FA & Regular Sets.
2. Expression can be evaluated with postfix Traversals .
3. Register allocation can be modeled by graph coloring.
4. The parser constructs a production tree.

so answer is C

21 votes

-- shekhar chauhan (45.2k points)

2.4.7 Compilation Phases: GATE2016-2-19 top

<https://gateoverflow.in/39548>



Selected Answer

Option B.

Lexical Analysis phase uses regular expressions.

LMD is involved in top down parsing.

Type checking is done in semantic analysis phase.

Activation records are related to Run Time Environments

25 votes

-- Sharathkumar Anbu (671 points)

2.4.8 Compilation Phases: GATE2017-2-05 [top](#)

<https://gateoverflow.in/118592>

Selected Answer

Q - iv because Character stream is given as input to lexical analyser.

P - iii Syntax tree is given as input to semantic analyser

R - i Intermediate code given as input to code generator

S - ii Token stream given as input to syntax analyser.

Ans: C

19 votes

-- Arnabi Bej (7.9k points)

2.4.9 Compilation Phases: GATE2018-8 [top](#)

<https://gateoverflow.in/204082>

Selected Answer

A. Since Lexical rules are nothing but regular expressions, we can use CFGs to represent such rules.(Every Type-3 grammar is Type-2 grammar) Additionally, syntax rules can be represented by CFGs. (True)

B. Type checking is done during Semantic Analysis phase which comes after Parsing. (False)

C. We have various types of Intermediate Code Representations, ex 3-address code, Postfix notation, Syntax trees. (True)

D. Program stack holds the activation record of the function called, which stores function parameters, return value, return address etc.(True)

4 votes

-- Neelay Upadhyaya (1.1k points)

2.5

Expression Evaluation(2) [top](#)

2.5.1 Expression Evaluation: GATE2002-2.19 [top](#)

<https://gateoverflow.in/849>

To evaluate an expression without any embedded function calls

A. One stack is enough

- B. Two stacks are needed
 C. As many stacks as the height of the expression tree are needed
 D. A Turing machine is needed in the general case

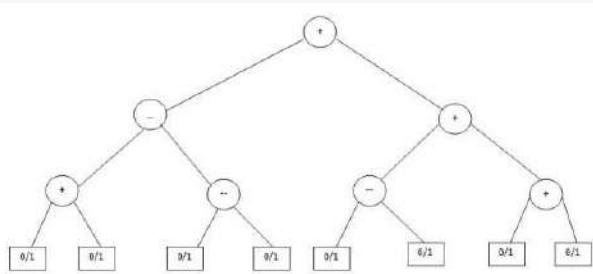
gate2002 compiler-design expression-evaluation easy

[Answer](#)

2.5.2 Expression Evaluation: GATE2014-2-39 [top](#)

<https://gateoverflow.in/1999>

Consider the expression tree shown. Each leaf represents a numerical value, which can either be 0 or 1. Over all possible choices of the values at the leaves, the maximum possible value of the expression represented by the tree is ____.



gate2014-2 compiler-design normal expression-evaluation numerical-answers

[Answer](#)

Answers: Expression Evaluation

2.5.1 Expression Evaluation: GATE2002-2.19 [top](#)

<https://gateoverflow.in/849>



Selected Answer

Expression without any calls in it => $1 + 2 * 3 - 4$

Expression with embedded calls => $1 + \text{fun1}(a, b, c) * \text{fun2}(3.4, 58) - \text{fun3}(x, yz) ;$

First we can convert Infix to Postfix using single stack (Using it as operator stack)

Then we can evaluate that expression using Single stack.

15 votes

-- Akash Kanase (42.5k points)

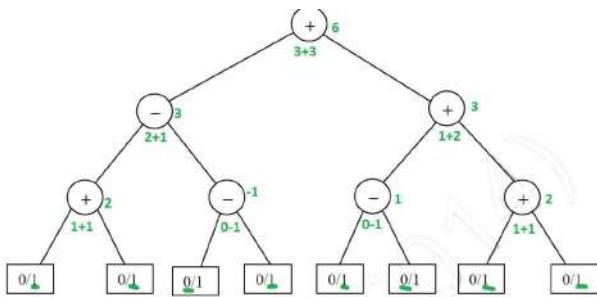
2.5.2 Expression Evaluation: GATE2014-2-39 [top](#)

<https://gateoverflow.in/1999>



Selected Answer

An [Expression Tree](#) is a binary tree in which each internal node corresponds to operator and each leaf node corresponds to operand so for example expression tree for $3 + ((5+9)*2)$ would be:. Below diagram shows values to pick to get the maximum value in expression tree



8 votes

-- Divya Bharti (8.1k points)

2.6

Grammar(39) top

2.6.1 Grammar: GATE1990-16a top

<https://gateoverflow.in/86869>

Show that grammar G_1 is ambiguous using parse trees:

$$G_1 : S \rightarrow \text{if } S \text{ then } S \text{ else } S$$

$$S \rightarrow \text{if } S \text{ then } S$$

gate1990 descriptive compiler-design grammar

Answer

2.6.2 Grammar: GATE1991-10a top

<https://gateoverflow.in/537>

Consider the following grammar for arithmetic expressions using binary operators — and / which are not associative

$$E \rightarrow E - T \mid T$$

$$T \rightarrow T/F \mid F$$

$$F \rightarrow (E) \mid id$$

(E is the start symbol)

Is the grammar unambiguous? If so, what is the relative precedence between — and /? If not, give an unambiguous grammar that gives / precedence over —.

gate1991 grammar compiler-design normal descriptive

Answer

2.6.3 Grammar: GATE1991-10b top

<https://gateoverflow.in/43604>

Consider the following grammar for arithmetic expressions using binary operators — and / which are not associative

$$E \rightarrow E - T \mid T$$

$$T \rightarrow T/F \mid F$$

$$F \rightarrow (E) \mid id$$

(E is the start symbol)

Does the grammar allow expressions with redundant parentheses as in (id/id) or in $id - (id/id)$? If so, convert the grammar into one which does not generate expressions with redundant parentheses. Do this with minimum number of changes to the given production rules and adding at most one more production rule.

[gate1991](#) [grammar](#) [compiler-design](#) [normal](#) [descriptive](#)

Answer

2.6.4 Grammar: GATE1991-10c [top](#)

<https://gateoverflow.in/43605>

Consider the following grammar for arithmetic expressions using binary operators $-$ and $/$ which are not associative

$$E \rightarrow E - T \mid T$$

$$T \rightarrow T/F \mid F$$

$$F \rightarrow (E) \mid id$$

(E is the start symbol)

Does the grammar allow expressions with redundant parentheses as in (id/id) or in $id - (id/id)$? If so, convert the grammar into one which does not generate expressions with redundant parentheses. Do this with minimum number of changes to the given production rules and adding at most one more production rule.

Convert the grammar obtained above into one that is not left recursive.

[gate1991](#) [grammar](#) [compiler-design](#) [normal](#) [descriptive](#)

Answer

2.6.5 Grammar: GATE1994-1.18 [top](#)

<https://gateoverflow.in/2461>

Which of the following features cannot be captured by context-free grammars?

- A. Syntax of if-then-else statements
- B. Syntax of recursive procedures
- C. Whether a variable has been declared before its use
- D. Variable names of arbitrary length

[gate1994](#) [compiler-design](#) [grammar](#) [normal](#)

Answer

2.6.6 Grammar: GATE1994-20 [top](#)

<https://gateoverflow.in/2516>

A grammar G is in Chomsky-Normal Form (CNF) if all its productions are of the form $A \rightarrow BC$ or $A \rightarrow a$, where A, B and C , are non-terminals and a is a terminal. Suppose G is a CFG in CNF and w is a string in $L(G)$ of length n , then how long is a derivation of w in G ?

[gate1994](#) [compiler-design](#) [grammar](#) [normal](#)

Answer

2.6.7 Grammar: GATE1994-3.5 [top](#)

<https://gateoverflow.in/2482>

Match the following items

(i) Backus-Naur form	(a) Regular expressions
(ii) Lexical analysis	(b) LALR(1) grammar
(iii) YACC	(c) LL(1) grammars
(iv) Recursive descent parsing	(d) General context-free grammars

gate1994 compiler-design grammar normal

Answer**2.6.8 Grammar: GATE1995-1.10** [top](#)<https://gateoverflow.in/2597>

Consider a grammar with the following productions

- $S \rightarrow a\alpha b \mid b\alpha c \mid aB$
- $S \rightarrow \alpha S \mid b$
- $S \rightarrow \alpha bb \mid ab$
- $S\alpha \rightarrow bdb \mid b$

The above grammar is:

- A. Context free
- B. Regular
- C. Context sensitive
- D. $LR(k)$

gate1995 compiler-design grammar normal

Answer**2.6.9 Grammar: GATE1995-9** [top](#)<https://gateoverflow.in/2644>

- A. Translate the arithmetic expression $a^* - (b + c)$ into syntax tree.
- B. A grammar is said to have cycles if it is the case that

$$A \Rightarrow^+ A$$

Show that no grammar that has cycles can be LL(1).

gate1995 compiler-design grammar normal

Answer**2.6.10 Grammar: GATE1996-11** [top](#)<https://gateoverflow.in/2763>

Let G be a context-free grammar where $G = (\{S, A, B, C\}, \{a, b, d\}, P, S)$ with the productions in P given below.

$$S \rightarrow ABAC$$

$$A \rightarrow aA \mid \epsilon$$

$$B \rightarrow bB \mid \epsilon$$

$$C \rightarrow d$$

(ϵ denotes the null string). Transform the grammar G to an equivalent context-free grammar G' that has no ϵ productions and no unit productions. (A unit production is of the form $x \rightarrow y$, and x and y are non terminals).

[gate1996](#) [compiler-design](#) [grammar](#) [normal](#)

Answer

2.6.11 Grammar: GATE1996-2.10 [top](#)

<https://gateoverflow.in/2739>

The grammar whose productions are

- > if id then <stmt>
- > if id then <stmt> else <stmt>
- > id := id

is ambiguous because

(a) the sentence

```
if a then if b then c:= d
```

has more than two parse trees

(b) the left most and right most derivations of the sentence

```
if a then if b then c:= d
```

give rise to different parse trees

(c) the sentence

```
if a then if b then c:= d else c:= f
```

has more than two parse trees

(d) the sentence

```
if a then if b then c:= d else c:= f
```

has two parse trees

[gate1996](#) [compiler-design](#) [grammar](#) [normal](#)

Answer

2.6.12 Grammar: GATE1997-1.6 [top](#)

<https://gateoverflow.in/2222>

In the following grammar

$$\begin{aligned} X &::= X \oplus Y \mid Y \\ Y &::= Z * Y \mid Z \\ Z &\quad ::= id \end{aligned}$$

Which of the following is true?

- A. ' \oplus ' is left associative while '*' is right associative
- B. Both ' \oplus ' and '*' are left associative
- C. ' \oplus ' is right associative while '*' is left associative
- D. None of the above

[gate1997](#) [compiler-design](#) [grammar](#) [normal](#)

Answer**2.6.13 Grammar: GATE1997-11** [top](#)<https://gateoverflow.in/2271>

Consider the grammar

- $S \rightarrow bSe$
- $S \rightarrow PQR$
- $P \rightarrow bPc$
- $P \rightarrow \epsilon$
- $Q \rightarrow cQd$
- $Q \rightarrow \epsilon$
- $R \rightarrow dRe$
- $R \rightarrow \epsilon$

where S, P, Q, R are non-terminal symbols with S being the start symbol; b, c, d, e are terminal symbols and ' ϵ ' is the empty string. This grammar generates strings of the form b^i, c^j, d^k, e^m for some $i, j, k, m \geq 0$.

- What is the condition on the values of i, j, k, m ?
- Find the smallest string that has two parse trees.

[gate1997](#) [compiler-design](#) [grammar](#) [normal](#) [theory-of-computation](#)
Answer**2.6.14 Grammar: GATE1998-14** [top](#)<https://gateoverflow.in/1728>

- A. Let $G_1 = (N, T, P, S_1)$ be a CFG where, $N = \{S_1 A, B\}, T = \{a, b\}$ and P is given by

$$S_1 \rightarrow aS_1b \quad S_1 \rightarrow aBb$$

$$S_1 \rightarrow aAb \quad B \rightarrow Bb$$

$$A \rightarrow aA \quad B \rightarrow b$$

$$A \rightarrow a$$

What is $L(G_1)$?

- B. Use the grammar in Part(a) to give a CFG for $L_2 = \{a^i b^j a^k b^l \mid i, j, k, l \geq 1, i = j \text{ or } k = l\}$ by adding not more than 5 production rules.
- C. Is L_2 inherently ambiguous?

[gate1998](#) [compiler-design](#) [grammar](#) [descriptive](#)
Answer**2.6.15 Grammar: GATE1998-6b** [top](#)<https://gateoverflow.in/1697>

Consider the grammar

- $S \rightarrow Aa \mid b$

- $A \rightarrow Ac \mid Sd \mid \epsilon$

Construct an equivalent grammar with no left recursion and with minimum number of production rules.

gate1998 compiler-design grammar descriptive

[Answer](#)

2.6.16 Grammar: GATE1999-2.15 [top](#)

<https://gateoverflow.in/1493>

A grammar that is both left and right recursive for a non-terminal, is

- Ambiguous
- Unambiguous
- Information is not sufficient to decide whether it is ambiguous or unambiguous
- None of the above

gate1999 compiler-design grammar normal

[Answer](#)

2.6.17 Grammar: GATE2000-2.21, ISRO2015-24 [top](#)

<https://gateoverflow.in/668>

Given the following expression grammar:

$$E \rightarrow E * F \mid F + E \mid F$$

$$F \rightarrow F - F \mid id$$

Which of the following is true?

- * has higher precedence than +
- has higher precedence than *
- + and - have same precedence
- + has higher precedence than *

gate2000 grammar normal compiler-design isro2015

[Answer](#)

2.6.18 Grammar: GATE2001-1.18 [top](#)

<https://gateoverflow.in/711>

Which of the following statements is false?

- An unambiguous grammar has same leftmost and rightmost derivation
- An LL(1) parser is a top-down parser
- LALR is more powerful than SLR
- An ambiguous grammar can never be LR(k) for any k

gate2001 compiler-design grammar normal

[Answer](#)

2.6.19 Grammar: GATE2001-18 [top](#)

<https://gateoverflow.in/759>

- A. Remove left-recursion from the following grammar: $S \rightarrow Sa \mid Sb \mid a \mid b$
 B. Consider the following grammar:

$$S \rightarrow aSbS \mid bSaS \mid \epsilon$$

Construct all possible parse trees for the string abab. Is the grammar ambiguous?

gate2001 compiler-design grammar descriptive

Answer

2.6.20 Grammar: GATE2003-56 [top](#)

<https://gateoverflow.in/944>

Consider the grammar shown below

- $S \rightarrow iEtSS' \mid a$
- $S' \rightarrow eS \mid \epsilon$
- $E \rightarrow b$

In the predictive parse table, M , of this grammar, the entries $M[S', e]$ and $M[S', \$]$ respectively are

- A. $\{S' \rightarrow eS\}$ and $\{S' \rightarrow \epsilon\}$
 B. $\{S' \rightarrow eS\}$ and $\{\}$
 C. $\{S' \rightarrow \epsilon\}$ and $\{S' \rightarrow \epsilon\}$
 D. $\{S' \rightarrow eS, S' \rightarrow \epsilon\}$ and $\{S' \rightarrow \epsilon\}$

gate2003 compiler-design grammar normal

Answer

2.6.21 Grammar: GATE2003-58 [top](#)

<https://gateoverflow.in/946>

Consider the translation scheme shown below.

$$S \rightarrow T R$$

$$R \rightarrow + T \{print('+');\} R \mid \epsilon$$

$$T \rightarrow \text{num} \{print(\text{num}.val);\}$$

Here **num** is a token that represents an integer and **num.val** represents the corresponding integer value. For an input string '9 + 5 + 2', this translation scheme will print

- A. 9 + 5 + 2
 B. 9 5 + 2 +
 C. 9 5 2 + +
 D. + + 9 5 2

gate2003 compiler-design grammar normal

Answer

2.6.22 Grammar: GATE2004-45 [top](#)

<https://gateoverflow.in/1042>

Consider the grammar with the following translation rules and E as the start symbol

$E \rightarrow E_1 \# T$	$\{E.value = E_1.value * T.value\}$
 $T \rightarrow T_1 \& F$	$\{E.value = T.value\}$
 $F \rightarrow \text{num}$	$\{T.value = T_1.value + F.value\}$
	$\{T.value = F.value\}$
	$\{F.value = \text{num.value}\}$

Compute E.value for the root of the parse tree for the expression: $2 \# 3 \& 5 \# 6 \& 4$

- A. 200
- B. 180
- C. 160
- D. 40

gate2004 compiler-design grammar normal

Answer

2.6.23 Grammar: GATE2004-8 top

<https://gateoverflow.in/1005>

Which of the following grammar rules violate the requirements of an operator grammar? P, Q, R are nonterminals, and r, s, t are terminals.

- I. $P \rightarrow Q R$
 - II. $P \rightarrow Q s R$
 - III. $P \rightarrow \epsilon$
 - IV. $P \rightarrow Q t R r$
- A. (I) only
 B. (I) and (III) only
 C. (II) and (III) only
 D. (III) and (IV) only

gate2004 compiler-design grammar normal

Answer

2.6.24 Grammar: GATE2004-88 top

<https://gateoverflow.in/1082>

Consider the following grammar G:

$$S \rightarrow bS \mid aA \mid b$$

$$A \rightarrow bA \mid aB$$

$$B \rightarrow bB \mid aS \mid a$$

Let $N_a(w)$ and $N_b(w)$ denote the number of a's and b's in a string w respectively.

The language $L(G)$ over $\{a, b\}^+$ generated by G is

- A. $\{w \mid N_a(w) > 3N_b(w)\}$
- B. $\{w \mid N_b(w) > 3N_a(w)\}$
- C. $\{w \mid N_a(w) = 3k, k \in \{0, 1, 2, \dots\}\}$

- D. $\{w \mid N_b(w) = 3k, k \in \{0, 1, 2, \dots\}\}$

gate2004 compiler-design grammar normal

Answer

2.6.25 Grammar: GATE2005-59 [top](#)

<https://gateoverflow.in/1382>

Consider the grammar:

$$E \rightarrow E + n \mid E \times n \mid n$$

For a sentence $n + n \times n$, the handles in the right-sentential form of the reduction are:

- A. $n, E + n$ and $E + n \times n$
- B. $n, E + n$ and $E + E \times n$
- C. $n, n + n$ and $n + n \times n$
- D. $n, E + n$ and $E \times n$

gate2005 compiler-design grammar normal

Answer

2.6.26 Grammar: GATE2006-32, ISRO2016-35 [top](#)

<https://gateoverflow.in/995>

Consider the following statements about the context free grammar

$$G = \{S \rightarrow SS, S \rightarrow ab, S \rightarrow ba, S \rightarrow \epsilon\}$$

- I. G is ambiguous
- II. G produces all strings with equal number of a 's and b 's
- III. G can be accepted by a deterministic PDA.

Which combination below expresses all the true statements about G ?

- A. I only
- B. I and III only
- C. II and III only
- D. I, II and III

gate2006 compiler-design grammar normal isro2016

Answer

2.6.27 Grammar: GATE2006-59 [top](#)

<https://gateoverflow.in/1837>

Consider the following translation scheme.

- $S \rightarrow ER$
- $R \rightarrow^* E \{print('*');\} R \mid \epsilon$
- $E \rightarrow F + E \{print('+');\} \mid F$
- $F \rightarrow (S) \mid id \{print(id.value);\}$

Here id is a token that represents an integer and $id.value$ represents the corresponding integer value. For an input ' $2 * 3 + 4$ ', this translation scheme prints

- A. $2 * 3 + 4$
 B. $2 * + 3 4$
 C. $2 3 * 4 +$
 D. $2 3 4 + *$

gate2006 compiler-design grammar normal

Answer

2.6.28 Grammar: GATE2006-84 top

<https://gateoverflow.in/1856>

Which one of the following grammars generates the language $L = \{a^i b^j \mid i \neq j\}$?

- A. $S \rightarrow AC \mid CB$
 $C \rightarrow aCb \mid a \mid b$
 $A \rightarrow aA \mid \epsilon$
 $B \rightarrow Bb \mid \epsilon$
- B. $S \rightarrow aS \mid Sb \mid a \mid b$
- C. $S \rightarrow AC \mid CB$
 $C \rightarrow aCb \mid \epsilon$
 $A \rightarrow aA \mid \epsilon$
 $B \rightarrow Bb \mid \epsilon$
- D. $S \rightarrow AC \mid CB$
 $C \rightarrow aCb \mid \epsilon$
 $A \rightarrow aA \mid a$
 $B \rightarrow Bb \mid b$

gate2006 compiler-design grammar normal theory-of-computation

Answer

2.6.29 Grammar: GATE2006-85 top

<https://gateoverflow.in/79801>

The grammar

- $S \rightarrow AC \mid CB$
- $C \rightarrow aCb \mid \epsilon$
- $A \rightarrow aA \mid a$
- $B \rightarrow Bb \mid b$

generates the language $L = \{a^i b^j \mid i \neq j\}$. In that grammar what is the length of the derivation (number of steps starting from S) to generate the string $a^l b^m$ with $l \neq m$

- A. $\max(l, m) + 2$
 B. $l + m + 2$
 C. $l + m + 3$
 D. $\max(l, m) + 3$

gate2006 compiler-design grammar normal

Answer

2.6.30 Grammar: GATE2007-52 top

<https://gateoverflow.in/1250>

Consider the grammar with non-terminals $N = \{S, C, S_1\}$, terminals $T = \{a, b, i, t, e\}$, with S as the start symbol, and the following set of rules:

$$S \rightarrow iCtSS_1 \mid a$$

$$S_1 \rightarrow eS \mid \epsilon$$

$$C \rightarrow b$$

The grammar is NOT LL(1) because:

- A. it is left recursive
- B. it is right recursive
- C. it is ambiguous
- D. it is not context-free

[gate2007](#) [compiler-design](#) [grammar](#) [normal](#)

Answer

2.6.31 Grammar: GATE2007-53 [top](#)

<https://gateoverflow.in/1251>

Consider the following two statements:

- P: Every regular grammar is LL(1)
- Q: Every regular set has a LR(1) grammar

Which of the following is **TRUE**?

- A. Both P and Q are true
- B. P is true and Q is false
- C. P is false and Q is true
- D. Both P and Q are false

[gate2007](#) [compiler-design](#) [grammar](#) [normal](#)

Answer

2.6.32 Grammar: GATE2007-78 [top](#)

<https://gateoverflow.in/1272>

Consider the CFG with $\{S, A, B\}$ as the non-terminal alphabet, $\{a, b\}$ as the terminal alphabet, S as the start symbol and the following set of production rules:

$$S \rightarrow aB \quad S \rightarrow bA$$

$$B \rightarrow b \quad A \rightarrow a$$

$$B \rightarrow bS \quad A \rightarrow aS$$

$$B \rightarrow aBB \quad S \rightarrow bAA$$

Which of the following strings is generated by the grammar?

- A. *aaaabb*
- B. *aabbbb*
- C. *aabbab*

D. $abba$

gate2007 compiler-design grammar normal

[Answer](#)

2.6.33 Grammar: GATE2007-79 [top](#)

<https://gateoverflow.in/43512>

Consider the CFG with $\{S, A, B\}$ as the non-terminal alphabet, $\{a, b\}$ as the terminal alphabet, S as the start symbol and the following set of production rules:

$$\begin{array}{ll} S \rightarrow aB & S \rightarrow bA \\ B \rightarrow b & A \rightarrow a \\ B \rightarrow bS & A \rightarrow aS \\ B \rightarrow aBB & S \rightarrow bAA \end{array}$$

For the string $aabbab$, how many derivation trees are there?

- A. 1
- B. 2
- C. 3
- D. 4

gate2007 compiler-design grammar normal

[Answer](#)

2.6.34 Grammar: GATE2007-IT-9 [top](#)

<https://gateoverflow.in/3442>

Consider an ambiguous grammar G and its disambiguated version D . Let the language recognized by the two grammars be denoted by $L(G)$ and $L(D)$ respectively. Which one of the following is true?

- A. $L(D) \subset L(G)$
- B. $L(D) \supset L(G)$
- C. $L(D) = L(G)$
- D. $L(D)$ is empty

gate2007-it compiler-design grammar normal

[Answer](#)

2.6.35 Grammar: GATE2008-50 [top](#)

<https://gateoverflow.in/395>

Which of the following statements are true?

- I. Every left-recursive grammar can be converted to a right-recursive grammar and vice-versa
 - II. All ϵ -productions can be removed from any context-free grammar by suitable transformations
 - III. The language generated by a context-free grammar all of whose productions are of the form $X \rightarrow w$ or $X \rightarrow wY$ (where, w is a string of terminals and Y is a non-terminal), is always regular
 - IV. The derivation trees of strings generated by a context-free grammar in Chomsky Normal Form are always binary trees
- A. I, II, III and IV
 - B. II, III and IV only

- C. I, III and IV only
D. I, II and IV only

gate2008 normal compiler-design grammar

Answer

2.6.36 Grammar: GATE2010-38 [top](#)

<https://gateoverflow.in/2339>

The grammar $S \rightarrow aSa \mid bS \mid c$ is

- A. LL(1) but not LR(1)
B. LR(1) but not LL(1)
C. Both LL(1) and LR(1)
D. Neither LL(1) nor LR(1)

gate2010 compiler-design grammar normal

Answer

2.6.37 Grammar: GATE2014-2-17 [top](#)

<https://gateoverflow.in/1973>

Consider the grammar defined by the following production rules, with two operators * and +

- $S \rightarrow T * P$
- $T \rightarrow U \mid T * U$
- $P \rightarrow Q + P \mid Q$
- $Q \rightarrow Id$
- $U \rightarrow Id$

Which one of the following is TRUE?

- A. + is left associative, while * is right associative
B. + is right associative, while * is left associative
C. Both + and * are right associative
D. Both + and * are left associative

gate2014-2 compiler-design grammar normal

Answer

2.6.38 Grammar: GATE2016-2-45 [top](#)

<https://gateoverflow.in/39594>

Which one of the following grammars is free from left recursion?

- A. $S \rightarrow AB$
 $A \rightarrow Aa \mid b$
 $B \rightarrow c$
- B. $S \rightarrow Ab \mid Bb \mid c$
 $A \rightarrow Bd \mid \epsilon$
 $B \rightarrow e$
- C. $S \rightarrow Aa \mid B$
 $A \rightarrow Bb \mid Sc \mid \epsilon$
 $B \rightarrow d$
- D. $S \rightarrow Aa \mid Bb \mid c$

$$\begin{array}{l} A \rightarrow Bd | \epsilon \\ B \rightarrow Ae | \epsilon \end{array}$$

gate2016-2 compiler-design grammar easy

Answer

2.6.39 Grammar: GATE2016-2-46 top

<https://gateoverflow.in/39598>

A student wrote two context-free grammars G1 and G2 for generating a single C-like array declaration. The dimension of the array is at least one. For example,

```
int a[10] [3];
```

The grammars use D as the start symbol, and use six terminal symbols **int** ; **id** [] **num**.

Grammar **G1**

$$\begin{array}{ll} D \rightarrow \text{int } L; & D \rightarrow \text{int } L; \\ L \rightarrow \text{id } [E] & L \rightarrow \text{id } E \\ E \rightarrow \text{num }] & E \rightarrow E [\text{num}] \\ E \rightarrow \text{num }] [E & E \rightarrow [\text{num}] \end{array}$$

Grammar **G2**

Which of the grammars correctly generate the declaration mentioned above?

- A. Both **G1** and **G2**
- B. Only **G1**
- C. Only **G2**
- D. Neither **G1** nor **G2**

gate2016-2 compiler-design grammar normal

Answer

Answers: Grammar

2.6.1 Grammar: GATE1990-16a top

<https://gateoverflow.in/86869>



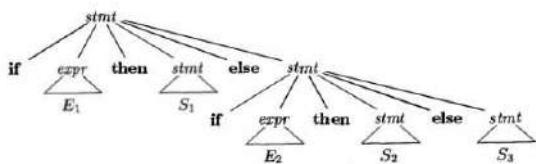
Selected Answer

The given grammer is well known as " [Dangling Else](#) " problem . The given grammer is ambiguous and ambiguity can be resolved .

$$\begin{array}{l} \text{stmt} \rightarrow \text{if expr then stmt} \\ \quad \quad \quad \mid \text{if expr then stmt else stmt} \\ \quad \quad \quad \mid \text{other} \end{array}$$

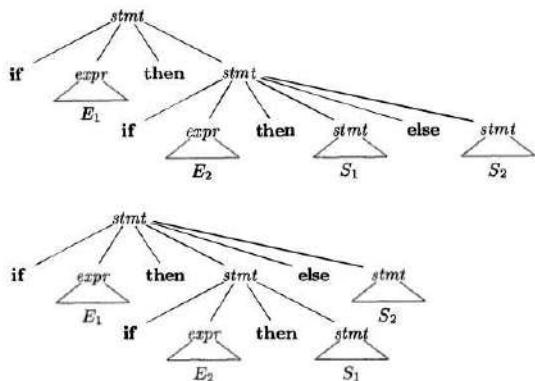
Consider the compound conditional statement for the above grammer

if *E1* **then** *S1* **else** **if** *E2* **then** *S2* **else** *S3* has the following parse tree



Well this is ambiguous due to the statement
if E1 then if E2 then S1 else S2

The two parse trees are



Ambiguity can be resolved by parsers as follows

In all programming languages with conditional statements of this form, the first parse tree is preferred.

The general rule is, "Match each else with the closest unmatched then"

In practice it is rarely built into the productions .

```

stmt → matched_stmt
      | open_stmt
matched_stmt → if expr then matched_stmt else matched_stmt
      | other
open_stmt → if expr then stmt
      | if expr then matched_stmt else open_stmt
    
```

However this grammar is also ambiguous .

! Exercise 4.3.3: The following grammar is proposed to remove the "dangling-else ambiguity" discussed in Section 4.3.2:

```

stmt → if expr then stmt
      | matchedStmt
matchedStmt → if expr then matchedStmt else stmt
      | other
    
```

Show that this grammar is still ambiguous.

Solution is Given [Here](#) . Moreover there is no grammar exists that will **eliminate** ambiguity in Dangling Else Grammer . Hence we can say that Languge produced by dangling else grammar is "**inherently ambiguous**" .

Try running this code in any compilers . No doubt that all compilers will successfully parse and produce output

```
#include <stdio.h>
int main(void) {
```

```

if (1<3)
if (1<2)
printf("1 is the smallest");
else
printf("2 is the smallest");
return 0;
}

```

Thus we can say that ambiguity in Dangling Else Problem can be resolved but Cannot be Eliminated from the grammar .

Source

- [Compilers: Principles, Techniques, & Tools -Aho & Ullman](#)

Good Read

- <http://cs.stackexchange.com/questions/32475/resolving-ambiguity-in-dangling-else>
- <https://gateoverflow.in/54718/inherently-ambiguous-languages-deterministic-context-grammars>
- <https://gateoverflow.in/84451/dangling-else-problem-and-ambiguity-elimination>

13 votes

-- pC (22k points)

2.6.2 Grammar: GATE1991-10a top

<https://gateoverflow.in/537>



Selected Answer

Yes. It is unambiguous grammar since for any string no more than 1 parse tree is possible.

For precedence draw the parse tree and find the depth of operator `-` and `/` .

Here `/` having more depth than `-` operator so precedence of `/` is higher than `-`.

12 votes

-- papesh (25.7k points)

2.6.3 Grammar: GATE1991-10b top

<https://gateoverflow.in/43604>

If the expression with redundant parentheses is (id/id) or $id - (id/id)$ then it can be generated by the given grammar.

To generate expression (id/id) we can go through following steps-

1. $E \rightarrow T$
 2. $E \rightarrow F$
 3. $E \rightarrow (E)$
 4. $E \rightarrow (T)$
 5. $E \rightarrow (T/F)$
 6. $E \rightarrow (F/F)$
 7. $E \rightarrow (id/id)$
- $(T \rightarrow F)$
 $(F \rightarrow (E))$
- $(E \rightarrow T)$
 $(T \rightarrow T/F)$
- $(T \rightarrow F)$
 $(F - > id)$

Now to generate expression $id - (id/id)$ we can go through following steps-

1. $E \rightarrow E - T$
 2. $E \rightarrow E - F$
- $(T \rightarrow F)$

- | | |
|---------------------------------|-----------------------|
| 3. $E \rightarrow E - (E)$ | $(F \rightarrow (E))$ |
| 4. $E \rightarrow E - (T)$ | $(E \rightarrow T)$ |
| 5. $E \rightarrow E - (T/F)$ | $(T \rightarrow T/F)$ |
| 6. $E \rightarrow E - (F/F)$ | $(T \rightarrow F)$ |
| 7. $E \rightarrow T - (F/F)$ | $(E \rightarrow T)$ |
| 8. $E \rightarrow F - (F/F)$ | $(T \rightarrow F)$ |
| 9. $E \rightarrow id - (id/id)$ | $(F \rightarrow id)$ |

Up 6 votes

-- Dhananjay Kumar Sharma (25.7k points)

2.6.4 Grammar: GATE1991-10c top

<https://gateoverflow.in/43605>



Selected Answer

Here we have to convert the grammar into one which does not generate expressions with redundant parentheses. So the grammar which does not generate expressions with redundant parentheses

$$E \rightarrow E - T \mid T$$

$$E' \rightarrow E - T$$

$$T \rightarrow id \mid T'/F$$

$$T' \rightarrow F \mid T'/F$$

$$F \rightarrow id \mid (E')$$

Now equivalent grammar after removing left recursion

$$E \rightarrow TX'$$

$$X' \rightarrow -TX' \mid \epsilon$$

$$E' \rightarrow E - T$$

$$T \rightarrow id \mid T'/F$$

$$T' \rightarrow FY'$$

$$Y' \rightarrow /FY' \mid \epsilon$$

$$F \rightarrow id \mid (E')$$

Up 14 votes

-- Manoj Kumar (38.6k points)

2.6.5 Grammar: GATE1994-1.18 top

<https://gateoverflow.in/2461>



Selected Answer

It will be C.

Since CFG's are used to show syntactic rules while designing compiler, and syntactic rules don't check for meaningful things such as if a variable has been declared before its use or not. Such things are meant to be handled by Semantic Analysis phase (requires power of a context sensitive grammar).

For D, a regular expression does not restrict the string length. Languages have restriction for variable name length for storing purpose like in symbol table.

For A, if then else is inherently ambiguous. But CFG can represent inherently ambiguous languages just that there are more than one parse trees possible for some strings.

36 votes

-- Gate Keeda (19,6k points)

2.6.6 Grammar: GATE1994-20 [top](#)

<https://gateoverflow.in/2516>



Selected Answer

Its answer is $2n - 1$ for n length string, because in CNF at every step only 1 terminal can replace a variable, for example

$S - AB$

$A - a$

$B - c$

For generating string 'ac' 3 productions will be used.

Reference :- Peter Linz

24 votes

-- Manu Thakur (39,6k points)

2.6.7 Grammar: GATE1994-3.5 [top](#)

<https://gateoverflow.in/2482>



Selected Answer

Backus normal form (BNF) is a notation technique for context-free grammars, often used to describe the syntax of languages used in computing

Yacc (Yet Another Compiler-Compiler) is a computer program for the Unix operating system. It is a Look Ahead Left-to-Right (LALR) parser generator, generating a parser, the part of a compiler that tries to make syntactic sense of the source code, specifically a LALR parser, based on an analytic grammar

Yacc is written in portable C. The class of specifications accepted is a very general one: LALR(1) grammars with disambiguating rules.

8 votes

-- set2018 (7,6k points)

2.6.8 Grammar: GATE1995-1.10 [top](#)

<https://gateoverflow.in/2597>



Selected Answer

- $S\alpha \rightarrow$

This violates the condition of context-free grammar that the **LHS** must be a single non-terminal symbol.

- $S\alpha \rightarrow b$

This violates even the weaker requirement for CSG that the length of **RHS** of a production must

be at least same as that of LHS. So, the grammar is not even context-sensitive.

Ref: <https://stackoverflow.com/questions/8236422/context-free-grammars-versus-context-sensitive-grammars>

25 votes

-- Arjun Suresh (350k points)

2.6.9 Grammar: GATE1995-9 [top](#)

<https://gateoverflow.in/2644>

A grammer having left recursion generates a cycle.

And no left recursive grammer is LL(1) grammar.

11 votes

-- simran (103 points)

2.6.10 Grammar: GATE1996-11 [top](#)

<https://gateoverflow.in/2763>



Selected Answer

Final grammar is

- $S \rightarrow ABAC \mid ABC \mid BAC \mid BC \mid AC \mid AAC \mid d$
- $A \rightarrow aA \mid a$
- $B \rightarrow bB \mid b$
- $C \rightarrow d$

27 votes

-- Mithlesh Upadhyay (5.8k points)

2.6.11 Grammar: GATE1996-2.10 [top](#)

<https://gateoverflow.in/2739>



Selected Answer

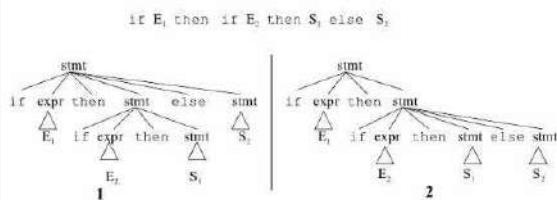
(d) the sentence

- if a then if b then c:= d else c:= f has two parse trees as follows:

 1. if a then (if b then c:= d) else c:= f
 2. and if a then (if b then c:=d else c:= f)

Ambiguity – “dangling else”

stmt \rightarrow if expr then stmt |
if expr then stmt else stmt | other stmts



22 votes

-- Arjun Suresh (350k points)

2.6.12 Grammar: GATE1997-1.6 [top](#)

<https://gateoverflow.in/2222>



Selected Answer

It will be A. For multiple ' \oplus ', the derivation is possible only via ' X ' which is on left side of ' \oplus ' in the production. Hence it is left associative.

For multiple ' $*$ ', the derivation is possible only via ' Y ' which is on the right side of ' $*$ ' in the production. Hence it is right associative.

If both left and right derivations were possible, the grammar would have been ambiguous and we couldn't have given associativity.

27 votes

-- Gate Keeda (19.6k points)

2.6.13 Grammar: GATE1997-11 [top](#)

<https://gateoverflow.in/2271>



Selected Answer

a. $i + k = j + m$
where $i, j, k, m \geq 0$

b. bcde

14 votes

-- Danish (3.8k points)

2.6.14 Grammar: GATE1998-14 [top](#)

<https://gateoverflow.in/1728>



Selected Answer

(a) $L(G_1) = \{a^n b^m \mid n \neq m\}$

(b)

- $S \rightarrow S_1 S_2 \mid S_2 S_1 \quad (2)$
- $S_2 \rightarrow a S_2 b \mid \epsilon \quad (2)$
- $S_1 \rightarrow \epsilon \quad (1)$

So, totally $2 + 2 + 1 = 5$ extra productions.

(c) If grammar is ambiguous and no unambiguous grammar is possible for the language, then language is inherently ambiguous.

Non-determinism of language (if it means, L is not DCFL) + ambiguity for all possible grammars implies inherent ambiguity. Or if a language is deterministic, it is surely not inherently ambiguous. But if a language is not deterministic, it may or may not be inherently ambiguous.

The given language L_2 is the set of strings where number of a 's is followed by an equal number of b 's and if not, for the remaining part, the number of a 's is followed by an equal number of b 's. This is actually a deterministic language as we do not need to do multiple counts here. Only after the first condition is violated we need to start doing the count for the second part. This makes the language deterministic and hence it cannot be inherently ambiguous.

2 votes

-- Praveen Saini (54.6k points)

2.6.15 Grammar: GATE1998-6b [top](#)

<https://gateoverflow.in/1697>



(b) As it is the case of indirect recursion so let first make it as direct recursion then apply rules of removal of left recursion.

to make it as direct recursion first production remain unchanged while in second production substitute the right hand side of first production wherever it comes. In the question S comes in middle of A so substitute the right hand side of production S . Now after substituting it looks like..

$$A \rightarrow Ac \mid Aad \mid bd \mid \epsilon$$

Now remove direct recursion from it

For removal of direct recursion rule--

$$A \rightarrow A\alpha_1 \mid \dots \mid A\alpha_n \mid \beta_1 \mid \dots \mid \beta_m$$

Replace these with two sets of productions, one set for A :

$$A \rightarrow \beta_1 A' \mid \dots \mid \beta_m A'$$

and another set for the fresh nonterminal A'

$$A' \rightarrow \alpha_1 A' \mid \dots \mid \alpha_n A' \mid \epsilon$$

After applying these rule we'll get..

$$A \rightarrow bdA' \mid A'$$

$$A' \rightarrow cA' \mid adA' \mid \epsilon$$

Now complete production without left recursion is...

$$S \rightarrow Aa \mid b$$

$$A \rightarrow bdA' \mid A'$$

$$A' \rightarrow cA' \mid adA' \mid \epsilon$$

15 votes

-- shashi shekhar (569 points)

2.6.16 Grammar: GATE1999-2.15 top

<https://gateoverflow.in/1493>



Let grammar is like this :

$$S \rightarrow a$$

$$A \rightarrow AbA$$

This grammar is left as well as right recursive but still unambiguous.. A is useless production but still part of grammar..

A grammar having both left as well as right recursion may or may not be ambiguous ..

Let's take a grammar say

$$S \rightarrow SS$$

Now, according to the link https://en.wikipedia.org/wiki/Formal_grammar

For a grammar G, if we have $L(G)$ then all the strings/sentences in this language can be produced after some finite number of steps .

But, for the grammar

$$S \rightarrow SS$$

Can we produce any string after a finite number of steps ? NO, and hence language of this grammar is empty set {} .

Hence, If a grammar is having both left and right recursion, then grammar may or may not be ambiguous .

29 votes

-- Digvijay (54.9k points)

2.6.17 Grammar: GATE2000-2.21, ISRO2015-24 top

<https://gateoverflow.in/668>



Selected Answer

I guess its B.

Operator which is at lower level in the grammar is termed to have higher precedence.

27 votes

-- Gate Keeda (19.6k points)

2.6.18 Grammar: GATE2001-1.18 top

<https://gateoverflow.in/711>



Selected Answer

(A) is the answer.

(A) We can not have different Left Most Derivation and Right Most Derivation parse trees BUT we can certainly have different LMD and RMD for a given string.(LMD and RMD here refers to the order of various productions used for derivation which could be different.)

(D) is wrong w.r.t. question because IT IS TRUE that any LR(k) IS NEVER AMBIGUOUS and so an ambiguous can never be an LR(K) for any k, no matter how large k becomes.

(B) and (C) can not be the answer because LL(1) is TOP-DOWN parser and LALR is powerful than SLR. So both are TRUE.

31 votes

-- Sandeep_Uniyal (7.5k points)

2.6.19 Grammar: GATE2001-18 top

<https://gateoverflow.in/759>



Selected Answer

a. $S \rightarrow aS' \mid bS'$
 $S' \rightarrow aS' \mid bS' \mid \epsilon$

b. $S \rightarrow aSbS \rightarrow abS \rightarrow abaSbS \rightarrow ababS \rightarrow abab$
 $S \rightarrow aSbS \rightarrow abSaSbS \rightarrow abaSbS \rightarrow ababS \rightarrow abab$

The above two derivations corresponds to two different parse trees and hence the grammar is ambiguous.

23 votes

-- Pooja Palod (31.3k points)

2.6.20 Grammar: GATE2003-56 top

<https://gateoverflow.in/944>



Selected Answer

- $\text{FIRST}(S) = \{i, a\}$
- $\text{FIRST}(S') = \{e, \epsilon\}$
- $\text{FIRST}(E) = \{b\}$
- $\text{FOLLOW}(S') = \{e, \$\}$

Only when **FIRST** contains ϵ , we need to consider **FOLLOW** for getting the parsing table entry.

$$M[S', e] = \{S' \rightarrow eS(\text{FIRST}), S' \rightarrow \epsilon \text{ (considering FOLLOW)}\}$$

$$M[S', \$] = \{S \rightarrow \epsilon\}$$

	a	i	b	e	t	\$
S	S $\rightarrow a$	S $\rightarrow ietSS'$				
S'				$S' \rightarrow eS$ $S' \rightarrow \epsilon$		$S' \rightarrow \epsilon$
E			E $\rightarrow b$			

Answer is D

32 votes

-- Pooja Palod (31.3k points)

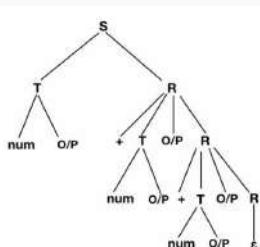
2.6.21 Grammar: GATE2003-58 top

<https://gateoverflow.in/946>



Selected Answer

Answer = option B
9 5 + 2 +



33 votes

-- Amar Vashishth (30.5k points)

2.6.22 Grammar: GATE2004-45 top

<https://gateoverflow.in/1042>



Selected Answer

Here # is multiplication and & is addition by semantics rules given in the question.
By observation of productions,

1. here & (+) is higher precedence than #(*), because & is far from starting symbol
2. both &, # are left associative

So, we can solve the expression as $((2 * (3 + 5)) * (6 + 4)) = 160$

Answer is (C).

32 votes

-- sonam vyas (14.8k points)

2.6.23 Grammar: GATE2004-8 top

<https://gateoverflow.in/1005>



Selected Answer

answer is B.

Operator grammar cannot contain

1. Nullable variable
2. Two adjacent non-terminal on RHS of production

22 votes

-- koushiksng264 (233 points)

2.6.24 Grammar: GATE2004-88 top

<https://gateoverflow.in/1082>



Selected Answer

above CFG generate string b, aaa..

b will eliminate options A and D

aaa eliminate options B.

C is answer i.e. number of a = $3k, k = 0, 1, 2 \dots$

23 votes

-- Digvijay (54.9k points)

2.6.25 Grammar: GATE2005-59 top

<https://gateoverflow.in/1382>



Selected Answer

n+n*n

E+n*n

E*n

E

String in **RED** indicates handle here

So, answer is D

1 23 votes

-- Pooja Palod (31.3k points)

2.6.26 Grammar: GATE2006-32, ISRO2016-35 [top](#)

<https://gateoverflow.in/995>



Selected Answer

- I. **True.** G is ambiguous. E.g. the string ab has multiple derivation trees like $S \rightarrow SS \rightarrow abS \rightarrow ab$, and $S \rightarrow ab$.
- II. **False.** G does not produce all strings with equal no. of a 's and b 's. ($aabb$ cannot be generated).
- III. **True.** The given grammar G generates the language $(ab + ba)^*$, which is Regular and therefore also DCFL. So, a D-PDA can be designed for G .

Hence, the answer is option **B**.

1 70 votes

-- Pooja Palod (31.3k points)

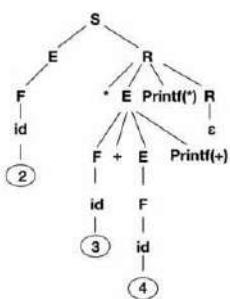
2.6.27 Grammar: GATE2006-59 [top](#)

<https://gateoverflow.in/1837>



Selected Answer

It will be **D**. Make a tree and perform post order evaluation.



1 26 votes

-- Gate Keeda (19.6k points)

2.6.28 Grammar: GATE2006-84 [top](#)

<https://gateoverflow.in/1856>



Selected Answer

option A

$C \Rightarrow a$

or, $C \Rightarrow b$

or, $C \Rightarrow aCb \Rightarrow aaCbb \Rightarrow aaaCbbb \dots$ soon

at last you have to put either $C \rightarrow a$ or $C \rightarrow b$

so production C is used to derive $a^{n+1}b^n$ or $a^n b^{n+1}$ $n \geq 0$

$S \rightarrow AC$ [$Aa^n b^{n+1}$] can make $a^{n+1}b^{n+1}$ as single a can be derived from A [$A \Rightarrow aA \Rightarrow a$ as $A \rightarrow \epsilon$] , similarly $S \rightarrow CB$

simple way, ab can be derived from grammar as $S \Rightarrow AC \Rightarrow aAC \Rightarrow aC \Rightarrow ab$

option A is wrong

option B , language is used to drive a^+b^* or a^*b^+ , ab will be derived as $S \Rightarrow aS \Rightarrow ab$

option B is wrong

Option C

$C \Rightarrow \epsilon$

or $C \Rightarrow aCb \Rightarrow aaCbb \Rightarrow aaaCbbb \dots$ soon at last need to put $C \rightarrow \epsilon$

Production C will generate $a^n b^n$ $n \geq 0$

$S \rightarrow AC$ can generate $a^n b^n$ as A can be ϵ , similarly $S \rightarrow CB$

option C is wrong

Option D .

production C is used for generate $a^n b^n$ as in option C

$S \rightarrow AC$ will increase no of a's before $a^n b^n$

as A will generate a or aa or $aaa\dots$ i.e a^+ , so $S \rightarrow AC$ will generate $a^+ a^n b^n$, i.e $a^i b^j$ $i > j$

$S \rightarrow CB$ will generate $a^n b^n b^+$ i.e $a^i b^j$ $i < j$

option D is right .

Upvote 30 votes

-- Praveen Saini (54.6k points)

2.6.29 Grammar: GATE2006-85 top

<https://gateoverflow.in/79801>



Selected Answer

$L = a^l b^m$ $l \neq m$ means either $l > m$ or $l < m$

Case I [$l > m$]

if $l > m$, $a^l b^m$ can be written as $a^{l-m} a^m b^m$ [$l - m$ cannot be 0 as l should be $> m$]

$S \rightarrow AC$, one step

a^{l-m} use $l - m$ steps using productions of A

[as $l - m = 1$, one step $A \rightarrow a$

$l - m = 2$, two steps $A \rightarrow aA \rightarrow aa$

$l - m = 3$, three steps, $A \rightarrow aA \rightarrow aaA \rightarrow aaa..$ so on]

$a^m b^m$ will be generated in $m + 1$ steps using production C

[as $m = 0$ one step $C \rightarrow \epsilon$

$m = 1$, two steps $C \rightarrow aCb \rightarrow ab$

$m = 2$, three steps $C \rightarrow aCb \rightarrow aaCbb \rightarrow aabb..$ so on]

so if $l > m$ total steps $= 1 + l - m + m + 1 = l + 2$

Case II [$l < m$]

similar if $l < m$, $a^l b^m$ can be written as $a^l b^l b^{m-l}$ [$m - l$ cannot be 0 as m should be $> l$]

$S \rightarrow CB$ one step

$a^l b^l$ will be derived using $l + 1$ steps

b^{m-l} will be derived using $m - l$ steps

total $= 1 + l + 1 + m - l = m + 2$

so $L = a^l b^m$ $l \neq m$ will take $\max(l, m) + 2$ steps

1 votes

-- Praveen Saini (54.6k points)

Option A is correct. We can try some strings:

$w = a^l b^m, l \neq m \Rightarrow aabbS \rightarrow \underset{1}{CB} \rightarrow \underset{2}{aCbB} \rightarrow \underset{3}{aaCbbB} \rightarrow \underset{4}{aaCbbB} \rightarrow \underset{5}{aabbb}$

$w = aaaabbS \rightarrow \underset{1}{AC} \rightarrow \underset{2}{AaCb} \rightarrow \underset{3}{AaaCbb} \rightarrow \underset{4}{Aaa\epsilon bb} \rightarrow \underset{5}{aAaabb} \rightarrow \underset{6}{aaaabb}$

$w = aaaab$

$S \rightarrow \underset{1}{AC} \rightarrow \underset{2}{AaCb} \rightarrow \underset{3}{Aa\epsilon b} \rightarrow \underset{4}{aAab} \rightarrow \underset{5}{aaAab} \rightarrow \underset{6}{aaaab}$

In cases 2 and 3 reducing number of $b's$ has no affect on the number of productions required.

13 votes

-- rameshbabu (3.5k points)

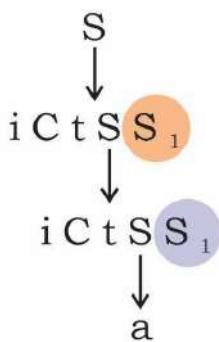
2.6.30 Grammar: GATE2007-52 top

<https://gateoverflow.in/1250>



Selected Answer

here, we can expand any one of S_1 to ϵ and other to ea , but which one will it be need not matter, coz in the end we will still get the same string.



this means that the Grammar is Ambiguous. LL(1) cannot be ambiguous. [Here's a Proof for that.](#)

LL(1) Derivations

Left to Right Scan of input

Leftmost Derivation

(1) look ahead 1 token at each step

Alternative characterization of LL(1) Grammars:

Whenever $A \rightarrow \alpha \mid \beta \in G$

1. $FIRST(\alpha) \cap FIRST(\beta) = \{ \}$, and

2. if $\alpha \xrightarrow{*} \epsilon$ then $FIRST(\beta) \cap FOLLOW(A) = \{ \}$.

Corollary: No Ambiguous Grammar is LL(1).

answer = **option C**

1 34 votes

-- Amar Vashishth (30.5k points)

2.6.31 Grammar: GATE2007-53 top

<https://gateoverflow.in/1251>



Selected Answer

answer = **option C**

LL Grammar

Grammars which can be parsed by an LL parser.

LL parser

Parses the input from Left to right, and constructs a Leftmost derivation of the sentence(i.e. it is always the leftmost non-terminal which is rewritten). LL parser is a top-down parser for a subset of context-free languages.

An LL parser is called an LL(k) parser if it uses k tokens of lookahead when parsing a sentence and can do it without backtracking.

Consider a Grammar

G :

$S \rightarrow a|aa$

this grammar is **Regular** but cannot be parsed by a LL(1) parser w/o backtracking, coz here, lookahead is of 1 symbol only and in the grammar for both productions, parser while looking at just one(first) symbol, which is a , fails to select the correct rule for parsing.

Hence, not every Regular grammar is LL(1); Statement P is False.

LR Grammar

Grammars which can be parsed by LR parsers.

LR Parsers

they are a type of bottom-up parsers that efficiently handle deterministic context-free languages(DCFL) in guaranteed linear time.

all Regular Languages are also DCFL hence, They all can be parsed by a LR(1) grammar.

Hence, Statement Q is True.

28 votes

-- Amar Vashishth (30.5k points)

2.6.32 Grammar: GATE2007-78 top

<https://gateoverflow.in/1272>



Selected Answer

$$\begin{aligned} S &\rightarrow aB \\ &\rightarrow aaBB \\ &\rightarrow aabB \\ &\rightarrow aabbS \\ &\rightarrow aabbaB \\ &\rightarrow aabbab \end{aligned}$$

22 votes

-- Arjun Suresh (350k points)

2.6.33 Grammar: GATE2007-79 top

<https://gateoverflow.in/43512>



Selected Answer

$$\begin{aligned} S &\rightarrow aB \\ &\rightarrow aaBB \\ &\rightarrow aabB \\ &\rightarrow aabbS \\ &\rightarrow aabbaB \\ &\rightarrow aabbab \end{aligned}$$

$$\begin{aligned} S &\rightarrow aB \\ &\rightarrow aaBB \text{ (till now, only 1 choice possible)} \\ &\rightarrow aabSB \text{ (last time we took } B \rightarrow b, \text{ now taking } B \rightarrow bS) \\ &\rightarrow aabbAB \\ &\rightarrow aabbaB \\ &\rightarrow aabbab \end{aligned}$$

So, totally 2 possible derivation trees.

22 votes

-- Arjun Suresh (350k points)

2.6.34 Grammar: GATE2007-IT-9 top

<https://gateoverflow.in/3442>



Selected Answer

(c) $L(D) = L(G)$. **Both must represent same language.** Also, if we are converting a grammar from ambiguous to unambiguous form, we must ensure that our new grammar represents the same language as previous grammar.

For ex $G1 : S \rightarrow Sa \mid aS \mid a$; {Ambiguous (2 parse trees for string 'aa')}

$G1' : S \rightarrow aS \mid a$; {Unambiguous}

Both represent the language represented by the regular expression: a^+

22 votes

-- Anurag Semwal (8k points)

2.6.35 Grammar: GATE2008-50 top

<https://gateoverflow.in/395>



Answer is C:

Statement 1 is **true**: Using GNF we can convert Left recursive grammar to right recursive and by using reversal of CFG and GNF we can convert right recursive to left recursive.

Statement 2 is **false**: because if ϵ is in the language then we can't remove ϵ production from Start symbol. (For example $L = a^*$)

Statement 3 is **true** because right linear grammar generates regular set

Statement 4 is **true**, only two non-terminals are there in each production in CNF. So it always form a binary tree.

39 votes

-- Vikrant Singh (13.5k points)

2.6.36 Grammar: GATE2010-38 top

<https://gateoverflow.in/2339>



It will be **C**.

For LL(1) take First(S). and do intersection between the result. if intersection is Phi then LL(1) else not.

Making a parsing table and checking if there are two or more entries under any terminal. If yes then neither LL(1) nor LR(1).

28 votes

-- Gate Keeda (19.6k points)

2.6.37 Grammar: GATE2014-2-17 top

<https://gateoverflow.in/1973>



$P \rightarrow Q + P$ here P is to right of $+$

So, $+$ is right associative

Similarly, for $T \rightarrow T * U$ $*$ is left associative as T is to left of $*$

So, answer is **B**

23 votes

-- Pooja Palod (31.3k points)

2.6.38 Grammar: GATE2016-2-45 top

<https://gateoverflow.in/39594>



Option (A) has immediate left recursion. " $A \rightarrow Aa$ "

Option (C) has indirect left recursion " $S \rightarrow Aa \xrightarrow{A \rightarrow Sc} Sca$ "

Option (D) has indirect left recursion " $A \rightarrow Bd \xrightarrow{B \rightarrow Ae} Aed$ "

Option (B) is free from left recursion. No direct left recursion. No indirect left recursion.

option **B** is correct

35 votes

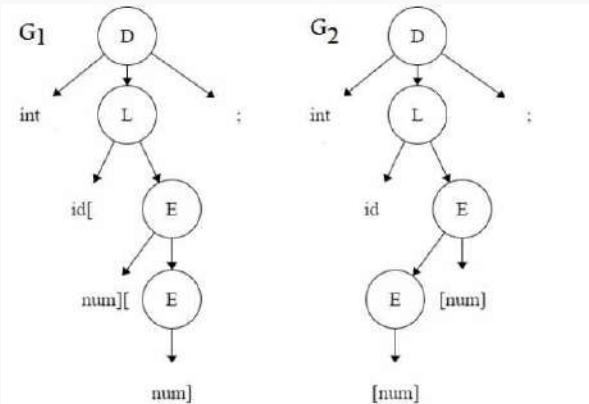
-- Ashish Deshmukh (1.5k points)

2.6.39 Grammar: GATE2016-2-46 top

<https://gateoverflow.in/39598>



Option **A** Both G1 and G2



35 votes

-- Shashank Chavan (3.3k points)

2.7

Intermediate Code(8) top

<https://gateoverflow.in/87885>

Is the following code template for the if-then-else statement correct? if not, correct it.

if expression then statement 1

else statement 2

Template:

Code for expression

(*result in $E, E > O$ indicates true *)

Branch on $E > O$ to $L1$

Code for statement 1

L1: Code for statement 2

descriptive | gate1989 | compiler-design | intermediate-code

Answer

2.7.2 Intermediate Code: GATE1992-11a [top](#)

<https://gateoverflow.in/590>

Write syntax directed definitions (semantic rules) for the following grammar to add the type of each identifier to its entry in the symbol table during semantic analysis. Rewriting the grammar is not permitted and semantic rules are to be added to the ends of productions only.

- $D \rightarrow TL;$
- $T \rightarrow \text{int}$
- $T \rightarrow \text{real}$
- $L \rightarrow L, id$
- $L \rightarrow id$

gate1992 | compiler-design | syntax-directed-translation | intermediate-code | normal

Answer

2.7.3 Intermediate Code: GATE1992-11b [top](#)

<https://gateoverflow.in/43583>

Write 3 address intermediate code (quadruples) for the following boolean expression in the sequence as it would be generated by a compiler. Partial evaluation of boolean expressions is not permitted. Assume the usual rules of precedence of the operators.

$$(a + b) > (c + d) \text{ or } a > c \text{ and } b < d$$

gate1992 | compiler-design | syntax-directed-translation | intermediate-code | descriptive

Answer

2.7.4 Intermediate Code: GATE1994-1.12 [top](#)

<https://gateoverflow.in/2453>

Generation of intermediate code based on an abstract machine model is useful in compilers because

- A. it makes implementation of lexical analysis and syntax analysis easier
- B. syntax-directed translations can be written for intermediate code generation
- C. it enhances the portability of the front end of the compiler
- D. it is not possible to generate code for real machines directly from high level language programs

gate1994 | compiler-design | intermediate-code | easy

Answer

2.7.5 Intermediate Code: GATE2014-2-34 [top](#)

<https://gateoverflow.in/1993>

For a C program accessing $X[i][j][k]$, the following intermediate code is generated by a compiler. Assume that the size of an **integer** is 32 bits and the size of a **character** is 8 bits.

```
t0 = i * 1024
t1 = j * 32
```

```
t2 = k * 4
t3 = t1 + t0
t4 = t3 + t2
t5 = X[t4]
```

Which one of the following statements about the source code for the C program is CORRECT?

- A. **X** is declared as "int **X[32][32][8]**".
- B. **X** is declared as "int **X[4][1024][32]**".
- C. **X** is declared as "char **X[4][32][8]**".
- D. **X** is declared as "char **X[32][16][2]**".

[gate2014-2](#) [compiler-design](#) [intermediate-code](#) [programming-in-c](#) [normal](#)

[Answer](#)

2.7.6 Intermediate Code: GATE2014-3-17 [top](#)

<https://gateoverflow.in/2051>

One of the purposes of using intermediate code in compilers is to

- A. make parsing and semantic analysis simpler.
- B. improve error recovery and error reporting.
- C. increase the chances of reusing the machine-independent code optimizer in other compilers.
- D. improve the register allocation.

[gate2014-3](#) [compiler-design](#) [intermediate-code](#) [easy](#)

[Answer](#)

2.7.7 Intermediate Code: GATE2015-1-55 [top](#)

<https://gateoverflow.in/8365>

The least number of temporary variables required to create a three-address code in static single assignment form for the expression $q + r/3 + s - t * 5 + u * v/w$ is_____.

[gate2015-1](#) [compiler-design](#) [intermediate-code](#) [normal](#) [numerical-answers](#)

[Answer](#)

2.7.8 Intermediate Code: GATE2015-1-8 [top](#)

<https://gateoverflow.in/8096>

For computer based on three-address instruction formats, each address field can be used to specify which of the following:

- (S1) A memory operand
 - (S2) A processor register
 - (S3) An implied accumulator register
- A. Either S_1 or S_2
 - B. Either S_2 or S_3
 - C. Only S_2 and S_3
 - D. All of S_1 , S_2 and S_3

[gate2015-1](#) [compiler-design](#) [intermediate-code](#) [normal](#)

[Answer](#)

Answers: Intermediate Code

2.7.1 Intermediate Code: GATE1989-4-v top

<https://gateoverflow.in/87885>



Selected Answer

The given template is wrong. The following should be correct:

Code for Expression

Branch on $E > 0$ to L

Code for segment 2

Branch to L_1

L : Code for statement 1

L_1 :

👍 4 votes

-- Pinaki Dash (1.6k points)

2.7.2 Intermediate Code: GATE1992-11a top

<https://gateoverflow.in/590>



Selected Answer

PRODUCTION RULE

$D \rightarrow TL$

$T \rightarrow int$

$T \rightarrow real$

$L \rightarrow L, id$

$L \rightarrow id$

SEMANTIC ACTIONS

$L.in$	
$\quad := T$	
$\quad .type$	
$\quad T.type$	
$\quad := integer$	
$\quad T.type$	
$\quad := real$	
$\quad L1.in$	
$\quad = L.in$	
$\quad Enter_type(id$	
$\quad \quad .entry,$	
$\quad \quad L.in)$	
$\quad Enter_type(id$	
$\quad \quad .entry,$	
$\quad \quad L.in)$	

👍 7 votes

-- Anupoju Satish Kumar (42.3 points)

2.7.3 Intermediate Code: GATE1992-11b top

<https://gateoverflow.in/43583>



Selected Answer

Each instruction in quadruples presentation is divided into four fields: operator, arg1, arg2, and result. The above example is represented below in quadruples format:

$(a + b) > (c + d)$ OR $a > c$ AND $b < d$

$(t1 > t2)$ OR $a > c$ AND $b < d$

$t3 \text{ OR } t4 \text{ AND } t5$

$t3 \text{ OR } t6$

$t1 = a + b$

$t2 = c + d$

$t3 = t1 > t2$

$t4 = a > c$

$t5 = b < d$

$t6 = t4 \text{ AND } t5$

$t7 = t3 \text{ OR } t6$

Op	arg1	arg2	Result
+	a	b	$t1$
+	c	d	$t2$
$>$	$t1$	$t2$	$t3$
$.>$	a	c	$t4$
$<$	b	d	$t5$
AND	$t4$	$t5$	$t6$
OR	$t3$	$t6$	$t7$

👍 25 votes

-- shekhar chauhan (45.2k points)

2.7.4 Intermediate Code: GATE1994-1.12 top

<https://gateoverflow.in/2453>



Selected Answer

C. stating the actual use of the Intermediate Code.

Also optimizations can be done on intermediate code enhancing the portability of the optimizer.

👍 24 votes

-- Gate Keeda (19.6k points)

2.7.5 Intermediate Code: GATE2014-2-34 top

<https://gateoverflow.in/1993>



Selected Answer

k is multiplied by 4, means sizeof(datatype) is int.

j is multiplied by 32, means the inner most dimension of array is $32/4 = 8$ (we have to divide by the size of the inner dimension- which here is a simple integer)

i is multiplied by 1024, means the second dimension of array is $1024/32 = 32$ ($32 = 8 * 4$ is the size of the inner dimension here)

So, (A) is correct. The first dimension is not needed for code generation and that is why in C language while passing an array to a function, we can omit the value of the first dimension but not any others.

We can also do as follows:

$$X[i][j][k] = *(*(*X + i) + j) + k$$

In Integer arithmetic, this equals

$$*(*(*X + i * \text{sizeof}(*X)) + j * \text{sizeof}(**X) + k * \text{sizeof}(***X))$$

as for every add to a pointer we have to multiply the size of the pointed value (to get a valid address)

So, from the given code we get

$$\text{sizeof}(**X) = 4, - \text{ int}$$

$$\text{sizeof}(**X) = 32 - \text{ int array of size } 8$$

$$\text{sizeof}(*X) = 1024 - 2 \text{ D int array of size [32] havinf size of inner } 1D \text{ array } 32.$$

So, the inner dimensions must be 32 and 8 and type must be integer. So, only option A matches.

48 votes

-- Arjun Suresh (350k points)

2.7.6 Intermediate Code: GATE2014-3-17 top

<https://gateoverflow.in/2051>



C. that is the actual use of intermediate code generator in a compiler.

24 votes

-- Gate Keeda (19.6k points)

2.7.7 Intermediate Code: GATE2015-1-55 top

<https://gateoverflow.in/8365>



Answer is 8.

In [compiler](#) design, **static single assignment form** (often abbreviated as **SSA form** or simply **SSA**) is a property of an [intermediate representation](#) (IR), which requires that each variable is assigned exactly once, and every variable is defined before it is used. Existing variables in the original IR are split into *versions*, new variables.

We will need a temporary variable for storing the result of each binary operation as SSA (Static Single Assignment) implies the variable cannot be repeated on LHS of assignment.

$$q + r/3 + s - t * 5 + u * v/w$$

$$\begin{aligned} t1 &= r/3; \\ t2 &= t * 5; \\ t3 &= u * v; \\ t4 &= t3/w; \\ t5 &= q + t1; \\ t6 &= t5 + s; \\ t7 &= t6 - t2; \\ t8 &= t7 + t4 \end{aligned}$$

<http://web.stanford.edu/class/archive/cs/cs143/cs143.1128/handouts/240%20TAC%20Examples.pdf>

71 votes

-- Arjun Suresh (350k points)

2.7.8 Intermediate Code: GATE2015-1-8 [top](#)

<https://gateoverflow.in/8096>



Selected Answer

Three address Instruction

Computer with three addresses instruction format can use each address field to specify either processor register or memory operand.

e.g., $X = (A + B) * (C + D)$

Equivalent Three address Instructions

ADD R1, A, B	$R1 \leftarrow M[A] + M[B]$
ADD R2, C, D	$R2 \leftarrow M[C] + M[D]$
MUL X, R1, R2	$M[X] \leftarrow R1 * R2$

The advantage of the three address formats is that it results in short program when evaluating arithmetic expression. The disadvantage is that the binary-coded instructions require too many bits to specify three addresses.

30 votes

-- Prasanna Ranganathan (4.7k points)

2.8

Left Recursion(1) [top](#)

2.8.1 Left Recursion: GATE2017-2-32 [top](#)

<https://gateoverflow.in/118374>

Consider the following expression grammar G :

- $E \rightarrow E - T \mid T$
- $T \rightarrow T + F \mid F$
- $F \rightarrow (E) \mid id$

Which of the following grammars is not left recursive, but is equivalent to G ?

A. $E \rightarrow E - T \mid T$

$$T \rightarrow T + F \mid F$$

$$F \rightarrow (E) \mid id$$

B. $E \rightarrow TE'$

$$E' \rightarrow -TE' \mid \epsilon$$

$$T \rightarrow T + F \mid F$$

$$F \rightarrow (E) \mid id$$

C. $E \rightarrow TX$

$$X \rightarrow -TX \mid \epsilon$$

$$T \rightarrow FY$$

$$Y \rightarrow +FY \mid \epsilon$$

$F \rightarrow (E) \mid id$

D. $E \rightarrow TX \mid (TX)$

$X \rightarrow -TX \mid +TX \mid \epsilon$

$T \rightarrow id$

gate2017-2 grammar left-recursion

Answer

Answers: Left Recursion

2.8.1 Left Recursion: GATE2017-2-32 top

<https://gateoverflow.in/118374>



Since, the grammar given in the question is left recursive, we need to remove left recursion ,

If Grammar is of form

$$A \rightarrow A\alpha \mid \beta$$

then after removal of left recursion it should be written as

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' \mid \epsilon$$

Since the grammar is :

$$E \rightarrow E - T \mid T \text{ (Here } \alpha \text{ is } '-T' \text{ and } \beta \text{ is } T\text{)}$$

$$T \rightarrow T + F \mid F \text{ (Here } \alpha \text{ is } '+F' \text{ and } \beta \text{ is } F\text{)}$$

$$F \rightarrow (E) \mid id \text{ (It is not having left recursion)}$$

Rewriting after removing left recursion :

$$E \rightarrow TE'$$

$$E' \rightarrow -TE' \mid \epsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow +FT' \mid \epsilon$$

$$F \rightarrow (E) \mid id$$

Now replace E' with X and T' with Y to match with Option C.

$$E \rightarrow TX$$

$$X \rightarrow -TX \mid \epsilon$$

$$T \rightarrow FY$$

$$Y \rightarrow +FY \mid \epsilon$$

$F \rightarrow (E) \mid id$

Hence C is correct.

22 votes

-- Indranil Maji (619 points)

2.9

Lexical Analysis(6) top

2.9.1 Lexical Analysis: GATE1987-1-xvii top

<https://gateoverflow.in/80364>

Using longer identifiers in a program will necessarily lead to:

- A. Somewhat slower compilation
- B. A program that is easier to understand
- C. An incorrect program
- D. None of the above

gate1987 compiler-design lexical-analysis

Answer

2.9.2 Lexical Analysis: GATE2000-1.18, ISRO2015-25 top

<https://gateoverflow.in/641>

The number of tokens in the following C statement is

```
printf ("i=%d, &i=%x", i, &i);
```

- A. 3
- B. 26
- C. 10
- D. 21

gate2000 compiler-design lexical-analysis easy isro2015

Answer

2.9.3 Lexical Analysis: GATE2010-13 top

<https://gateoverflow.in/2186>

Which data structure in a compiler is used for managing information about variables and their attributes?

- A. Abstract syntax tree
- B. Symbol table
- C. Semantic stack
- D. Parse table

gate2010 compiler-design lexical-analysis easy

Answer

2.9.4 Lexical Analysis: GATE2011-1 top

<https://gateoverflow.in/2103>

In a compiler, keywords of a language are recognized during

- A. parsing of the program
- B. the code generation
- C. the lexical analysis of the program

D. dataflow analysis

gate2011 compiler-design lexical-analysis easy

[Answer](#)

2.9.5 Lexical Analysis: GATE2011-19 [top](#)

<https://gateoverflow.in/2121>

The lexical analysis for a modern computer language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense?

- A. Finite state automata
- B. Deterministic pushdown automata
- C. Non-deterministic pushdown automata
- D. Turing machine

gate2011 compiler-design lexical-analysis easy

[Answer](#)

2.9.6 Lexical Analysis: GATE2018-37 [top](#)

<https://gateoverflow.in/204111>

A lexical analyzer uses the following patterns to recognize three tokens T_1 , T_2 , and T_3 over the alphabet $\{a, b, c\}$.

$$T_1: a?(b \mid c)^*a$$

$$T_2: b?(a \mid c)^*b$$

$$T_3: c?(b \mid a)^*c$$

Note that 'x?' means 0 or 1 occurrence of the symbol x. Note also that the analyzer outputs the token that matches the longest possible prefix.

If the string bbaacabc is processes by the analyzer, which one of the following is the sequence of tokens it outputs?

- A. $T_1 T_2 T_3$
- B. $T_1 T_1 T_3$
- C. $T_2 T_1 T_3$
- D. $T_3 T_3$

gate2018 compiler-design lexical-analysis normal

[Answer](#)

Answers: Lexical Analysis

2.9.1 Lexical Analysis: GATE1987-1-xvii [top](#)

<https://gateoverflow.in/80364>



Selected Answer

Answer: Option A) is Correct because lex will take more time to recognize the longer identifiers.

3 votes

-- sarveswara rao vangala (2.1k points)

2.9.2 Lexical Analysis: GATE2000-1.18, ISRO2015-25 [top](#)



Selected Answer

<https://gateoverflow.in/641>

answer - **C**

Tokens are:

1. printf
2. (
3. "i=%d, &i=%x"
4. ,
5. i
6. ,
7. &
8. i
9.)
10. ;

36 votes

-- Ankit Rokde (9.1k points)

2.9.3 Lexical Analysis: GATE2010-13 [top](#)

<https://gateoverflow.in/2186>

Selected Answer

(B) **Symbol table** is the answer.

It can be implemented by using an array, hash table, tree and even some time with the help of the linked list!

25 votes

-- Dexter (9.6k points)

2.9.4 Lexical Analysis: GATE2011-1 [top](#)

<https://gateoverflow.in/2103>

Selected Answer

Typically, the lexical analysis phase of compilation breaks the input text up into sequences of lexemes that each belongs to some particular token type that's useful in later analysis. Consequently, keywords are usually first recognized during lexical analysis in order to make parsing easier. Since parsers tend to be implemented by writing context-free grammars of tokens rather than of lexemes (that is, the *category* of the lexeme rather than the *contents* of the lexeme), it is significantly easier to build a parser when keywords are marked during lexing. Any identifier is also a token so it is recognized in lexical Analysis .

Hence, option **C** is True.

ref@ <http://stackoverflow.com/questions/5202709/phases-of-a-compiler>

34 votes

-- Mithlesh Upadhyay (5.8k points)

2.9.5 Lexical Analysis: GATE2011-19 [top](#)

<https://gateoverflow.in/2121>

Selected Answer

Answer - A

In compiler lexical analyzer categorizes character sequence into lexemes and produces tokens as output for parser. And tokens are expressed in regular expressions so a simple Finite Automata is sufficient for it.

31 votes

-- Ankit Rokde (9k points)

2.9.6 Lexical Analysis: GATE2018-37 top

<https://gateoverflow.in/204111>



Selected Answer

Option D is the correct answer.

You can think T_3 as $(\epsilon + c)(b + a)^*c$

Given string is *bbaacabc*

The longest matching prefix *bbaac* { from regex T_3 you can easily derive *bbaac* }

Now the remaining *abc* { This can also be derived from T_3 }

Hence T_3T_3 is the answer.

10 votes

-- Ruturaj Mohanty (337 points)

2.10

Linking(3) top

2.10.1 Linking: GATE1991-03,ix top

<https://gateoverflow.in/519>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only

A “link editor” is a program that:

- A. matches the parameters of the macro-definition with locations of the parameters of the macro call
- B. matches external names of one program with their location in other programs
- C. matches the parameters of subroutine definition with the location of parameters of subroutine call.
- D. acts as a link between text editor and the user
- E. acts as a link between compiler and the user program

gate1991 compiler-design normal linking

Answer

2.10.2 Linking: GATE2003-76 top

<https://gateoverflow.in/962>

Which of the following is NOT an advantage of using shared, dynamically linked libraries as opposed to using statistically linked libraries?

- A. Smaller sizes of executable files
- B. Lesser overall page fault rate in the system

- C. Faster program startup
 D. Existing programs need not be re-linked to take advantage of newer versions of libraries

gate2003 compiler-design runtime-environments linking easy

Answer

2.10.3 Linking: GATE2004-9 [top](#)

<https://gateoverflow.in/1006>

Consider a program P that consists of two source modules M_1 and M_2 contained in two different files. If M_1 contains a reference to a function defined in M_2 the reference will be resolved at

- A. Edit time
 B. Compile time
 C. Link time
 D. Load time

gate2004 compiler-design easy linking

Answer

Answers: Linking

2.10.1 Linking: GATE1991-03,ix [top](#)

<https://gateoverflow.in/519>



Link editor or (linker) performs

1. external symbol resolution
2. relocation.

ANS: B

Matches external names of one program with their location in other programs

22 votes

-- pramod (3.3k points)

2.10.2 Linking: GATE2003-76 [top](#)

<https://gateoverflow.in/962>



option C: DLL takes more time in program setup (in loading and linking phase to set up the global offset table and load and link the required libraries)

- Since DLLs are separated from executable, the size of executable becomes smaller. Since DLLs are shared among multiple executables, the total memory usage of the system goes down and hence overall page fault rate decreases.
- Dynamic linking takes place during program runtime. So, if a DLL is replaced to a new version, it will automatically get linked during runtime. There is no explicit relinking required as in the case of static linking. (This works by linking the DLL calls to Global Offset Table and the contents of this table is filled during program run. A simple jump in static linking becomes an indirect jump in dynamic linking).

Refer: Galvin section 8.1.5, Dynamic Linking and Shared Libraries

42 votes

-- GateMaster Prime (1.5k points)

2.10.3 Linking: GATE2004-9 top

<https://gateoverflow.in/1006>



Selected Answer

answer - **C.** Each module is compiled separately and then linked together to make the executable. The below commands shows how to do this for two modules *c1.c* and *c2.c* using *gcc*.

```
gcc -c c1.c -o c1.o
gcc -c c2.c -o c2.o
gcc c1.o c2.o -o C.exe
```

28 votes

-- Ankit Rokde (9k points)

2.11

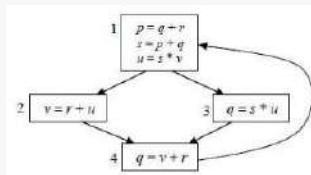
Live Variable(1) top

2.11.1 Live Variable: GATE2015-1-50 top

<https://gateoverflow.in/8356>

A variable x is said to be live at a statement s_i in a program if the following three conditions hold simultaneously:

- There exists a statement S_j that uses x
- There is a path from S_i to S_j in the flow graph corresponding to the program
- The path has no intervening assignment to x including at S_i and S_j



The variables which are live both at the statement in basic block 2 and at the statement in basic block 3 of the above control flow graph are

- A. p, s, u
- B. r, s, u
- C. r, u
- D. q, v

[gate2015-1](#) [compiler-design](#) [live-variable](#) [normal](#)

Answer

Answers: Live Variable

2.11.1 Live Variable: GATE2015-1-50 top

<https://gateoverflow.in/8356>



Selected Answer

r, u.

p, and s are assigned to in 1 and there is no intermediate use of them before that. Hence p, and s are not live in both 2 and 3.

q is assigned to in 4 and hence is not live in both 2 and 3.

v is live at 3 but not at 2.

u is live at 3 and also at 2 if we consider a path of length 0 from 2 – 2.

So, r, u is the answer.

17 votes

-- Arjun Suresh (350k points)

2.12

Macros(3) top

2.12.1 Macros: GATE1995-1.11 top

<https://gateoverflow.in/2598>

What are *x* and *y* in the following macro definition?

```
macro Add x, y
    Load y
    Mul x
    Store y
end macro
```

- A. Variables
- B. Identifiers
- C. Actual parameters
- D. Formal parameters

gate1995 compiler-design macros easy

Answer

2.12.2 Macros: GATE1996-2.16 top

<https://gateoverflow.in/2745>

Which of the following macros can put a macro assembler into an infinite loop?

i.

```
.MACRO M1, X
.IF EQ, X ;if X=0 then
1 X + 1
.ENDC
.IF NE, X ;if X ≠ 0 then
.WORD X ;address (X) is stored here
.ENDC
.ENDM
```

ii.

```
.MACRO M2, X
.IF EQ, X
2 X
.ENDC
.IF NE, X
.WORD X + 1
.ENDC
.ENDM
```

- A. (ii) only
- B. (i) only
- C. both (i) and (ii)

- D. None of the above

[gate1996](#) [compiler-design](#) [macros](#) [normal](#)

[Answer](#)

2.12.3 Macros: GATE1997-1.9 [top](#)

<https://gateoverflow.in/2225>

The conditional expansion facility of macro processor is provided to

- A. test a condition during the execution of the expanded program
- B. to expand certain model statements depending upon the value of a condition during the execution of the expanded program
- C. to implement recursion
- D. to expand certain model statements depending upon the value of a condition during the process of macro expansion

[gate1997](#) [compiler-design](#) [macros](#) [easy](#)

[Answer](#)

Answers: Macros

2.12.1 Macros: GATE1995-1.11 [top](#)

<https://gateoverflow.in/2598>

ans is D

12 votes

-- jayendra (8.2k points)

2.12.2 Macros: GATE1996-2.16 [top](#)

<https://gateoverflow.in/2745>



if $M2$ macro is called with $x = 0$, then it'll go into an infinite loop.
Hence correct option would be A.

6 votes

-- suraj (5.6k points)

2.12.3 Macros: GATE1997-1.9 [top](#)

<https://gateoverflow.in/2225>



Macro is expanded during the process of macro expansion. Hence, correct answer would be (d).

14 votes

-- suraj (5.6k points)

2.13

Parameter Passing(14) [top](#)

2.13.1 Parameter Passing: CMI2010-A-08 [top](#)

<https://gateoverflow.in/46140>

In programming language terminology, **call by value** refers to the fact that:

- A. A function call can return a value.
- B. When a function is called, arguments are copied into local storage.
- C. Functions can indirectly modify the value of external variables.
- D. Every argument passed to a function must have a value.

[cmi2010](#) [compiler-design](#) [runtime-environments](#) [parameter-passing](#)

Answer

2.13.2 Parameter Passing: GATE1988-2xv [top](#)

<https://gateoverflow.in/94333>

What is printed by following program, assuming call-by reference method of passing parameters for all variables in the parameter list of procedure P?

```
program      Main(inout, output);
var         a, b:integer;
procedure P(x, y, z:integer);
begin
    y:=y+1
    z:=x+x
end P;
begin
    a:=2; b:=3;
    p(a+b, a, a);
    Write(a)
end.
```

[gate1988](#) [descriptive](#) [compiler-design](#) [runtime-environments](#) [parameter-passing](#) [numerical-answers](#)

Answer

2.13.3 Parameter Passing: GATE1989-3-viii [top](#)

<https://gateoverflow.in/37264>

In which of the following case(s) is it possible to obtain different results for call-by-reference and call-by-name parameter passing?

- A. Passing an expression as a parameter
- B. Passing an array as a parameter
- C. Passing a pointer as a parameter
- D. Passing an array element as a parameter

[gate1989](#) [parameter-passing](#) [runtime-environments](#) [compiler-design](#)

Answer

2.13.4 Parameter Passing: GATE1991-03,X [top](#)

<https://gateoverflow.in/524>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Indicate all the true statements from the following:

- A. Recursive descent parsing cannot be used for grammar with left recursion.
- B. The intermediate form for representing expressions which is best suited for code optimization is the postfix form.
- C. A programming language not supporting either recursion or pointer type does not need the support of dynamic memory allocation.
- D. Although C does not support call-by-name parameter passing, the effect can be correctly simulated in C
- E. No feature of Pascal typing violates strong typing in Pascal.

gate1991 | compiler-design | parameter-passing | programming | difficult

Answer

2.13.5 Parameter Passing: GATE1991-09a [top](#)

<https://gateoverflow.in/536>

Consider the following pseudo-code (all data items are of type integer):

```
procedure P(a, b, c);
    a := 2;
    c := a + b;
end {P}

begin
    x := 1;
    y := 5;
    z := 100;
    P(x, x*y, z);
    Write ('x = ', x, 'z = ', z);
end
```

Determine its output, if the parameters are passed to the Procedure P by

- i. value
- ii. reference
- iii. name

gate1991 | compiler-design | parameter-passing | normal | runtime-environments

Answer

2.13.6 Parameter Passing: GATE1991-09b [top](#)

<https://gateoverflow.in/43603>

For the following code, indicate the output if

- a. static scope rules
- b. dynamic scope rules

are used

```
var a,b : integer;

procedure P;
    a := 5;
    b := 10;
end {P};

procedure Q;
    var a, b : integer;
    P;
end {Q};

begin
    a := 1;
    b := 2;
    Q;
    Write ('a = ', a, 'b = ', b);
end
```

gate1991 | runtime-environments | normal | compiler-design | parameter-passing

Answer**2.13.7 Parameter Passing: GATE1993-26** [top](#)<https://gateoverflow.in/2322>

A stack is used to pass parameters to procedures in a procedure call.

- A. If a procedure P has two parameters as described in procedure definition:

```
procedure P (var x :integer; y: integer);
```

and if P is called by ; $P(a,b)$

State precisely in a sentence what is pushed on stack for parameters a and b

- B. In the generated code for the body of procedure P , how will the addressing of formal parameters x and y differ?

[gate1993](#) [compiler-design](#) [parameter-passing](#) [runtime-environments](#) [normal](#)

Answer**2.13.8 Parameter Passing: GATE1995-2.4** [top](#)<https://gateoverflow.in/2616>

What is the value of X printed by the following program?

```
program COMPUTE (input, output);
var X:integer;
procedure FIND (X:real);
begin
    X:=sqrt(X);
end;
begin
    X:=2
    FIND(X);
    writeln(X);
end.
```

- A. 2
- B. $\sqrt{2}$
- C. Run time error
- D. None of the above

[gate1995](#) [programming](#) [parameter-passing](#) [runtime-environments](#) [easy](#)

Answer**2.13.9 Parameter Passing: GATE2001-2.17 | UGCNET-AUG2016-III-21** [top](#)<https://gateoverflow.in/735>

What is printed by the print statements in the program $P1$ assuming call by reference parameter passing?

```
Program P1 ()
{
    x = 10;
    y = 3;
    func1(y,x,x);
    print x;
    print y;
}
```

```
func1(x, y, z)
{
    y = y + 4;
    z = x + y + z
}
```

- A. 10, 3
 B. 31, 3
 C. 27, 7
 D. None of the above

gate2001 programming compiler-design parameter-passing normal runtime-environments ugcnetaug2016iii

Answer

2.13.10 Parameter Passing: GATE2003-74 [top](#)

<https://gateoverflow.in/43575>

The following program fragment is written in a programming language that allows global variables and does not allow nested declarations of functions.

```
global int i=100, j=5;
void P(x) {
    int i=10;
    print(x+10);
    i=200;
    j=20;
    print (x);
}
main() {P(i+j);}
```

If the programming language uses dynamic scoping and call by name parameter passing mechanism, the values printed by the above program are

- A. 115, 220
 B. 25, 220
 C. 25, 15
 D. 115, 105

gate2003 programming compiler-design parameter-passing runtime-environments normal

Answer

2.13.11 Parameter Passing: GATE2004-2,ISRO2017-54 [top](#)

<https://gateoverflow.in/998>

Consider the following function

```
void swap(int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

In order to exchange the values of two variables x and y .

- A. call $swap(x, y)$
 B. call $swap(&x, &y)$

- C. $\text{swap}(x,y)$ cannot be used as it does not return any value
 D. $\text{swap}(x,y)$ cannot be used as the parameters are passed by value

gate2004 compiler-design programming-in-c parameter-passing easy isro2017 runtime-environments

Answer

2.13.12 Parameter Passing: GATE2007-IT-33 [top](#)

<https://gateoverflow.in/3466>

Consider the program below in a hypothetical language which allows global variable and a choice of call by reference or call by value methods of parameter passing.

```
int i ;
program main ()
{
    int j = 60;
    i = 50;
    call f (i, j);
    print i, j;
}
procedure f (x, y)
{
    i = 100;
    x = 10;
    y = y + i ;
}
```

Which one of the following options represents the correct output of the program for the two parameter passing mechanisms?

- A. Call by value : $i = 70, j = 10$; Call by reference : $i = 60, j = 70$
 B. Call by value : $i = 50, j = 60$; Call by reference : $i = 50, j = 70$
 C. Call by value : $i = 10, j = 70$; Call by reference : $i = 100, j = 60$
 D. Call by value : $i = 100, j = 60$; Call by reference : $i = 10, j = 70$

gate2007-it programming parameter-passing normal compiler-design runtime-environments

Answer

2.13.13 Parameter Passing: GATE2016-1-36 [top](#)

<https://gateoverflow.in/39701>

What will be the output of the following pseudo-code when parameters are passed by reference and dynamic scoping is assumed?

```
a = 3;
void n(x) { x = x * a; print (x); }
void m(y) { a = 1 ; a = y - a; n(a); print (a); }
void main () { m(a); }
```

- A. 6,2
 B. 6,6
 C. 4,2
 D. 4,4

gate2016-1 parameter-passing normal

Answer

2.13.14 Parameter Passing: GATE2018-29 [top](#)

<https://gateoverflow.in/204103>

```
#include<stdio.h>
void fun1(char* s1, char* s2) {
    char* temp;
    temp = s1;
    s1 = s2;
    s2 = temp;
}
void fun2(char** s1, char** s2) {
    char* temp;
    temp = *s1;
    *s1 = *s2;
    *s2 = temp;
}
int main() {
    char *str1="Hi", *str2 = "Bye";
    fun1(str1, str2); printf("%s %s", str1, str2);
    fun2(&str1, &str2); printf("%s %s", str1, str2);
    return 0;
}
```

The output of the program above is:

- A. Hi Bye Bye Hi
- B. Hi Bye Hi Bye
- C. Bye Hi Hi Bye
- D. Bye Hi Bye Hi

gate2018 programming-in-c pointers parameter-passing normal

[Answer](#)

Answers: Parameter Passing

2.13.1 Parameter Passing: CMI2010-A-08 [top](#)

<https://gateoverflow.in/46140>



B) When a function is called, arguments are copied into local storage.

In call by value , a copy of variable send from where it is called and those are local to the calling function

4 votes

-- srestha (87k points)

2.13.2 Parameter Passing: GATE1988-2xv [top](#)

<https://gateoverflow.in/94333>



let variable "a" has address 100 and "b" has 200 .

and a variable in which "a + b" is stored has address 300.

now $p(300, 100, 100)$ which represent x, y, z

$y := y + 1$ // it makes $a = 3$;

$z := x + x$ // x means the value contained at address 300 i.e. 5

$5 + 5 = 10$ hence value at address 100 i.e. variable "a" will get the value 10 .

hence value of a i.e. 10 will be printed.

12 votes

-- psb (757 points)

2.13.3 Parameter Passing: GATE1989-3-viii top

<https://gateoverflow.in/37264>



Answer A, D.

A is correct as call-by-name works like a macro and substitution happens only during use time. For example if we pass $2 + 3$ to the below function

```
int foo (int x)
{
    return x * x;
}
```

we get $2 + 3 * 2 + 3$ which will be 11 due to the higher precedence for *. But, call by reference will return $5 * 5 = 25$. (For call by reference, when an expression is passed, a temporary variable is created and passed to the function)

D is also correct: Passing an array element as a parameter

See the below example:

```
void m(int x, int y) {
    for(int k = 0; k < 10; k++) {
        y = 0; x++;
    }
}

int main() {
    int j; int A[10];
    j = 0;
    m(j, A[j]);
    return 0;
}
```

For the above example if we use 'Call by name' its initialize all the array elements with 0. But if we apply ' Call by Reference ' it will only initialize $A[0]$ with 0.

3 votes

-- mystylecse (2.5k points)

2.13.4 Parameter Passing: GATE1991-03,X top

<https://gateoverflow.in/524>



- is TRUE. Left recursive grammars if used directly in recursive descent parsing causes an infinite loop. So, left recursion must be removed before giving to a recursive descent parser.
- is a strong statement- but I do not have any proof or reference for this- so for now I consider this FALSE.

- c. is false. The language can have dynamic datatypes which requires dynamically growing memory when data type size increases.
- d. is true and using macro we can do this.
- e. out of syllabus now,

15 votes

-- Arjun Suresh (350k points)

2.13.5 Parameter Passing: GATE1991-09a top

<https://gateoverflow.in/536>



1. Pass by value: Function cannot modify a variable in the calling function. So,
 $x = 1, z = 100$
2. Pass by reference: An alias of the variable (a different name but having same memory location) is used to pass the variable to a function. So, whatever change occurs for the variable in the called function is reflected in the calling function.
 $x = 2, z = 7(2 + 5)$
3. Pass by name: The expression used to call a function is copy pasted for each formal parameter. So, the body of P becomes,
 $x := 2;$
 $z := x + x * y;$

So, printed value will be
 $x = 2, z = 12$

24 votes

-- Arjun Suresh (350k points)

2.13.6 Parameter Passing: GATE1991-09b top

<https://gateoverflow.in/43603>



In static scoping, if a variable is not found (variable definition - memory allocation) in the local scope, it is looked upon in global scope. In dynamic scoping, if a variable is not found in local scope, it is looked upon in the function which called the current executing one.

1. $a = 5, b = 10$. main is using global variables. P is also using global variables.
2. $a = 1, b = 2$. main is using global variables. P is using the local variables defined in Q.

(The modification in Q, happens to the variables in P but in main we use the global variables)

15 votes

-- Arjun Suresh (350k points)

2.13.7 Parameter Passing: GATE1993-26 top

<https://gateoverflow.in/2322>

a is pointer variable so address and b is variable so its value pushed into stack.

4 votes

-- Digvijay (54.9k points)

2.13.8 Parameter Passing: GATE1995-2.4 [top](#)

<https://gateoverflow.in/2616>

i think ans should be A.

as per call by value concept. X in the procedure FIND is a local variable. no change will be reflected in global var X .

8 votes

-- jayendra (8.2k points)

2.13.9 Parameter Passing: GATE2001-2.17 | UGCNET-AUG2016-III-21 [top](#)

<https://gateoverflow.in/735>

Selected Answer
Answer is B.

Here, variable x of func1 points to address of variable y .

and variables y and z of func1 points to address of variable x .

Therefore, $y = y + 4 \Rightarrow y = 10 + 4 = 14$

and $z = x + y + z \Rightarrow z = 14 + 14 + 3 = 31$

z will be stored back in x . Hence, $x = 31$ and y will remain as it is. ($y = 3$)

Answer is 31,3

16 votes

-- jayendra (8.2k points)

2.13.10 Parameter Passing: GATE2003-74 [top](#)

<https://gateoverflow.in/43575>

Selected Answer

```
global int i=100, j=5; // memory created
//for i and j variable and 100 and 5 store in them respectively (1)

void P(x) { // p(i+j) (3)
//dynamic scoping means when a variable is not found locally,
//it refers to the caller function for it

    int i=10; // new variable created i with value 10 store in it. (4)

    print(x+10); // print(x+10); = print(i+j +10);= 10 +5 +10 = 25 (5)

    i=200; // local i and global j value changed to 200 and 20 respectively (6)

    j=20;
    print (x); // print (x);= print (i +j); = 200 +20 = 220 (7)
}

main() {P(i+j);} // 1st function call
//since it is call by name dont calculate value just send i+j
//Since there are no local i and j, they refer to the global variables (2)
```

Answer is (B).

Refer: <http://stackoverflow.com/questions/838079/what-is-pass-by-name-and-how-does-it-work-exactly>

20 votes

-- Prashant Singh (59.8k points)

2.13.11 Parameter Passing: GATE2004-2,ISRO2017-54 top



ans (D).

Option A will not exchange the values of x and y because parameters are passed by value in C. i.e., the code is exchanging x' and y' which are having the values of x and y respectively.

Option B will not swap the value

```
void swap(int a, int b)
```

Here, it is wrong to pass in address (int^*) as the parameters are of int type, even sizeof int and int^* varies depending on the compiler. Now, even if ignoring this error, the given code would not exchange the values of x and y as it is merely exchanging p'_1 and p'_2 where p'_1 and p'_2 are containing the copies of the addresses of x and y respectively. (Even addresses are passed by value in C language).

Option C is false, return value is not required for exchanging the variables.

Option D is correct. We cannot use $\text{swap}(x, y)$ because parameters are passed by value. Only way now to exchange the variables are by passing their addresses and then modifying the contents using the de-referencing operator (*).

22 votes

-- sonam vyas (14.8k points)

2.13.12 Parameter Passing: GATE2007-IT-33 top

<https://gateoverflow.in/3466>



Correct answer is (D)

CALL BY VALUE :- i as global variable declared. Then in $\text{main}()$ a local variable j as integer declared i.e $j = 60$ And global variable i initialized to 50 by $i = 50$. Now procedure f called and values of i and j are passed to it. i.e., in $f(i, j) \rightarrow f(x, y)$ content of memory location of i (here 50) is copied to memory location of x (which is different from i) and content of memory location of j (here, 60) is copied to memory location of y . Then in $f(x, y)$ $i = 100$ changes the global i to 100, $X = 10$ changes the local X from 50 to 10 and $Y = y + i$ means $y = 60 + 100 = 160$. Now when return back to main, i and j will be 100 and 60 respectively.

CALL BY REFERENCE:- Now procedure f called and passed reference of i and j to it. i.e., in $f(i, j) \rightarrow f(x, y)$ x and y are new names (aliases) pointing to the same memory location of i and j respectively. So, $i = 100$ changes the global i to 100 and $x = 10$ means x as well as global $i = 10$ (as the i being passed is the global variable and x and i share the same address). $y = y + i$ means $y = 60 + 10 = 70$ and this changes the value of j also to 70 as j and y have the same address. Now when return back to main, i and j will be 10 and 70 respectively.

29 votes

-- Kalpana Bhargav (3.3k points)

2.13.13 Parameter Passing: GATE2016-1-36 [top](#)

<https://gateoverflow.in/39701>



Selected Answer

It is a bit confusing as variable declaration is not explicit. But we can see that " $a = 3$ " and " $a = 1$ " are declaring new variables, one in global and other in local space.

ain is calling $m(a)$. Since there is no local ' a' , ' a ' here is the global one.

In m , we have " $a = 1$ " which declares a local " a " and gives 1 to it. " $a = y - a$ " assigns $3 - 1 = 2$ to ' a' .

Now, in $n(x)$, ' a ' is used and as per dynamic scoping this ' a ' comes from ' $m()$ ' and not the global one. So, " $x = x * a$ " assigns " $2 * 2 = 4$ " to " x " and 4 is printed. Being passed by reference, " a " in $m()$ also get updated to 4 . So, **D** is the answer here.

👍 69 votes

-- Arjun Suresh (350k points)

2.13.14 Parameter Passing: GATE2018-29 [top](#)

<https://gateoverflow.in/204103>



Selected Answer

```
func1(char* s1, char* s2) {
    char* temp;
    temp = s1;
    s1 = s2;
    s2 = temp;
}
```

Everything is local here. So, once function completes its execution all modification go in vain.

```
func2(char** s1, char** s2) {
    char* temp
    temp = *s1
    *s1 = *s2
    *s2 = temp
}
```

This will retain modification and swap pointers of string.

So output would be Hi Bye Bye Hi

👍 12 votes

-- Digvijay (54.9k points)

2.14

Parsing(43) [top](#)

2.14.1 Parsing: GATE1987-1-xiv [top](#)

<https://gateoverflow.in/80295>

An operator precedence parser is a

- Bottom-up parser.
- Top-down parser.
- Back tracking parser.
- None of the above.

[gate1987](#)
[compiler-design](#)
[parsing](#)
Answer

2.14.2 Parsing: GATE1989-1-iii [top](#)

<https://gateoverflow.in/87046>

Merging states with a common core may produce _____ conflicts and does not produce _____ conflicts in an LALR parser.

[gate1989](#)
[descriptive](#)
[compiler-design](#)
[parsing](#)
Answer

2.14.3 Parsing: GATE1992-02,xiii [top](#)

<https://gateoverflow.in/570>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

For a context free grammar, FOLLOW(A) is the set of terminals that can appear immediately to the right of non-terminal A in some "sentential" form. We define two sets LFOLLOW(A) and RFOLLOW(A) by replacing the word "sentential" by "left sentential" and "right most sentential" respectively in the definition of FOLLOW (A).

- A. FOLLOW(A) and LFOLLOW(A) may be different.
- B. FOLLOW(A) and RFOLLOW(A) are always the same.
- C. All the three sets are identical.
- D. All the three sets are different.

[gate1992](#)
[parsing](#)
[compiler-design](#)
[normal](#)
Answer

2.14.4 Parsing: GATE1992-02,xiv [top](#)

<https://gateoverflow.in/571>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Consider the $SLR(1)$ and $LALR(1)$ parsing tables for a context free grammar. Which of the following statement is/are true?

- A. The *goto* part of both tables may be different.
- B. The *shift* entries are identical in both the tables.
- C. The *reduce* entries in the tables may be different.
- D. The *error* entries in tables may be different

[gate1992](#)
[compiler-design](#)
[normal](#)
[parsing](#)
Answer

2.14.5 Parsing: GATE1998-1.26 [top](#)

<https://gateoverflow.in/1663>

Which of the following statements is true?

- A. SLR paper is more powerful than LALR
- B. LALR parser is more powerful than Canonical LR parser
- C. Canonical LR parser is more powerful than LALR parser
- D. The parsers SLR, Canonical CR, and LALR have the same power

gate1998 compiler-design parsing normal

Answer

2.14.6 Parsing: GATE1998-1.27 [top](#)

<https://gateoverflow.in/1664>

Type checking is normally done during

- A. lexical analysis
- B. syntax analysis
- C. syntax directed translation
- D. code optimization

gate1998 compiler-design parsing easy

Answer

2.14.7 Parsing: GATE1998-22 [top](#)

<https://gateoverflow.in/1737>

- A. An identifier in a programming language consists of up to six letters and digits of which the first character must be a letter. Derive a regular expression for the identifier.
- B. Build an $LL(1)$ parsing table for the language defined by the $LL(1)$ grammar with productions

$\text{Program} \rightarrow \text{begin } d \text{ semi } X \text{ end}$

$X \rightarrow d \text{ semi } X \mid sY$

$Y \rightarrow \text{semi } sY \mid \epsilon$

gate1998 compiler-design parsing descriptive

Answer

2.14.8 Parsing: GATE1999-1.17 [top](#)

<https://gateoverflow.in/1470>

Which of the following is the most powerful parsing method?

- A. LL (1)
- B. Canonical LR
- C. SLR
- D. LALR

gate1999 compiler-design parsing easy

Answer

2.14.9 Parsing: GATE2000-1.19, UGCNET-Dec2013-II-30 [top](#)

<https://gateoverflow.in/642>

Which of the following derivations does a top-down parser use while parsing an input string? The input is assumed to be scanned in left to right order.

- A. Leftmost derivation
- B. Leftmost derivation traced out in reverse
- C. Rightmost derivation
- D. Rightmost derivation traced out in reverse

gate2000 compiler-design parsing normal ugcnetdec2013ii

Answer**2.14.10 Parsing: GATE2001-16** [top](#)<https://gateoverflow.in/757>

Consider the following grammar with terminal alphabet $\Sigma\{a, (,), +, *\}$ and start symbol E . The production rules of the grammar are:

$$E \rightarrow aA$$

$$E \rightarrow (E)$$

$$A \rightarrow +E$$

$$A \rightarrow * E$$

$$A \rightarrow \epsilon$$

- a. Compute the FIRST and FOLLOW sets for E and A .
- b. Complete the LL(1) parse table for the grammar.

[gate2001](#) [compiler-design](#) [parsing](#) [normal](#)**Answer****2.14.11 Parsing: GATE2002-22** [top](#)<https://gateoverflow.in/875>

- A. Construct all the parse trees corresponding to $i + j * k$ for the grammar

$$E \rightarrow E + E$$

$$E \rightarrow E * E$$

$$E \rightarrow id$$

- B. In this grammar, what is the precedence of the two operators $*$ and $+$?
- C. If only one parse tree is desired for any string in the same language, what changes are to be made so that the resulting LALR(1) grammar is unambiguous?

[gate2002](#) [compiler-design](#) [parsing](#) [normal](#) [descriptive](#)**Answer****2.14.12 Parsing: GATE2003-16** [top](#)<https://gateoverflow.in/906>

Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?

- A. Removing left recursion alone
- B. Factoring the grammar alone
- C. Removing left recursion and factoring the grammar
- D. None of the above

[gate2003](#) [compiler-design](#) [parsing](#) [easy](#)**Answer****2.14.13 Parsing: GATE2003-17** [top](#)<https://gateoverflow.in/907>

Assume that the SLR parser for a grammar G has n_1 states and the LALR parser for G has n_2 states. The relationship between n_1 and n_2 is

- A. n_1 is necessarily less than n_2
- B. n_1 is necessarily equal to n_2
- C. n_1 is necessarily greater than n_2

- D. None of the above

gate2003 compiler-design parsing easy

Answer

2.14.14 Parsing: GATE2003-57 [top](#)

<https://gateoverflow.in/945>

Consider the grammar shown below.

$$S \rightarrow C\ C$$

$$C \rightarrow c\ C \mid d$$

This grammar is

- A. LL(1)
- B. SLR(1) but not LL(1)
- C. LALR(1) but not SLR(1)
- D. LR(1) but not LALR(1)

gate2003 compiler-design grammar parsing normal

Answer

2.14.15 Parsing: GATE2005-14 [top](#)

<https://gateoverflow.in/1350>

The grammar $A \rightarrow AA \mid (A) \mid \epsilon$ is not suitable for predictive-parsing because the grammar is:

- A. ambiguous
- B. left-recursive
- C. right-recursive
- D. an operator-grammar

gate2005 compiler-design parsing grammar easy

Answer

2.14.16 Parsing: GATE2005-60 [top](#)

<https://gateoverflow.in/1383>

Consider the grammar:

$$S \rightarrow (S) \mid a$$

Let the number of states in SLR (1), LR(1) and LALR(1) parsers for the grammar be n_1, n_2 and n_3 respectively. The following relationship holds good:

- A. $n_1 < n_2 < n_3$
- B. $n_1 = n_3 < n_2$
- C. $n_1 = n_2 = n_3$
- D. $n_1 \geq n_3 \geq n_2$

gate2005 compiler-design parsing normal

Answer

2.14.17 Parsing: GATE2005-83a [top](#)

<https://gateoverflow.in/1405>

Statement for Linked Answer Questions 83a & 83b:

Consider the following expression grammar. The semantic rules for expression evaluation are stated next to each grammar production.

$$\begin{array}{ll} E \rightarrow \text{number} & E.\text{val} = \text{number.val} \\ | E '+' E & E^{(1)}.\text{val} = E^{(2)}.\text{val} + E^{(3)}.\text{val} \\ | E ' \times ' E & E^{(1)}.\text{val} = E^{(2)}.\text{val} \times E^{(3)}.\text{val} \end{array}$$

The above grammar and the semantic rules are fed to a *yaac* tool (which is an LALR(1) parser generator) for parsing and evaluating arithmetic expressions. Which one of the following is true about the action of *yaac* for the given grammar?

- A. It detects *recursion* and eliminates recursion
- B. It detects *reduce-reduce* conflict, and resolves
- C. It detects *shift-reduce* conflict, and resolves the conflict in favor of a *shift* over a *reduce* action
- D. It detects *shift-reduce* conflict, and resolves the conflict in favor of a *reduce* over a *shift* action

gate2005 compiler-design parsing normal

Answer

2.14.18 Parsing: GATE2005-83b [top](#)

<https://gateoverflow.in/87037>

Consider the following expression grammar. The semantic rules for expression evaluation are stated next to each grammar production.

$$\begin{array}{ll} E \rightarrow \text{number} & E.\text{val} = \text{number.val} \\ | E '+' E & E^{(1)}.\text{val} = E^{(2)}.\text{val} + E^{(3)}.\text{val} \\ | E ' \times ' E & E^{(1)}.\text{val} = E^{(2)}.\text{val} \times E^{(3)}.\text{val} \end{array}$$

Assume the conflicts of this question are resolved using yacc tool and an LALR(1) parser is generated for parsing arithmetic expressions as per the given grammar. Consider an expression $3 \times 2 + 1$. What precedence and associativity properties does the generated parser realize?

- A. Equal precedence and left associativity; expression is evaluated to 7
- B. Equal precedence and right associativity; expression is evaluated to 9
- C. Precedence of ' \times ' is higher than that of '+', and both operators are left associative; expression is evaluated to 7
- D. Precedence of '+' is higher than that of ' \times ', and both operators are left associative; expression is evaluated to 9

gate2005 compiler-design parsing normal

Answer

2.14.19 Parsing: GATE2005-IT-83a [top](#)

<https://gateoverflow.in/3849>

Consider the context-free grammar

$$\begin{array}{l} E \rightarrow E + E \\ E \rightarrow (E * E) \end{array}$$

$E \rightarrow id$

where E is the starting symbol, the set of terminals is $\{id, (, +,), *\}$, and the set of nonterminals is $\{E\}$.

Which of the following terminal strings has more than one parse tree when parsed according to the above grammar?

- A. $id + id + id + id$
- B. $id + (id * (id * id))$
- C. $(id * (id * id)) + id$
- D. $((id * id + id) * id)$

gate2005-it compiler-design grammar parsing easy

Answer

2.14.20 Parsing: GATE2005-IT-83b top

<https://gateoverflow.in/3850>

Consider the context-free grammar

- $E \rightarrow E + E$
- $E \rightarrow (E * E)$
- $E \rightarrow id$

where E is the starting symbol, the set of terminals is $\{id, (, +,), *\}$, and the set of non-terminals is $\{E\}$.

For the terminal string $id + id + id + id$, how many parse trees are possible?

- A. 5
- B. 4
- C. 3
- D. 2

gate2005-it compiler-design parsing normal

Answer

2.14.21 Parsing: GATE2006-58 top

<https://gateoverflow.in/1836>

Consider the following grammar:

$S \rightarrow FR$

$R \rightarrow *S \mid \epsilon$

$F \rightarrow id$

In the predictive parser table, M, of the grammar the entries $M[S, id]$ and $M[R, \$]$ respectively are

- A. $\{S \rightarrow FR\}$ and $\{R \rightarrow \epsilon\}$
- B. $\{S \rightarrow FR\}$ and $\{\}$
- C. $\{S \rightarrow FR\}$ and $\{R \rightarrow *S\}$
- D. $\{F \rightarrow id\}$ and $\{R \rightarrow \epsilon\}$

gate2006 compiler-design parsing normal

Answer

2.14.22 Parsing: GATE2006-7 top

<https://gateoverflow.in/886>

Consider the following grammar

- $S \rightarrow S * E$
- $S \rightarrow E$
- $E \rightarrow F + E$
- $E \rightarrow F$
- $F \rightarrow id$

Consider the following LR(0) items corresponding to the grammar above

- i. $S \rightarrow S*.E$
- ii. $E \rightarrow F.+E$
- iii. $E \rightarrow F+.E$

Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?

- A. i and ii
- B. ii and iii
- C. i and iii
- D. None of the above

[gate2006](#) [compiler-design](#) [parsing](#) [normal](#)

Answer

2.14.23 Parsing: GATE2007-18 top

<https://gateoverflow.in/1216>

Which one of the following is a top-down parser?

- A. Recursive descent parser.
- B. Operator precedence parser.
- C. An LR(k) parser.
- D. An LALR(k) parser.

[gate2007](#) [compiler-design](#) [parsing](#) [normal](#)

Answer

2.14.24 Parsing: GATE2008-11 top

<https://gateoverflow.in/409>

Which of the following describes a handle (as applicable to LR-parsing) appropriately?

- A. It is the position in a sentential form where the next shift or reduce operation will occur
- B. It is non-terminal whose production will be used for reduction in the next step
- C. It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur
- D. It is the production p that will be used for reduction in the next step along with a position in the sentential form where the right hand side of the production may be found

[gate2008](#) [compiler-design](#) [parsing](#) [normal](#)

Answer**2.14.25 Parsing: GATE2008-55** [top](#)<https://gateoverflow.in/478>

An LALR(1) parser for a grammar G can have shift-reduce (S-R) conflicts if and only if

- A. The SLR(1) parser for G has S-R conflicts
- B. The LR(1) parser for G has S-R conflicts
- C. The LR(0) parser for G has S-R conflicts
- D. The LALR(1) parser for G has reduce-reduce conflicts

[gate2008](#) [compiler-design](#) [parsing](#) [normal](#)**Answer****2.14.26 Parsing: GATE2008-IT-79** [top](#)<https://gateoverflow.in/3393>

A CFG G is given with the following productions where S is the start symbol, A is a non-terminal and a and b are terminals.

$$S \rightarrow aS \mid AA \rightarrow aAb \mid bAa \mid \epsilon$$

For the string "*aabbaab*" how many steps are required to derive the string and how many parse trees are there?

- A. 6 and 1
- B. 6 and 2
- C. 7 and 2
- D. 4 and 2

[gate2008-it](#) [compiler-design](#) [context-free-language](#) [parsing](#) [normal](#)**Answer****2.14.27 Parsing: GATE2009-42** [top](#)<https://gateoverflow.in/1328>

Which of the following statements are TRUE?

- I. There exist parsing algorithms for some programming languages whose complexities are less than $\Theta(n^3)$
 - II. A programming language which allows recursion can be implemented with static storage allocation.
 - III. No L-attributed definition can be evaluated in the framework of bottom-up parsing.
 - IV. Code improving transformations can be performed at both source language and intermediate code level.
- A. I and II
 - B. I and IV
 - C. III and IV
 - D. I, III and IV

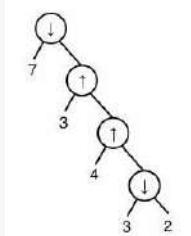
[gate2009](#) [compiler-design](#) [parsing](#) [normal](#)**Answer**

2.14.28 Parsing: GATE2011-27 [top](#)

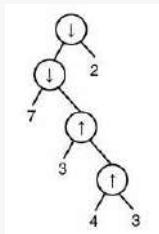
<https://gateoverflow.in/2129>

Consider two binary operators ' \uparrow ' and ' \downarrow ' with the precedence of operator \downarrow being lower than that of the operator \uparrow . Operator \uparrow is right associative while operator \downarrow is left associative. Which one of the following represents the parse tree for expression $(7 \downarrow 3 \uparrow 4 \uparrow 3 \downarrow 2)$

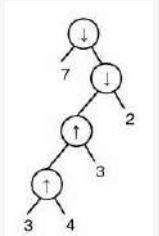
(A)



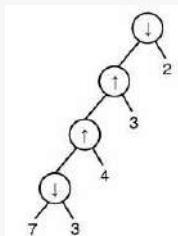
(B)



(C)



(D)



gate2011 compiler-design parsing normal

Answer

2.14.29 Parsing: GATE2012-52 top

<https://gateoverflow.in/2181>

For the grammar below, a partial $LL(1)$ parsing table is also presented along with the grammar. Entries that need to be filled are indicated as **E1**, **E2**, and **E3**. ϵ is the empty string, \$ indicates end of input, and, | separates alternate right hand sides of productions.

- $S \rightarrow aAbB \mid bAaB \mid \epsilon$

- $A \rightarrow S$
- $B \rightarrow S$

	a	b	\$
S	E1	E2	S $\rightarrow \epsilon$
A	A $\rightarrow S$	A $\rightarrow S$	error
B	B $\rightarrow S$	B $\rightarrow S$	E3

The FIRST and FOLLOW sets for the non-terminals A and B are

A.

$$\begin{aligned}\text{FIRST}(A) &= \{a, b, \epsilon\} = \text{FIRST}(B) \\ \text{FOLLOW}(A) &= \{a, b\} \\ \text{FOLLOW}(B) &= \{a, b, \$\}\end{aligned}$$

B.

$$\begin{aligned}\text{FIRST}(A) &= \{a, b, \$\} \\ \text{FIRST}(B) &= \{a, b, \epsilon\} \\ \text{FOLLOW}(A) &= \{a, b\} \\ \text{FOLLOW}(B) &= \{\$\}\end{aligned}$$

C.

$$\begin{aligned}\text{FIRST}(A) &= \{a, b, \epsilon\} = \text{FIRST}(B) \\ \text{FOLLOW}(A) &= \{a, b\} \\ \text{FOLLOW}(B) &= \emptyset\end{aligned}$$

D.

$$\begin{aligned}\text{FIRST}(A) &= \{a, b\} = \text{FIRST}(B) \\ \text{FOLLOW}(A) &= \{a, b\} \\ \text{FOLLOW}(B) &= \{a, b\}\end{aligned}$$

gate2012 compiler-design parsing normal

Answer

2.14.30 Parsing: GATE2012-53 top

<https://gateoverflow.in/43312>

For the grammar below, a partial $LL(1)$ parsing table is also presented along with the grammar. Entries that need to be filled are indicated as **E1**, **E2**, and **E3**. ϵ is the empty string, \$ indicates end of input, and, | separates alternate right hand sides of productions.

- $S \rightarrow aAbB \mid bAaB \mid \epsilon$
- $A \rightarrow S$
- $B \rightarrow S$

	a	b	\$

S	E1	E2	$S \rightarrow \epsilon$
A	$A \rightarrow S$	$A \rightarrow S$	error
B	$B \rightarrow S$	$B \rightarrow S$	E3

The appropriate entries for E1, E2, and E3 are

- A. $E1 : S \rightarrow aAbB, A \rightarrow S$
 $E2 : S \rightarrow bAaB, B \rightarrow S$
 $E1 : B \rightarrow S$
- B. $E1 : S \rightarrow aAbB, S \rightarrow \epsilon$
 $E2 : S \rightarrow bAaB, S \rightarrow \epsilon$
 $E3 : S \rightarrow \epsilon$
- C. $E1 : S \rightarrow aAbB, S \rightarrow \epsilon$
 $E2 : S \rightarrow bAaB, S \rightarrow \epsilon$
 $E3 : B \rightarrow S$
- D. $E1 : A \rightarrow S, S \rightarrow \epsilon$
 $E2 : B \rightarrow S, S \rightarrow \epsilon$
 $E3 : B \rightarrow S$

normal gate2012 compiler-design parsing

Answer

2.14.31 Parsing: GATE2013-40 top

<https://gateoverflow.in/1551>

Consider the following two sets of LR(1) items of an LR(1) grammar.

$$\begin{array}{ll} X \rightarrow c.X, c/d & X \rightarrow c.X, \$ \\ X \rightarrow .cX, c/d & X \rightarrow .cX, \$ \\ X \rightarrow .d, c/d & X \rightarrow .d, \$ \end{array}$$

Which of the following statements related to merging of the two sets in the corresponding LALR parser is/are **FALSE**?

1. Cannot be merged since look aheads are different.
 2. Can be merged but will result in S-R conflict.
 3. Can be merged but will result in R-R conflict.
 4. Cannot be merged since goto on c will lead to two different sets.
- A. 1 only
 - B. 2 only
 - C. 1 and \$44 only

- D. 1, 2, 3 and 4

gate2013 compiler-design parsing normal

[Answer](#)

2.14.32 Parsing: GATE2013-9 [top](#)

<https://gateoverflow.in/1418>

What is the maximum number of reduce moves that can be taken by a bottom-up parser for a grammar with no epsilon and unit-production (i.e., of type $A \rightarrow \epsilon$ and $A \rightarrow a$) to parse a string with n tokens?

- A. $n/2$
- B. $n - 1$
- C. $2n - 1$
- D. 2^n

gate2013 compiler-design parsing normal

[Answer](#)

2.14.33 Parsing: GATE2014-1-34 [top](#)

<https://gateoverflow.in/1807>

A canonical set of items is given below

$$S \rightarrow L. > R$$

$$Q \rightarrow R.$$

On input symbol $<$ the set has

- A. a shift-reduce conflict and a reduce-reduce conflict.
- B. a shift-reduce conflict but not a reduce-reduce conflict.
- C. a reduce-reduce conflict but not a shift-reduce conflict.
- D. neither a shift-reduce nor a reduce-reduce conflict.

gate2014-1 compiler-design parsing normal

[Answer](#)

2.14.34 Parsing: GATE2015-3-16 [top](#)

<https://gateoverflow.in/8413>

Among simple LR (SLR), canonical LR, and look-ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is the most powerful, in that order?

- A. SLR, LALR
- B. Canonical LR, LALR
- C. SLR, canonical LR
- D. LALR, canonical LR

gate2015-3 compiler-design parsing normal

[Answer](#)

2.14.35 Parsing: GATE2015-3-31 [top](#)

<https://gateoverflow.in/8488>

Consider the following grammar G

$$S \rightarrow F \mid H$$

$$F \rightarrow p \mid c$$

$$H \rightarrow d \mid c$$

Where S , F , and H are non-terminal symbols, p, d , and c are terminal symbols. Which of the following statement(s) is/are correct?

S1: LL(1) can parse all strings that are generated using grammar G

S2: LR(1) can parse all strings that are generated using grammar G

- A. Only S1
- B. Only S2
- C. Both S1 and S2
- D. Neither S1 and S2

[gate2015-3](#) [compiler-design](#) [parsing](#) [normal](#)

Answer

2.14.36 Parsing: GATE2016-1-45 [top](#)

<https://gateoverflow.in/39697>

The attribute of three arithmetic operators in some programming language are given below.

OPERATOR	PRECEDENCE	ASSOCIATIVITY	ARITY
+	high	Left	Binary
-	Medium	Right	Binary
*	Low	Left	Binary

The value of the expression $2 - 5 + 1 - 7 * 3$ in this language is _____.

[gate2016-1](#) [compiler-design](#) [parsing](#) [normal](#) [numerical-answers](#)

Answer

2.14.37 Parsing: GATE2017-1-17 [top](#)

<https://gateoverflow.in/118297>

Consider the following grammar:

- $P \rightarrow xQRS$
- $Q \rightarrow yz|z$
- $R \rightarrow w|\epsilon$
- $S \rightarrow y$

What is FOLLOW(Q)?

- A. $\{R\}$
- B. $\{w\}$
- C. $\{w, y\}$
- D. $\{w, \$\}$

[gate2017-1](#) [compiler-design](#) [parsing](#)

Answer

2.14.38 Parsing: GATE2017-1-43 [top](#)

<https://gateoverflow.in/118326>

Consider the following grammar:

- $\text{stmt} \rightarrow \text{if expr then expr else expr; stmt} \mid 0$
- $\text{expr} \rightarrow \text{term relop term} \mid \text{term}$
- $\text{term} \rightarrow \text{id} \mid \text{number}$
- $\text{id} \rightarrow \text{a} \mid \text{b} \mid \text{c}$
- $\text{number} \rightarrow [0 - 9]$

where **relop** is a relational operator (e.g.. $<$. $>$...). **0** refers to the empty statement, and **if**, **then**, **else** are terminals.

Consider a program P following the above grammar containing ten **if** terminals. The number of control flow paths in P is_____ . For example. the program

if e_1 **then** e_2 **else** e_3

has 2 control flow paths. $e_1 \rightarrow e_2$ and $e_1 \rightarrow e_3$.

gate2017-1 compiler-design parsing normal numerical-answers

Answer

2.14.39 Parsing: GATE2017-2-6 [top](#)

<https://gateoverflow.in/118343>

Which of the following statements about parser is/are CORRECT?

- I. Canonical LR is more powerful than SLR
 - II. SLR is more powerful than LALR
 - III. SLR is more powerful than Canonical LR
- A. I only
 - B. II only
 - C. III only
 - D. II and III only

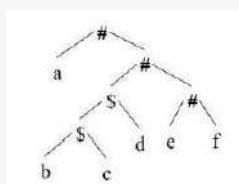
gate2017-2 compiler-design parsing

Answer

2.14.40 Parsing: GATE2018-38 [top](#)

<https://gateoverflow.in/204112>

Consider the following parse tree for the expression $a\#bcd\#e\#f$, involving two binary operators $\$$ and $\#$.



Which one of the following is correct for the given parse tree?

- A. $\$$ has higher precedence and is left associative; $\#$ is right associative
- B. $\#$ has higher precedence and is left associative; $\$$ is right associative
- C. $\$$ has higher precedence and is left associative; $\#$ is left associative
- D. $\$$ has higher precedence and is right associative; $\#$ is left associative

gate2018 compiler-design parsing normal

Answer

2.14.41 Parsing: TIFR2012-B-17 [top](#)

<https://gateoverflow.in/25215>

Which of the following correctly describes $LR(k)$ parsing?

- A. The input string is alternately scanned left to right and right to left with k reversals.
- B. Input string is scanned once left to right with rightmost derivation and k symbol look-ahead.
- C. $LR(k)$ grammars are expressively as powerful as context-free grammars.
- D. Parser makes k left-to-right passes over input string.
- E. Input string is scanned from left to right once with k symbol to the right as look-ahead to give left-most derivation.

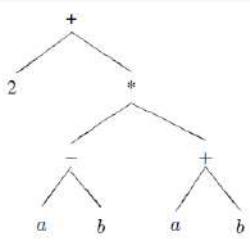
[tifr2012](#) [compiler-design](#) [parsing](#)

Answer

2.14.42 Parsing: TIFR2012-B-8 [top](#)

<https://gateoverflow.in/25108>

Consider the parse tree



Assume that $*$ has higher precedence than $+$, $-$ and operators associate right to left (i.e $a + b + c = (a + (b + c))$). Consider

- i. $2 + a - b$
- ii. $2 + a - b * a + b$
- iii. $2 + ((a - b) * (a + b)))$
- iv. $2 + (a - b) * (a + b)$

The parse tree corresponds to

- A. Expression (i)
- B. Expression (ii)
- C. Expression (iv) only
- D. Expression (ii), (iii), and (iv)
- E. Expression (iii) and (iv) only

[tifr2012](#) [compiler-design](#) [parsing](#)

Answer

2.14.43 Parsing: TIFR2015-B-15 [top](#)

<https://gateoverflow.in/30079>

Consider the following grammar (the start symbol is E) for generating expressions.

$$E \rightarrow T - E | T + E | T$$

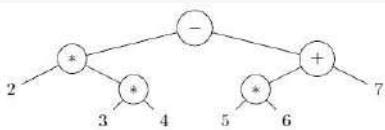
$$T \rightarrow T * F | F$$

$$F \rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$

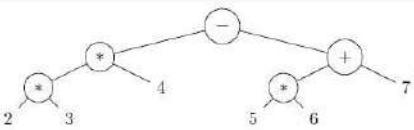
With respect to this grammar, which of the following trees is the valid evaluation tree for the

expression $2 * 3 * 4 - 5 * 6 + 7$?

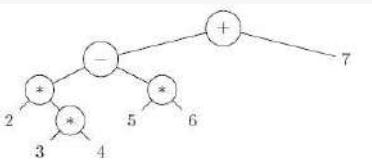
(a)



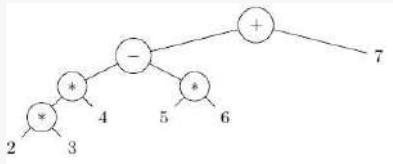
(b)



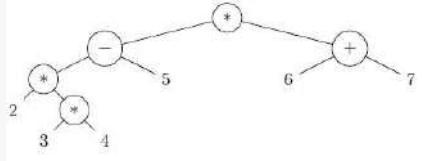
(c)



(d)



(e)



tifr2015 parsing

Answers: Parsing

2.14.1 Parsing: GATE1987-1-xiv

<https://gateoverflow.in/80295>



Selected Answer

A. Bottom-up parser.

An operator-precedence parser is a simple shift-reduce parser that is capable of parsing a subset of LR(1) grammars. More precisely, the operator-precedence parser can parse all LR(1) grammars where two consecutive non-terminals and epsilon never appear in the right-hand side of any rule.

 4 votes

-- Soumya Jain (10k points)

2.14.2 Parsing: GATE1989-1-iii top

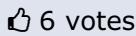
<https://gateoverflow.in/87046>



Selected Answer

Merging states with a common core may produce **Reduce - Reduce** conflicts and does not produce **Shift- Reduce** conflicts in an LALR parser.

Refer : [Link](#)



6 votes

-- Prashant Singh (59.8k points)

2.14.3 Parsing: GATE1992-02,xiii top

<https://gateoverflow.in/570>



Selected Answer

Ans - a,b, LFollow may be different but RFollow and Follow will be same

Consider a Grammar -

$$S \rightarrow AB$$

$$A \rightarrow a$$

$$B \rightarrow b$$

Now only string derivable is $\{ab\}$.

Let's find Follow(A) in all cases :

- i. **Follow(A)** - set of terminals that can appear immediately to the right of non-terminal A in some "sentential" form

$$S \rightarrow AB \rightarrow Ab \rightarrow ab$$

Here, we notice only ' b ' can appear to the right of A .

$$\text{Follow}(A) = \{b\}$$

- i. **LFollow(A)** - set of terminals that can appear immediately to the right of non-terminal A in some "left sentential" form

$$S \rightarrow AB \rightarrow aB \rightarrow ab$$

Here, we notice no terminal can appear to the right of A .

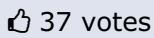
$$\text{LFollow}(A) = \{\}$$

- i. **RFollow(A)** - set of terminals that can appear immediately to the right of non-terminal A in some "right most sentential" form

$$S \rightarrow A\bar{B} \rightarrow A\bar{b} \rightarrow ab$$

Here, we notice only ' b ' can appear to the right of A .

$$\text{RFollow}(A) = \{b\}$$



37 votes

-- Himanshu Agarwal (15.3k points)

2.14.4 Parsing: GATE1992-02,xiv top

<https://gateoverflow.in/571>



- Goto part & shift entry must be same.
- Reduce entry & error entry may b different due to conflicts.

1 21 votes

-- Digvijay (54.9k points)

2.14.5 Parsing: GATE1998-1.26 top

<https://gateoverflow.in/1663>



- SLR paper is more powerful than LALR . **False** .
- LALR parser is more powerful than Canonical LR parser . **False** .
- Canonical LR parser is more powerful than LALR parser. **True**.
- The parsers SLR, Canonical CR, and LALR have the same power. **False**.

answer - C

1 16 votes

-- Ankit Rokde (9k points)

2.14.6 Parsing: GATE1998-1.27 top

<https://gateoverflow.in/1664>



The answer is c .

The use of syntax analyser is used to create parse Tree. But along with Grammar as input to Syntax Analyser we add even semantic rules which form the basis of Syntax Directed Translation That help us in Evaluation of Expression .Remember that

Syntax Directed Translation is used in following cases

1. Conversion of infix to Postfix
2. Calculation of infix expression
3. For creating a Acyclic graph
4. Type Checking
5. Conversion of Binary number to Decimal
6. Counting the numbers of bits (0 or 1) in a binary number
7. Creation of syntax tree
8. To generate Intermediate code
9. Storing the data into Symbol table

1 35 votes

-- spriti1991 (1.9k points)

2.14.7 Parsing: GATE1998-22 top

<https://gateoverflow.in/1737>



a. $(letter)(letter + digit + \epsilon)^5$

b. Program \rightarrow begin d semi X end ----- A

$X \rightarrow d \text{ semi} X$ ----- B

$X \rightarrow sY$ ----- C

$Y \rightarrow \text{semi } sY$ ----- D

$Y \rightarrow \epsilon$ ----- E

Variable	First	Follow
Program	begin	\$
X	d, s	end
Y	semi, ϵ	end

Here First(Y) contains ϵ so we need to add $Y \rightarrow \epsilon$ at follow(Y)

Variable	begin	d	semi	s	end	\$
Program	A					
X		B		C		
Y			D		Y $\rightarrow \epsilon$	

16 votes

-- papesh (25.7k points)

2.14.8 Parsing: GATE1999-1.17 top

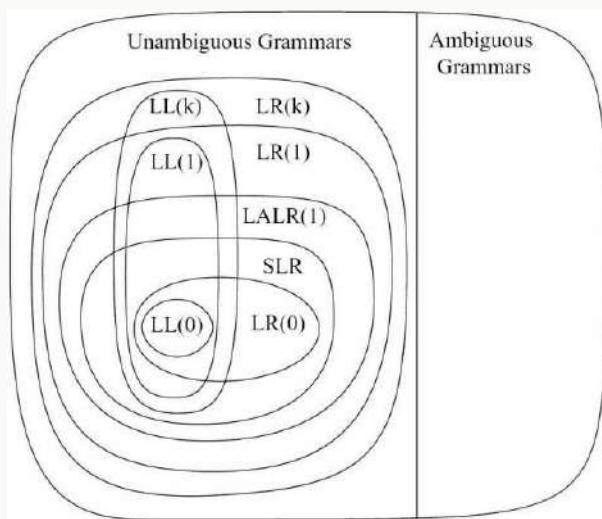
<https://gateoverflow.in/1470>



Canonical LR is most powerful method

LR > LALR > SLR

so ans is B



10 votes

-- Pooja Palod (31.3k points)

2.14.9 Parsing: GATE2000-1.19, UGCNET-Dec2013-II-30 [top](#)



ans A

11 votes

-- Aditi Dan (5.2k points)

2.14.10 Parsing: GATE2001-16 [top](#)

<https://gateoverflow.in/757>



First (E) = { $a, ()$ }

First (A) = { $+, *, \epsilon$ }

Follow (E) = Follow (A) = { $\$,)$ }

LL(1) Parsing Table :

	a	()	+	*	\$
E	$E \rightarrow aA$	$E \rightarrow (E)$				
A			$A \rightarrow \epsilon$	$A \rightarrow +E$	$A \rightarrow *E$	$A \rightarrow \epsilon$

20 votes

-- Aditya Gaurav (3k points)

2.14.11 Parsing: GATE2002-22 [top](#)

<https://gateoverflow.in/875>



- a. two parse tree for $i+j*k$.
- b. $+$ and $*$ having same precedence..
- c. to make grammar LALR compatible give priority to $+$ over $*$ or vice versa.

following grammar is LALR(1)

$$\begin{aligned} E &\rightarrow E + T \\ &\quad | T \\ T &\rightarrow T * F \\ &\quad | F \\ F &\rightarrow id \end{aligned}$$

18 votes

-- Digvijay (54.9k points)

2.14.12 Parsing: GATE2003-16 [top](#)

<https://gateoverflow.in/906>



LL(1) parser is top down parser.

For top down parsers, the grammar should be unambiguous, deterministic and should not be left recursive.

All the **3** conditions must be satisfied for LL(1) parsers.

Now, even if all **3** conditions are satisfied we cannot get an LL(1) or even LL(k) (for any k) grammar for even a DCFG. This is because there are DCFLs which does not have an LL(k) grammar (see ref below). On the other hand for any DCFL, we can always have an LR(1) grammar.

<http://mathoverflow.net/questions/31733/can-i-have-an-ll-grammar-for-every-deterministic-context-free-language>

So, option **D** is correct.

1 34 votes

-- Monanshi Jain (9.3k points)

2.14.13 Parsing: GATE2003-17 [top](#)

<https://gateoverflow.in/907>



Selected Answer

no of states in SLR and LALR are equal

and no of states in SLR and LALR are less than or equal to LR(1)

1 17 votes

-- Pooja Palod (31.3k points)

2.14.14 Parsing: GATE2003-57 [top](#)

<https://gateoverflow.in/945>



Selected Answer

ans is **A**

$$\text{First}(S) = \text{First}(C) = \{c, d\}$$

there are no multiple in single row of parsing table hence grammar is LL1

note : if we have $A \rightarrow B/C$ for grammar to be LL(1) first(B) intersection First(C) should be null otherwise grammar is not LL1. If First(B) contains ϵ then Follow(A) intersection First(C) should be null. Using this we can say grammar is LL(1) or not without constructing parsing table.

An ϵ free LL(1) grammar is also SLR(1) and hence LALR(1) and LR(1) too.

1 31 votes

-- Pooja Palod (31.3k points)

2.14.15 Parsing: GATE2005-14 [top](#)

<https://gateoverflow.in/1350>



Selected Answer

both **A** and **B** can be answers but **A** is a better answer. Because we have standard procedure for removing left-recursion but ambiguity is not easy to remove. - checking if a given CFG is ambiguous is a undecidable problem.

32 votes

-- Vikrant Singh (13.5k points)

2.14.16 Parsing: GATE2005-60 top

<https://gateoverflow.in/1383>

Selected Answer

ans B

Both in SLR(1) and LALR(1), states are the LR(0) items while in LR(1) the states are LR(1) set of items. Number of LR(0) items can never be greater than number of LR(1) items. So, $n_1 = n_3 \leq n_2$, B choice. If we construct the states for the grammar we can replace \leq with $<$.

 23 votes

-- Aditi Dan (5.2k points)

2.14.17 Parsing: GATE2005-83a top

<https://gateoverflow.in/1405>

Selected Answer

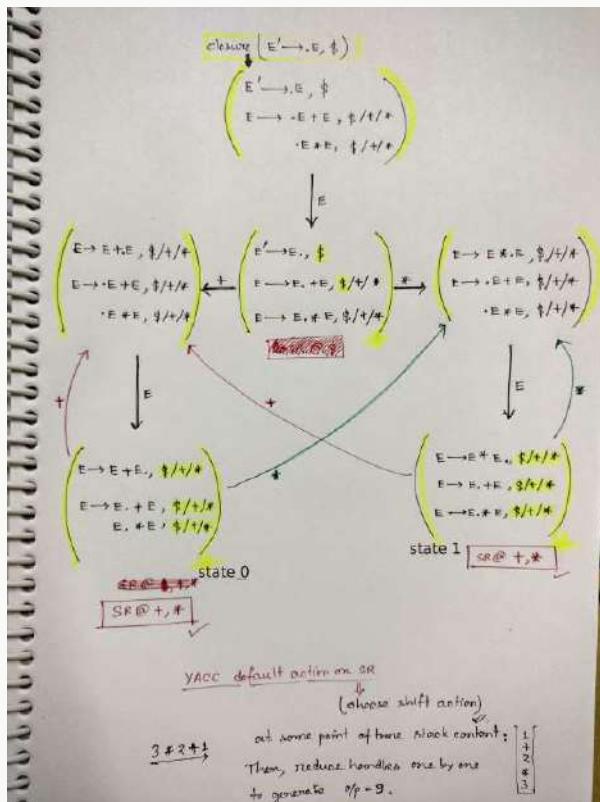
Given grammar :

$$E \rightarrow \text{num}$$

$$E \rightarrow E + E \quad | \quad E * E$$

First $LR(1)$ item

$$E' \rightarrow \bullet E, \$$$



- num does not create any conflict.
- Additionally here no states differ by lookahead symbols only.
- \Rightarrow LALR(1) and LR(1) tables are same.

- $LR(1)$ table only for `state0` and `state1`:

	+	*	\$
0	SR	SR	R
1	SR	SR	R

So total $2 + 2 = 4$ SR conflict originated in two states of the DFA.

- **Shift-reduce conflict:** Yacc's default action in the case of a shift-reduce conflict is to choose the shift action.
- **Reduce-reduce conflict :** Yacc's default action in the case of a reduce-reduce conflict is to reduce using the production that comes first, textually, in the input grammar specification.

and LEX-YACC-gcc output after implementing the given grammar :

```
exp : number      ($$ = $1;    printf("\n%d step = %d : reduction,(E->num) %d->%d\n",++step,$$,$$);)
| exp '+' exp  ($$ = $1 + $3; printf("\n%d step = %d : reduction,(E->E+E) %d->%d\n",++step,$$,$$,$$);)
| exp '*' exp  ($$ = $1 * $3; printf("\n%d step = %d : reduction,(E->E*E) %d->%d\n",++step,$$,$$,$$);)
| identifier    ($$ = symbolVal($1););
|
```

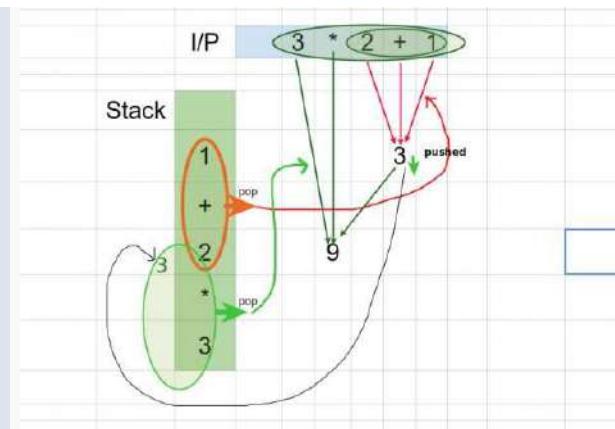
```
% - o  d@D:/Desktop/Gate_C_programs/yacc
d@D:/Desktop/Gate_C_programs/yacc$ yacc -d calc.y
calc.y: Warning: 4 shift/reduce conflicts [-Wconflicts-sr]
d@D:/Desktop/Gate_C_programs/yacc$ lex calc.l
d@D:/Desktop/Gate_C_programs/yacc$ gcc lex.yy.c y.tab.c -o calc
d@D:/Desktop/Gate_C_programs/yacc$ ./calc

a = 3*2+1;
    step = 1 : i/p scan: 3
    step = 2 : reduction,(E->num) 3->3
    step = 3 : i/p scan: *
    step = 4 : i/p scan: 2
    step = 5 : reduction,(E->num) 2->2
    step = 6 : i/p scan: +
    step = 7 : i/p scan: 1
    step = 8 : reduction,(E->num) 1->1
    step = 9 : i/p scan: ;
    step = 10 : reduction,(E->E+E) 3->2+1
    step = 11 : reduction,(E->E*E) 9->3*3
print a;
    step = 12 : i/p scan: ;
Printing 9
```

As we can see from the output reduction on $E \rightarrow \text{num}$ is carried out as soon as top of stack contains a `num`. So, no conflict related to $E \rightarrow \text{num}$.

one example : Because of YACC shift preference, even if $3 * 2$ ($E * E$) handle found on top of the stack at some point of time, it will shift on reading $+$ instead of reducing with $E \rightarrow E * E$. In this way, the complete input will be pushed into the stack. After that only reduce work starts as shown below.

- **Equal precedence** because of the given grammar $E \rightarrow E + E \mid E * E$, (single level)
- and **Right associativity** :



[How YACC handles conflicts](#)

Here are the **required files** ([calc.l](#) and [calc.y](#)) to regenerate the above interpreter.

66 votes

-- Debashish Deka (56.7k points)

2.14.18 Parsing: GATE2005-83b [top](#)

<https://gateoverflow.in/87037>



Selected Answer

LALR Parser is type of **Bottom up Parser** which uses **Right most Derivation**

For $3 \times 2 + 1$

$E \rightarrow E * E$ (Both shift and reduce possible but yacc prefers shift)

$$\begin{aligned} &\rightarrow E * E + E \\ &\rightarrow E * E + 1 \\ &\rightarrow E * 2 + 1 \\ &\rightarrow E * 3 \\ &\rightarrow 3 * 3 \\ &\rightarrow 9 \end{aligned}$$

All the productions are in same level therefore all have same precedence

Therefore Ans is **B. Equal precedence and right associativity; expression is evaluated to 9**

23 votes

-- Prajwal Bhat (11.3k points)

2.14.19 Parsing: GATE2005-IT-83a [top](#)

<https://gateoverflow.in/3849>



Selected Answer

Answer is **A**.

18 votes

-- Gate Keeda (19.6k points)

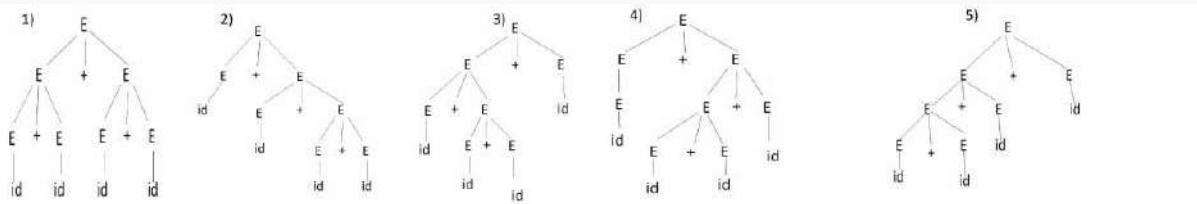
2.14.20 Parsing: GATE2005-IT-83b top

<https://gateoverflow.in/3850>



Selected Answer

5 Parse trees are possible



11 votes

-- Shaji Thorn Blue (7.4k points)

2.14.21 Parsing: GATE2006-58 top

<https://gateoverflow.in/1836>



Selected Answer

First $S = \{id\}$
Follow $R = \{\$\}$

so $M[S, id] = S \rightarrow FR$
 $M[S, \$] = R \rightarrow \epsilon$

So ans is A

17 votes

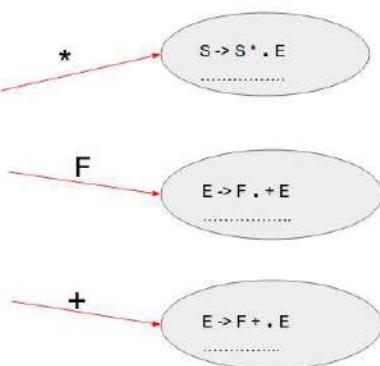
-- Pooja Palod (31.3k points)

2.14.22 Parsing: GATE2006-7 top

<https://gateoverflow.in/886>



Selected Answer

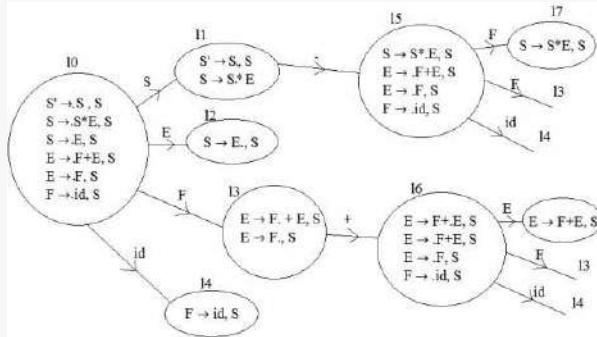


⇒ NOT possible for these three items to be in same state

11 votes

-- Debashish Deka (56.7k points)

ans is D.



20 votes

-- jayendra (8.2k points)

2.14.23 Parsing: GATE2007-18 [top](#)

<https://gateoverflow.in/1216>



1. Recursive descent parser-TOP DOWN PARSER
 2. Operator precedence parser-BOTTOM UP PARSER
 3. An LR(k) parser.-BOTTOM UP PARSER
 4. An LALR(k) parser-BOTTOM UP PARSER

7 votes

-- vnc (2.6k points)

2.14.24 Parsing: GATE2008-11 [top](#)

<https://gateoverflow.in/409>



A **sentential form** is the start symbol S of a grammar or any string in $(V \cup T)^*$ that can be derived from S .

Consider the linear grammar

$(\{S, B\}, \{a, b\}, S, \{S \rightarrow aS, S \rightarrow B, B \rightarrow bB, B \rightarrow \lambda\})$.

A derivation using this grammar might look like this:

$$S \Rightarrow aS \Rightarrow aB \Rightarrow abB \Rightarrow abbB \Rightarrow abb$$

Each of $\{S, aS, aB, abB, abbB, abb\}$ is a sentential form.

Because this grammar is linear, each sentential form has at most one variable. Hence there is never any choice about which variable to expand next.

Here, in option D the sentential forms are same but generated differently coz we are using here

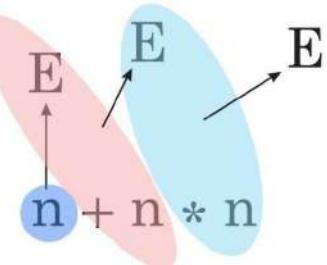
Bottom Up production.

Handle:

for example the grammar is:

$$\begin{aligned} E &\rightarrow E + n \\ E &\rightarrow E * n \\ E &\rightarrow n \end{aligned}$$

then say to derive string $n + n * n$:



these are three different handles shown in 3 different colors = $\{n, E + n, E * n\}$

that's what **option D** says

39 votes

-- Amar Vashishth (30.5k points)

2.14.25 Parsing: GATE2008-55 top

<https://gateoverflow.in/478>



Both LALR(1) and LR(1) parser uses LR(1) set of items to form their parsing tables. And LALR(1) states can be found by merging LR(1) states of LR(1) parser that have the same set of first components of their items.

i.e. if LR(1) parser has 2 states I and J with items $A \rightarrow a.bP, x$ and $A \rightarrow a.bP, y$ respectively, where x and y are look ahead symbols, then as these items are same with respect to their first component, they can be merged together and form one single state, let's say K . Here we have to take union of look ahead symbols. After merging, State K will have one single item as $A \rightarrow a.bP, x \cup y$. This way LALR(1) states are formed (i.e. after merging the states of LR(1)).

Now, $S - R$ conflict in LR(1) items can be there whenever a state has items of the form :

$A \rightarrow a.bB, p$
 $C \rightarrow d., b$

i.e. it is getting both shift and reduce at symbol b , hence a conflict.

Now, as LALR(1) have items similar to LR(1) in terms of their first component, shift-reduce form will only take place if it is already there in LR(1) states. If there is no S-R conflict in LR(1) state it will never be reflected in the LALR(1) state obtained by combining LR(1) states.

21 votes

-- Madhab Paul Choudhury (4.2k points)

2.14.26 Parsing: GATE2008-IT-79 top

<https://gateoverflow.in/3393>



$S \rightarrow aS$
 $\quad 1 \rightarrow aA$
 $\quad 2 \rightarrow aaAb$
 $\quad 3 \rightarrow aabAab$
 $\quad 4 \rightarrow aabbAaab$
 $\quad 5 \rightarrow aabbaab$
 $\quad 6 \rightarrow aabbaab$

Thus 6 steps are needed and only one way to derive the string so only one parse tree.

Upvotes: 34 votes

-- Shreyans Dhankhar (2.5k points)

2.14.27 Parsing: GATE2009-42 top

<https://gateoverflow.in/1328>



Answer is **B**.

- A. Yes there does exist parsing algos less than $\Theta(n^3)$.
- B. It cannot be implemented with static storage allocation. It needs dynamic memory allocation.
- C. If the L-attributed definitions contain synthesized attributes then it can be evaluated.
- D. True.

Upvotes: 26 votes

-- Gate Keeda (19.6k points)

2.14.28 Parsing: GATE2011-27 top

<https://gateoverflow.in/2129>



Answer is **B**.

To make the parse tree start compiling the identifiers into blocks based on associativity and precedence.

Grouping: $(7 \downarrow (3 \uparrow (4 \uparrow 3))) \downarrow 2$

Tree can be made by opening inner braces and move towards braces.

Upvotes: 28 votes

-- Sona Praneeth Akula (4.2k points)

2.14.29 Parsing: GATE2012-52 top

<https://gateoverflow.in/2181>



$\text{First}(S) = \text{First}(A) = \text{First}(B) = \{a, b, \epsilon\}$

$\text{Follow}(A) = \{a, b\}$

$\text{Follow}(B) = \text{Follow}(S) = \{a, b, \$\}$

So, the answer to question 52 is option A.

12 votes

-- Pooja Palod (31.3k points)

2.14.30 Parsing: GATE2012-53 top

<https://gateoverflow.in/43312>



Selected Answer

Make your own parse table. Firstly calculate First and Follow from the given grammar.

- $\text{First}(S) = \{a, b, \epsilon\}$
- $\text{First}(A) = \{a, b, \epsilon\}$
- $\text{First}(B) = \{a, b, \epsilon\}$
- $\text{Follow}(S) = \{a, b, \$\}$
- $\text{Follow}(A) = \{a, b\}$
- $\text{Follow}(B) = \{a, b, \$\}$

Now make $LL(1)$ parse table

Non Terminal	a	b	\$
S	$S \rightarrow aAbB$	$S \rightarrow bAbB$	$S \rightarrow \epsilon$
	$S \rightarrow \epsilon$	$S \rightarrow \epsilon$	
A	$A \rightarrow S$	$A \rightarrow S$	
B	$B \rightarrow S$	$B \rightarrow S$	$B \rightarrow S$

Here is the explanation of entries asked in question

1. For E1 and E2 Look into $\text{First}(S) = \{a, b, \epsilon\}$.
a is because of $S \rightarrow aAbB$ and b is because of $B \rightarrow bAaB$
So $M[S, a]$ and $M[S, b]$ will contain $S \rightarrow aAbB$ and $B \rightarrow bAaB$ respectively. For epsilon
Look into $\text{Follow}(S) = \{a, b, \$\}$. So $S \rightarrow \epsilon$ will be in $M[S, a]$, $M[S, b]$ and $M[S, \$]$
2. Now for E2 look into $\text{First}(B) = \{a, b, \$\}$. a and b are because of $B \rightarrow S$.
So $M[B, a]$ and $M[B, b]$ will contain $B \rightarrow S$ and for epsilon look into $\text{Follow}(B) = \{a, b, \$\}$.
Hence $M[B, \$]$ will contain $B \rightarrow S$

Now we get the answer as E1 is $S \rightarrow aAbB$, $S \rightarrow \epsilon$, E2 is $S \rightarrow bAaB$, $S \rightarrow \epsilon$ and E3 is $B \rightarrow S$.

Hence **Option (C)** is correct.

20 votes

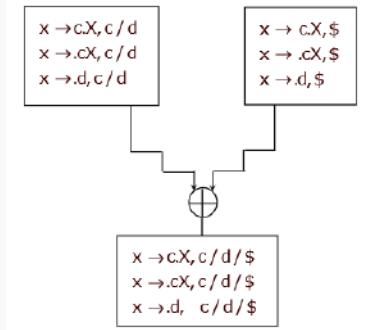
-- Ashwani Kumar (11.4k points)

2.14.31 Parsing: GATE2013-40 top

<https://gateoverflow.in/1551>



Selected Answer



The TRUE statements are about merging of two states for LALR(1) parser from RR(1) parser.i.e.

1. These can be merged because kernel of these are same, look ahead don't matter in merging
2. Two states are not containing reduces item , so after merging , the merged states can not be contain any S-R conflict.
3. There is no reduction possible so no R-R conflict
4. Merging of states does not depend on further GOTO part on any terminal.

Therefore , ALL given statement in question are FALSE , so option (D) is correct.

31 votes

-- Viral Kapoor (2.1k points)

2.14.32 Parsing: GATE2013-9 top

<https://gateoverflow.in/1418>



Selected Answer

Ans will be **B**

- $A \rightarrow BC$
- $B \rightarrow aa$
- $C \rightarrow bb$

Now suppose string is *aabb*. Then

$A \rightarrow BC$ (reduction 3)
 $\rightarrow aaC$ (reduction 2)
 $\rightarrow aabb$ (reduction 1)

$n = 4$ and number of reductions is 3. So, $n - 1$

38 votes

-- rahulkrr (805 points)

2.14.33 Parsing: GATE2014-1-34 top

<https://gateoverflow.in/1807>



Selected Answer

The question is asked with respect to the symbol ' $<$ ' which is **not present** in the given canonical set of items. Hence it is neither a shift-reduce conflict nor a reduce-reduce conflict on symbol ' $<$ '. Hence **D** is the correct option. But if the question would have asked with respect to the symbol ' $>$ ' then it would have been a shift-reduce conflict.

12 votes

-- Divya Bharti (8.1k points)

2.14.34 Parsing: GATE2015-3-16 top

<https://gateoverflow.in/8413>

Selected Answer

Answer is **C**.

SLR is the simplest to implement and Canonical LR is the most powerful.

http://en.wikipedia.org/wiki/LALR_parser_generator

30 votes

-- Arjun Suresh (350k points)

2.14.35 Parsing: GATE2015-3-31 top

<https://gateoverflow.in/8488>

Selected Answer

A parser works on the basis of given grammar. It takes the grammar as it is. Parser does not work on the basis of the yield of the grammar. Also, while constructing the LL(1) parser table, that entry for terminal 'c' will contain multiple entries. So, LL(1) parser cannot be constructed for the given grammar.

$$S \rightarrow F|H$$

$$F \rightarrow p|c$$

$$H \rightarrow d|c$$

That $\{p, d, c\}$ are the strings generated by the grammar is absolutely correct. But LL(1) and LR(1) can parse these strings successfully only if the grammar is unambiguous and like given below...

$$S \rightarrow P|D|C$$

$$P \rightarrow p$$

$$D \rightarrow d$$

$$C \rightarrow c$$

Please note the difference between these two grammars. Both derive the same strings, but in different manner. With the grammar given in the question, both top-down and bottom-up parsers will get confused while deriving "c". Top-down parser will get confused between $F \rightarrow c$ and $H \rightarrow c$. Similarly, bottom-up parser will get confused while reducing "c". This confusion in case of bottom-up parsing is technically termed as "reduce-reduce" conflict.

While top-down parsing, first(F) and first(H) are not disjoint, so the grammar cannot be LL(1). Therefore, LL(1) parser cannot parse it.

Hence, the answer should be option **(D)**. Neither S1 nor S2.

42 votes

-- ashishacm (331 points)

2.14.36 Parsing: GATE2016-1-45 top

<https://gateoverflow.in/39697>

Selected Answer

$2 - 5 + 1 - 7 * 3$ will be evaluated according to the precedence and associativity as given in the question as follows:
 $((2 - ((5 + 1) - 7)) * 3) \Rightarrow ((2 - (-1)) * 3) \Rightarrow 9$

45 votes

-- Monanshi Jain (9.3k points)

2.14.37 Parsing: GATE2017-1-17 top

<https://gateoverflow.in/118297>



Selected Answer

Follow of Q is first of R so we get $\{w\}$

but since R can be Null so we have to check first of S which is $\{y\}$

so FOLLOW $Q = \{w, y\}$

Correct option (C)

13 votes

-- sriv_shubham (3.3k points)

2.14.38 Parsing: GATE2017-1-43 top

<https://gateoverflow.in/118326>



Selected Answer

This question is picked from area of **Counting in Combinatorics**.

Given:

if e_1 then e_2 else e_3 has 2 control flow paths $e_1 \rightarrow e_2$ and $e_1 \rightarrow e_3$.
 (Meaning of "how many control flow" for if structure is clearly mentioned)

What is asked:

Number of control flow paths for 10 if terminals?

Solution:

To get 10 if's we need to use grammar to get,

```
if <expr> then <expr> else <expr> ; stmt
if <expr> then <expr> else <expr> ; if <expr> then <expr> else <expr> ; stmt
```

.....

(keep doing it 10 times to get 10 if's)

Observe that there is a **semi-colon after every if structure**.

We know that every if structure has 2 control flows as given in question. Hence,

We have 2 control flow choices for 1st if terminal.

We have 2 control flow choices for 2nd if terminal.

.....

.....

.....

We have 2 control flow choices for 10th if terminal.

By using multiplicative law of counting we get,

Total choices as $2 * 2 * 2 * 2 * 2 10 \text{ times} = 2^{10} = 1024$

Once again, **one need not know "what control flow"** is, but needs to know **"how many"**

"control flows" are in if structure which is given in question.

133 votes

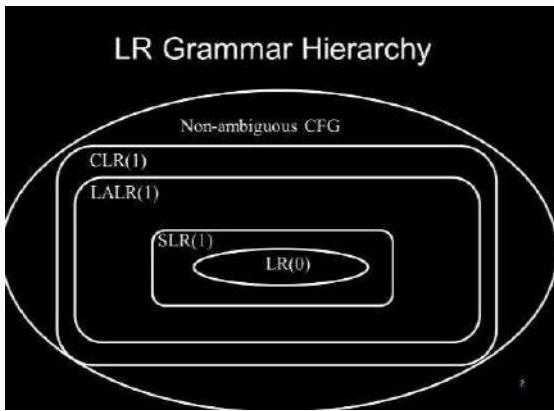
-- prs11 (255 points)

2.14.39 Parsing: GATE2017-2-6 top

<https://gateoverflow.in/118343>



Selected Answer



so only 1st one is correct ans

16 votes

-- 2018 (6.7k points)

2.14.40 Parsing: GATE2018-38 top

<https://gateoverflow.in/204112>



Selected Answer

\$ is the highest precedence with left associative(**since evaluated first**) and # is right associative.

A is answer

11 votes

-- Prashant Singh (59.8k points)

2.14.41 Parsing: TIFR2012-B-17 top

<https://gateoverflow.in/25215>



Selected Answer

- a. Does not make any sense. false.
- b. This is definition of $LR(k)$ Parser. True
- c. False. $LR(k)$ is subset of CFL.
- d. False.
- e. $LR(k)$, bottom up parser . We have Right most derivation. This is False.

Answer :- B

13 votes

-- Akash Kanase (42.5k points)

2.14.42 Parsing: TIFR2012-B-8 top

<https://gateoverflow.in/25108>



Selected Answer

e is correct

Because as the expression evaluated right to left , so in (ii). $2 + (a - (b * (a + b)))$ this evaluation performed, which is not a correct evaluation as the parse tree

13 votes

-- srestha (87k points)

2.14.43 Parsing: TIFR2015-B-15 top

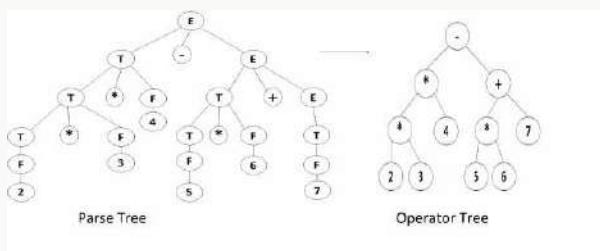
<https://gateoverflow.in/30079>



Selected Answer

Answer is option **B**.

The corresponding parse tree is drawn for the given expression according to the given grammar.



13 votes

-- Riya Roy(Arayana) (7.2k points)

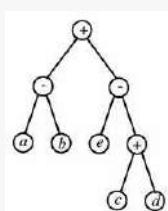
2.15

Register Allocation(2) top

2.15.1 Register Allocation: GATE2011-36 top

<https://gateoverflow.in/2138>

Consider evaluating the following expression tree on a machine with load-store architecture in which memory can be accessed only through load and store instructions. The variables a, b, c, d , and e are initially stored in memory. The binary operators used in this expression tree can be evaluated by the machine only when operands are in registers. The instructions produce result only in a register. If no intermediate results can be stored in memory, what is the minimum number of registers needed to evaluate this expression?



- A. 2
- B. 9
- C. 5
- D. 3

gate2011 compiler-design register-allocation normal

Answer

2.15.2 Register Allocation: GATE2017-1-52 top

<https://gateoverflow.in/118746>

Consider the expression $(a - 1) * (((b + c)/3) + d)$. Let X be the minimum number of registers required by an *optimal* code generation (without any register spill) algorithm for a load/store architecture, in which

- A. only load and store instructions can have memory operands and
 B. arithmetic instructions can have only register or immediate operands. The value of X is _____.

gate2017-1 compiler-design register-allocation normal numerical-answers

Answer

Answers: Register Allocation

2.15.1 Register Allocation: GATE2011-36 [top](#)

<https://gateoverflow.in/2138>



Given is Load Store Architecture, that means we can access memory using Load and Store Instructions.

Key Idea:- Pick new register only when it is required.

We want to add **c** and **d**, and initially both are in memory, therefore copy these into registers.

- load $R1, c$ ($R1 \leftarrow c$)
- load $R2, d$ ($R2 \leftarrow d$)

(here no compensation can be done, we need two registers)

- add $R1, R1, R2$ ($R1 \leftarrow R1 + R2$)

(at this point $R1$ is holding $c + d$ and $R2$ is holding d , i.e. $R1 \leftarrow c + d$ and $R2 \leftarrow d$)

Now, **e** comes into picture and my question is, Can i make use of $R1$ or $R2$ to store **e**?
 I can not use $R1$ to store **e** as its value will be needed later but I can use $R2$.

- load $R2, e$

(currently $R1 \leftarrow c + d$ and $R2 \leftarrow e$)

- Sub $R1, R2, R1$ ($R1 \leftarrow R2 - R1$)

Doing this all gives, final value of right sub-tree is stored in $R1$, and $R2$ stores **e**.

Now, coming to left subtree, to perform " $a - b$ " we need to copy both variables in registers.

We can copy one of the variable in $R2$, but we can not obviously copy in $R1$ as value of $R1$ will be required later.

- Load $R2, a$
- Load $R3, b$ (**here comes extra register, and we can not avoid using it.**)

Current mapping is $R2 \leftarrow a$, $R3 \leftarrow b$ and $R1$ contains final value of Right subtree.

- SUB $R2, R2, R3$ ($R2 \leftarrow R2 - R3$)
- ADD $R1, R1, R2$

Hence answer is 3 i.e. **D**

39 votes

-- Sachin Mittal (15.7k points)

2.15.2 Register Allocation: GATE2017-1-52 [top](#)

<https://gateoverflow.in/118746>

Selected Answer

Load $R1, b$ Load $R2, c$ ADD $R1, R2$ Div $R1, 3$ Load $R2, d$ Add $R1, R2$ Load $R2, a$ Sub $R2, 1$ ul $R2, R1$

hence minimum 2 registers required

45 votes

-- sriv_shubham (3.3k points)

2.16

Runtime Environments(18) [top](#)

2.16.1 Runtime Environments: CMI2011-A-08 [top](#)

<https://gateoverflow.in/46975>

In programming languages like C, C++, Python . . . the memory used by a program is typically separated into two parts, the stack and the heap. Consider the following statements:

1. A stack is efficient for managing nested function calls.
2. Stack space is limited while heap space is not.
3. The stack cannot be used for persistent data structures.

Then:

- A. 1 and 2 are true but 3 is false.
- B. 1 and 3 are true but 2 is false.
- C. 2 and 3 are true but 1 is false.
- D. All three statements are true.

cmi2011 compiler-design runtime-environments

Answer

2.16.2 Runtime Environments: CMI2014-A-06 [top](#)

<https://gateoverflow.in/46972>

Suppose we are working with a programming language that supports automatic garbage collection. This means that:

- A. Uninitialized variables are assigned null values.
- B. Unreferenced dynamically allocated memory is added back to free space.

- C. Unreachable *if – then – else* branches are pruned.
 D. Expressions where array indices exceed array bounds are flagged.

cmi2014 compiler-design runtime-environments

Answer

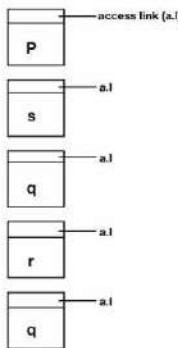
2.16.3 Runtime Environments: GATE1988-2xii [top](#)

<https://gateoverflow.in/93966>

Consider the following program skeleton and below figure which shows activation records of procedures involved in the calling sequence.

$$P \rightarrow s \rightarrow q \rightarrow r \rightarrow q.$$

Write the access links of the activation records to enable correct access and variables in the procedures from other procedures involved in the calling sequence



```

procedure P;
procedure q;
procedure r;
begin
  q
end r;
begin
r
end q;
procedure s;
begin
  q
end s;
begin
  s
end p;
  
```

gate1988 normal descriptive runtime-environments compiler-design

Answer

2.16.4 Runtime Environments: GATE1990-2-v [top](#)

<https://gateoverflow.in/83980>

Match the pairs in the following questions:

- (a) Pointer data type (p) Type conversion
 (b) Activation record (q) Dynamic data structure
 (c) Repeat-until (r) Recursion
 (d) Coercion (s) Nondeterministic loop

[gate1990](#) [match-the-following](#) [compiler-design](#) [runtime-environments](#) [recursion](#)

Answer

2.16.5 Runtime Environments: GATE1990-4-v [top](#)

<https://gateoverflow.in/85394>

State whether the following statements are TRUE or FALSE with reason:

The Link-load-and-go loading scheme required less storage space than the link-and-go loading scheme.

[gate1990](#) [true-false](#) [compiler-design](#) [runtime-environments](#)

Answer

2.16.6 Runtime Environments: GATE1993-7.7 [top](#)

<https://gateoverflow.in/2295>

A part of the system software which under all circumstances must reside in the main memory is:

- A. text editor
- B. assembler
- C. linker
- D. loader
- E. none of the above

[gate1993](#) [compiler-design](#) [runtime-environments](#) [easy](#)

Answer

2.16.7 Runtime Environments: GATE1995-1.14 [top](#)

<https://gateoverflow.in/2601>

A linker is given object modules for a set of programs that were compiled separately. What information need to be included in an object module?

- A. Object code
- B. Relocation bits
- C. Names and locations of all external symbols defined in the object module
- D. Absolute addresses of internal symbols

[gate1995](#) [compiler-design](#) [runtime-environments](#) [normal](#)

Answer

2.16.8 Runtime Environments: GATE1997-1.10 [top](#)

<https://gateoverflow.in/2226>

Heap allocation is required for languages.

- A. that support recursion
- B. that support dynamic data structure
- C. that use dynamic scope rules

- D. None of the above

gate1997 compiler-design easy runtime-environments

[Answer](#)

2.16.9 Runtime Environments: GATE1997-1.8 [top](#)

<https://gateoverflow.in/2224>

A language L allows declaration of arrays whose sizes are not known during compilation. It is required to make efficient use of memory. Which one of the following is true?

- A. A compiler using static memory allocation can be written for L
- B. A compiler cannot be written for L ; an interpreter must be used
- C. A compiler using dynamic memory allocation can be written for L
- D. None of the above

gate1997 compiler-design easy runtime-environments

[Answer](#)

2.16.10 Runtime Environments: GATE1998-1.25, ISRO2008-41

<https://gateoverflow.in/1662>

In a resident – OS computer, which of the following systems must reside in the main memory under all situations?

- A. Assembler
- B. Linker
- C. Loader
- D. Compiler

gate1998 compiler-design runtime-environments normal isro2008

[Answer](#)

2.16.11 Runtime Environments: GATE1998-1.28 [top](#)

<https://gateoverflow.in/1665>

A linker reads four modules whose lengths are 200, 800, 600 and 500 words, respectively. If they are loaded in that order, what are the relocation constants?

- A. 0, 200, 500, 600
- B. 0, 200, 1000, 1600
- C. 200, 500, 600, 800
- D. 200, 700, 1300, 2100

gate1998 compiler-design runtime-environments normal

[Answer](#)

2.16.12 Runtime Environments: GATE1998-2.15 [top](#)

<https://gateoverflow.in/1687>

Faster access to non-local variables is achieved using an array of pointers to activation records called a

- A. stack
- B. heap
- C. display
- D. activation tree

[gate1998](#)
[programming](#)
[compiler-design](#)
[normal](#)
[runtime-environments](#)
Answer

2.16.13 Runtime Environments: GATE2001-1.17 [top](#)

<https://gateoverflow.in/710>

The process of assigning load addresses to the various parts of the program and adjusting the code and the data in the program to reflect the assigned addresses is called

- A. Assembly
- B. parsing
- C. Relocation
- D. Symbol resolution

[gate2001](#)
[compiler-design](#)
[runtime-environments](#)
[easy](#)
Answer

2.16.14 Runtime Environments: GATE2008-54 [top](#)

<https://gateoverflow.in/477>

Which of the following are true?

- I. A programming language which does not permit global variables of any kind and has no nesting of procedures/functions, but permits recursion can be implemented with static storage allocation
 - II. Multi-level access link (or display) arrangement is needed to arrange activation records only if the programming language being implemented has nesting of procedures/functions
 - III. Recursion in programming languages cannot be implemented with dynamic storage allocation
 - IV. Nesting procedures/functions and recursion require a dynamic heap allocation scheme and cannot be implemented with a stack-based allocation scheme for activation records
 - V. Programming languages which permit a function to return a function as its result cannot be implemented with a stack-based storage allocation scheme for activation records
- A. II and V only
 - B. I, III and IV only
 - C. I, II and V only
 - D. II, III and V only

[gate2008](#)
[compiler-design](#)
[difficult](#)
[runtime-environments](#)
Answer

2.16.15 Runtime Environments: GATE2010-14 [top](#)

<https://gateoverflow.in/2187>

Which languages necessarily need heap allocation in the runtime environment?

- A. Those that support recursion.
- B. Those that use dynamic scoping.
- C. Those that allow dynamic data structure.
- D. Those that use global variables.

[gate2010](#)
[compiler-design](#)
[easy](#)
[runtime-environments](#)
Answer

2.16.16 Runtime Environments: GATE2012-36 [top](#)

<https://gateoverflow.in/1758>

Consider the program given below, in a block-structured pseudo-language with lexical scoping and nesting of procedures permitted.

```
Program main;
  Var ...

  Procedure A1;
    Var ...
    Call A2;
  End A1

  Procedure A2;
    Var ...

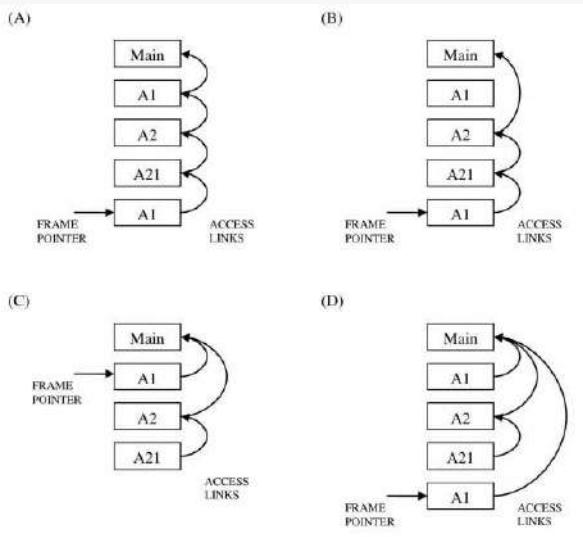
    Procedure A21;
      Var ...
      Call A1;
    End A21

    Call A21;
  End A2

  Call A1;
End main.
```

Consider the calling chain: *Main* → *A1* → *A2* → *A21* → *A1*

The correct set of activation records along with their access links is given by:



gate2012 compiler-design runtime-environments normal

Answer

2.16.17 Runtime Environments: GATE2014-2-18 top

<https://gateoverflow.in/1975>

Which one of the following is **NOT** performed during compilation?

- A. Dynamic memory allocation
- B. Type checking
- C. Symbol table management
- D. Inline expansion

gate2014-2 compiler-design easy runtime-environments

Answer**2.16.18 Runtime Environments: GATE2014-3-18** [top](#)<https://gateoverflow.in/2052>

Which of the following statements are CORRECT?

1. Static allocation of all data areas by a compiler makes it impossible to implement recursion.
 2. Automatic garbage collection is essential to implement recursion.
 3. Dynamic allocation of activation records is essential to implement recursion.
 4. Both heap and stack are essential to implement recursion.
- A. 1 and 2 only
 B. 2 and 3 only
 C. 3 and 4 only
 D. 1 and 3 only

[gate2014-3](#) [compiler-design](#) [runtime-environments](#) [normal](#)
Answer**Answers: Runtime Environments****2.16.1 Runtime Environments: CMI2011-A-08** [top](#)<https://gateoverflow.in/46195>

Selected Answer

Option (B) 1 and 3 are true but 2 is false.

2 is FALSE because **Size of heap and stack both are limited by the available main memory.** Usually the OS also restricts the maximum memory a process can use.

1. is TRUE because stack facilitates creation of memory for function calls and free them on function return - just a change of stack pointer is needed.
3. is TRUE because the stack variables are cleared on function return. Usually persistent data structures refer to those which survive program exits but even for static variables which should live until program finishes, stack cannot be used.

8 votes

-- Muktinath Vishwakarma (35.4k points)

2.16.2 Runtime Environments: CMI2014-A-06 [top](#)<https://gateoverflow.in/46972>

Selected Answer

Garbage collection also known as automatic memory management, is the automatic recycling of dynamically allocated memory.

Garbage collection is performed by a garbage collector which **recycles memory** that it can prove will **never be used again**.

Systems and languages which use garbage collection can be described as garbage-collected.

As long as address space is available in the memory, the runtime continues to allocate space for new objects.

However, memory is not infinite. Eventually the garbage collector must perform a collection in order to free some memory.

The garbage collector's optimizing engine determines the best time to perform a collection, based

upon the allocations being made.

So, option **B** is correct.

Unreferenced dynamically allocated memory is added back to free space.

1 upvotes

-- Manoj Kumar (38.6k points)

2.16.3 Runtime Environments: GATE1988-2xii top

<https://gateoverflow.in/93966>



Selected Answer

An activation record has the following parts:-

1. A **control link** from record A points to the previous record on the stack. The chain of control links traces the dynamic execution of the program.
2. An **access link** from record A points to the record of the closest enclosing block in the program. The chain of access links traces the static structure (think: scopes) of the program.

Going by the definition of the access link we have the record of the closest enclosing block in the program as mentioned.

The access link for the above ques is as follows:

$$p \rightarrow q \rightarrow r \rightarrow q \rightarrow s$$

1 upvotes

-- Don't defeat anybody but yourself. (12.3k points)

2.16.4 Runtime Environments: GATE1990-2-v top

<https://gateoverflow.in/83980>



Selected Answer

- | | |
|----------------------|---------------------------|
| A. Pointer data type | Q. Dynamic data structure |
| B. Activation record | R. Recursion |
| C. Repeat-until | S. Nondeterministic loop |
| D. Coercion | P. Type Conversion |

16 upvotes

-- Digvijay (54.9k points)

2.16.5 Runtime Environments: GATE1990-4-v top

<https://gateoverflow.in/85394>

TRUE.

In link and go scheme the linkage editor coexists with program in main memory while performing linking task

whereas link,load and go scheme the linkage editor does not coexists with program in main memory while performing linking task

source

: http://www.answers.com/Q/What_are_link_and_go_and_link_load_and_go_loader_schemes

3 upvotes

-- junk_mayavi (3.9k points)

2.16.6 Runtime Environments: GATE1993-7.7 [top](#)<https://gateoverflow.in/2295>

Selected Answer

Answer: d

The loader is a program that loads the object program from the secondary memory into the main memory for execution of the program. The loader resides in main memory.

18 votes

-- Rajarshi Sarkar (34.1k points)

2.16.7 Runtime Environments: GATE1995-1.14 [top](#)<https://gateoverflow.in/2601>

Selected Answer

- A. is trivially there is an object module.
- B. must be there if we need to have relocation capability.
- C. is the **answer**. For linker to link external symbols (for example in C, to link an extern variable in one module to a global variable in another module), it must know the location of all external symbols. In C external symbols includes all global variables and function names.
- D. is no way needed.

32 votes

-- Arjun Suresh (350k points)

2.16.8 Runtime Environments: GATE1997-1.10 [top](#)<https://gateoverflow.in/2226>

Selected Answer

Memory is taken from heap for dynamic allocation.

So, option (B) is correct.

15 votes

-- Sankaranarayanan P.N (11.5k points)

2.16.9 Runtime Environments: GATE1997-1.8 [top](#)<https://gateoverflow.in/2224>

Selected Answer

C.

Using dynamic memory allocation, memory will be allocated to array at runtime.

19 votes

-- Gate Keeda (19.6k points)

2.16.10 Runtime Environments: GATE1998-1.25, ISRO2008-41<https://gateoverflow.in/1662>**top**



C) is answer ,

In many operating systems the loader is permanently resident in memory, although some operating systems that support [virtual memory](#) may allow the loader to be located in a region of memory that is [pageable](#).

reference @[http://en.wikipedia.org/wiki/Loader_\(computing\)](http://en.wikipedia.org/wiki/Loader_(computing))

👍 20 votes

-- Mithlesh Upadhyay (5.8k points)

2.16.11 Runtime Environments: GATE1998-1.28 top

<https://gateoverflow.in/1665>



answer - **B**

first module loaded starting at address 0. Size is 200. hence it will occupy first 200 address last address being 199. Second module will be present from 200 and so on.

👍 19 votes

-- Ankit Rokde (9k points)

2.16.12 Runtime Environments: GATE1998-2.15 top

<https://gateoverflow.in/1687>



it is **C**

properties of displays

1. Use a pointer array to store the activation records along the static chain.
2. Fast access for non-local but may be complicated to maintain.
3. Calling a subprogram in the same level – simply replace and restore.
4. Calling a subprogram in the higher level – add an entry and may need to save the old pointers.
5. Calling a subprogram in the lower level – shrink the pointer and restore it when the subprogram returns.

<http://users.dickinson.edu/~wahlst/356/ch10.pdf>

👍 24 votes

-- sumit kumar singh dixit (2.1k points)

2.16.13 Runtime Environments: GATE2001-1.17 top

<https://gateoverflow.in/710>



Relocation is the process of assigning load addresses to position-dependent code of a program and adjusting the code and data in the program to reflect the assigned addresses.

Hence [Option C](#) is Ans

Symbol resolution is the process of searching files and libraries to replace symbolic references or names of libraries with actual usable addresses in memory before running a program.

23 votes

-- Rajesh Pradhan (22.6k points)

2.16.14 Runtime Environments: GATE2008-54 [top](#)

<https://gateoverflow.in/477>



Selected Answer

- I. False. Recursion cannot be implemented using static allocation.
- II. True. Yes, we do need multi level access link in case of nested functions. Each level to traverse ARB of same level of nesting.
- III. False. Recursion can only be implemented using dynamic memory allocation.
- IV. False. Recursion is done using memory in stack (ARBs in stack), not in heap.
- V. True. Yes, they cannot, once a function returns its activation record is no longer valid, so we cannot return a function as a result.

So, option (A) is correct.

14 votes

-- Monanshi Jain (9.3k points)

2.16.15 Runtime Environments: GATE2010-14 [top](#)

<https://gateoverflow.in/2187>



Selected Answer

Those that allow dynamic data structure.

malloc etc uses memory from heap area.

23 votes

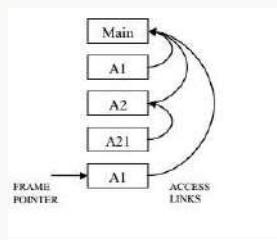
-- Sankaranarayanan P.N (11.5k points)

2.16.16 Runtime Environments: GATE2012-36 [top](#)

<https://gateoverflow.in/1758>



Selected Answer



Since, Activation records are created at procedure entry time and destroyed at procedure exit time. therefore here Calling sequence is given as ,

Main → A1 → A2 → A21 → A1

now *A1* and *A2* are defined under *Main*...So *A1*, *A2* Access link are pointed to main
A21 is Defined under *A2* hence its Access link will point to *A2*.

40 votes

-- Kalpish Singhal (2.1k points)

2.16.17 Runtime Environments: GATE2014-2-18 [top](#)

<https://gateoverflow.in/1975>



Selected Answer

Dynamic means- at runtime. Dynamic memory allocation happens during the execution time and hence (A) is the answer.

27 votes

-- Arjun Suresh (350k points)

2.16.18 Runtime Environments: GATE2014-3-18 top <https://gateoverflow.in/2052>



It will be D.

option 2 is wrong because it is not necessary to have automatic garbage collection to implement recursion.

option 4 is wrong because it says that both are required to implement recursion, which is wrong. Either of them will suffice.

23 votes

-- Gate Keeda (19.6k points)

2.17

Static Single Assignment(2) top

2.17.1 Static Single Assignment: GATE2016-1-19 top <https://gateoverflow.in/39675>

Consider the following code segment.

```
x = u - t;
y = x * v;
x = y + w;
y = t - z;
y = x * y;
```

The minimum number of *total* variables required to convert the above code segment to *static single assignment* form is _____.

[gate2016-1](#) [compiler-design](#) [static-single-assignment](#) [normal](#) [numerical-answers](#)

Answer

2.17.2 Static Single Assignment: GATE2017-1-12 top <https://gateoverflow.in/118292>

Consider the following intermediate program in three address code

```
p = a - b
q = p * c
p = u * v
q = p + q
```

Which one of the following corresponds to a *static single assignment* form of the above code?

- A. $p_1 = a - b$
 $q_1 = p_1 * c$
 $p_1 = u * v$
 $q_1 = p_1 + q_1$

- B. $p_3 = a - b$

```

q4 = p3 * c
p4 = u * v
q5 = p4 + q

```

C.

```

p1 = a - b
q1 = p2 * c
p3 = u * v
q2 = p4 + q3

```

D.

```

p1 = a - b
q1 = p * c
p2 = u * v
q2 = p + q

```

[gate2017-1](#) [compiler-design](#) [intermediate-code](#) [normal](#) [static-single-assignment](#)

Answer

Answers: Static Single Assignment

2.17.1 Static Single Assignment: GATE2016-1-19 [top](#)

<https://gateoverflow.in/39675>



Selected Answer

In Static Single Assignment when we assign the values, the variables to which the value is being assigned should unique.

$$T1 = u - t$$

$$T2 = T1 * v$$

$$T3 = T2 + w$$

$$T4 = t - z$$

$$T5 = t3 * t4$$

$$\text{So } T1.....T5 = 5 + (u, t, v, w, z) = 5$$

Total **10** variables.

Note: RHS of the operation can use the previously used variables, but LHS in SSA must always be unique.

21 votes

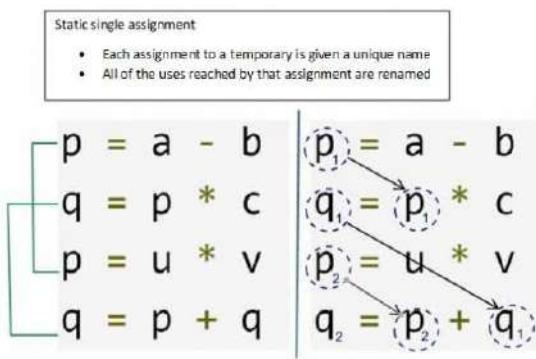
-- Prateek Khade (1.3k points)

2.17.2 Static Single Assignment: GATE2017-1-12 [top](#)

<https://gateoverflow.in/118292>



Selected Answer



https://en.wikipedia.org/wiki/Static_single_assignment_form

<http://www.cse.iitd.ernet.in/~nvkrishna/courses/winter07/ssa.pdf>

<https://www.cs.cmu.edu/~fp/courses/15411-f08/lectures/09-ssa.pdf>

<http://www.seas.harvard.edu/courses/cs252/2011sp/slides/Lec04-SSA.pdf>

so **B** is ans.

20 votes

-- 2018 (6.7k points)

2.18

Syntax Directed Translation(6) top

2.18.1 Syntax Directed Translation: GATE1995-2.10 top

A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar

- $S \rightarrow xxW\{\text{print} "1"\}$
- $S \rightarrow y\{\text{print} "2"\}$
- $W \rightarrow Sz\{\text{print} "3"\}$

What is the translation of $xxxxyzz$ using the syntax directed translation scheme described by the above rules?

- 23131
- 11233
- 11231
- 33211

[gate1995](#) [compiler-design](#) [grammar](#) [syntax-directed-translation](#) [normal](#)

Answer

2.18.2 Syntax Directed Translation: GATE1996-20 top <https://gateoverflow.in/2772>

Consider the syntax-directed translation schema (SDTS) shown below:

- $E \rightarrow E + E \{\text{print} "+"\}$
- $E \rightarrow E * E \{\text{print} "\cdot"\}$
- $E \rightarrow id \{\text{print} \text{id.name}\}$
- $E \rightarrow (E)$

An LR-parser executes the actions associated with the productions immediately after a reduction by

the corresponding production. Draw the parse tree and write the translation for the sentence.

$(a+b)*(c+d)$, using SDTS given above.

gate1996 compiler-design syntax-directed-translation normal

Answer

2.18.3 Syntax Directed Translation: GATE1998-23 [top](#) <https://gateoverflow.in/1738>

Let the attribute '*val*' give the value of a binary number generated by *S* in the following grammar:

$$\begin{aligned} S &\rightarrow L.L \mid L \\ L &\rightarrow LB \mid B \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

For example, an input 101.101 gives $S.\text{val} = 5.625$

Construct a syntax directed translation scheme using only synthesized attributes, to determine $S.\text{val}$.

gate1998 compiler-design syntax-directed-translation normal

Answer

2.18.4 Syntax Directed Translation: GATE2000-19 [top](#) <https://gateoverflow.in/690>

Consider the syntax directed translation scheme (SDTS) given in the following. Assume attribute evaluation with bottom-up parsing, i.e., attributes are evaluated immediately after a reduction.

$$E \rightarrow E_1 * T \{E.\text{val} = E_1.\text{val} * T.\text{val}\}$$

$$E \rightarrow T \{E.\text{val} = T.\text{val}\}$$

$$T \rightarrow F - T_1 \{T.\text{val} = F.\text{val} - T_1.\text{val}\}$$

$$T \rightarrow F \{T.\text{val} = F.\text{val}\}$$

$$F \rightarrow 2 \{F.\text{val} = 2\}$$

$$F \rightarrow 4 \{F.\text{val} = 4\}$$

- A. Using this SDTS, construct a parse tree for the expression $4 - 2 - 4 * 2$ and also compute its $E.\text{val}$.
- B. It is required to compute the total number of reductions performed to parse a given input. Using synthesized attributes only, modify the SDTS given, without changing the grammar, to find $E.\text{red}$, the number of reductions performed while reducing an input to *E*.

gate2000 compiler-design syntax-directed-translation normal descriptive

Answer

2.18.5 Syntax Directed Translation: GATE2003-18 [top](#) <https://gateoverflow.in/908>

In a bottom-up evaluation of a syntax directed definition, inherited attributes can

- A. always be evaluated
- B. be evaluated only if the definition is L-attributed
- C. be evaluated only if the definition has synthesized attributes
- D. never be evaluated

[gate2003](#) [compiler-design](#) [syntax-directed-translation](#) [normal](#)
[Answer](#)

2.18.6 Syntax Directed Translation: GATE2016-1-46 [top](#)

<https://gateoverflow.in/37700>
Consider the following Syntax Directed Translation Scheme (*SDTS*), with non-terminals $\{S, A\}$ and terminals $\{a, b\}$.

$$S \rightarrow aA \quad \{\text{print 1}\}$$

$$S \rightarrow a \quad \{\text{print 2}\}$$

$$A \rightarrow Sb \quad \{\text{print 3}\}$$

Using the above *SDTS*, the output printed by a bottom-up parser, for the input *aab* is:

- A. 1 3 2
- B. 2 2 3
- C. 2 3 1
- D. syntax error

[gate2016-1](#) [compiler-design](#) [syntax-directed-translation](#) [normal](#)
[Answer](#)

Answers: Syntax Directed Translation

2.18.1 Syntax Directed Translation: GATE1995-2.10 [top](#)


[Selected Answer](#)
<https://gateoverflow.in/2622>
A.

aking a tree and performing post order traversal will yield answer as A.

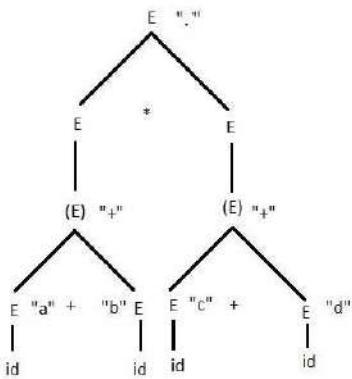
$$\begin{aligned} S &\rightarrow xxW(Pf'1') \\ W &\rightarrow Sz(Pf'3') \\ S &\rightarrow xxW(Pf'1') \\ W &\rightarrow Sz(Pf'3') \\ S &\rightarrow y(Pf'2') \end{aligned}$$

18 votes
-- Gate Keeda (19.6k points)

2.18.2 Syntax Directed Translation: GATE1996-20 [top](#) <https://gateoverflow.in/2772>


[Selected Answer](#)

ab + cd +.



19 votes

-- jayendra (8.2k points)

2.18.3 Syntax Directed Translation: GATE1998-23 top <https://gateoverflow.in/1738>



Selected Answer

$S \rightarrow L \cdot L \quad \{ S.dv = L_1.dv + L_2.dv / 2^{L_2.nb} \}$

| $L \quad \{ S.dv = L.dv \}$

$L \rightarrow LB \quad \{ L.dv = 2 * L_1.dv + B.dv$

$L.nb = L_1.nb + B.nb \}$

| $B \quad \{ L.dv = B.dv$

$L.nb = B.nb \}$

$B \rightarrow 0 \quad \{ B.dv = 0$

$B.nb = 1 \}$

| $1 \quad \{ B.dv = 1$

$B.nb = 1 \}$

here dv = decimal value

nb = number of bits.

18 votes

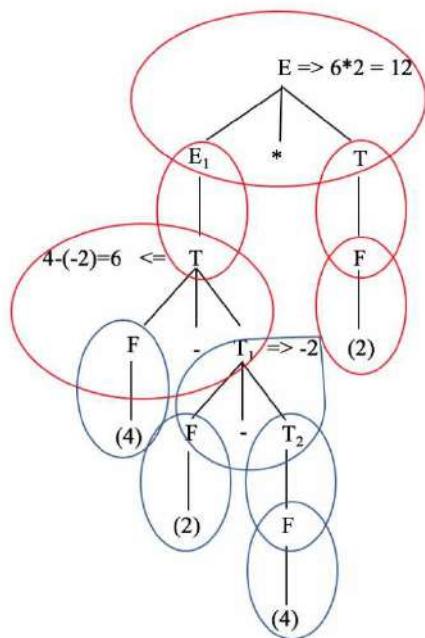
-- Gate Keeda (19.6k points)

2.18.4 Syntax Directed Translation: GATE2000-19 top <https://gateoverflow.in/690>



Selected Answer

Given expression $4 - 2 - 4 * 2$

**Total reduction****= 10**

- a. expression value =12
- b. a total number of reductions performed =10

17 votes

-- Prateek kumar (7.7k points)

2.18.5 Syntax Directed Translation: GATE2003-18 top https://gateoverflow.in/908



Selected Answer

- A. is false. If the grammar is not L-attributed; we cannot evaluate the inherited attributes in a bottom-up parse. In fact even for some L-attributed grammar, bottom-up parse is not possible for inherited attributes.
<http://infolab.stanford.edu/~ullman/dragon/slides2.pdf>
https://gateoverflow.in/?qa=blob&qa_blobid=14587629398289520039
- B. is true. Is there any non L-attributed grammar which can be parsed by a bottom-up parser? No, as shown in the above link. In fact only for the L-attributed grammar made from a LL(1) grammar, we can always guarantee a bottom-up parsing. Even for LR(1) grammar, bottom-up parsing is not a guarantee for all inherited attributes.
- C. is false. Some L-attributed grammars (including those with non-synthesized attributes) can be evaluated by a bottom-up parser.
- D. is false for above-told reasons.

A nice PDF for the same :- <https://acm.sjtu.edu.cn/w/images/a/a1/Compiler2013-lec07.pdf>

15 votes

-- Arjun Suresh (350k points)

2.18.6 Syntax Directed Translation: GATE2016-1-46 [top](#)



Selected Answer

<https://gateoverflow.in/39700>

aab could be derived as follows by the bottom up parser:

$$\begin{aligned} S &\rightarrow aA \text{ prints 1} \\ A &\rightarrow aSb \text{ prints 3} \\ A &\rightarrow aab \text{ prints 2} \end{aligned}$$

Now since bottom up parser will work in reverse of right most derivation, so it will print in bottom up fashion i.e., 231 which is option **C**.

Note that this could also be visualized easily by drawing the derivation tree.

41 votes

-- Monanshi Jain (9.3k points)

2.19

Target Code Generation(4) [top](#)

2.19.1 Target Code Generation: GATE1997-4.9 [top](#)

<https://gateoverflow.in/2250>

The expression $(a * b) * c \text{ op} \dots$

where 'op' is one of '+', '*' and ' \uparrow ' (exponentiation) can be evaluated on a CPU with single register without storing the value of $(a * b)$ if

- A. 'op' is '+' or '*'
- B. 'op' is ' \uparrow ' or '*'
- C. 'op' is ' \uparrow ' or '+'
- D. not possible to evaluate without storing

[gate1997](#) [compiler-design](#) [target-code-generation](#) [register-allocation](#) [normal](#)

Answer

2.19.2 Target Code Generation: GATE2003-59 [top](#)

<https://gateoverflow.in/947>

Consider the syntax directed definition shown below.

$$\begin{array}{ll} S \rightarrow \mathbf{id} := E & \{ \mathbf{gen}(\mathbf{id}.place = E.place); \} \\ E \rightarrow E_1 + E_2 & \{ t = \mathbf{newtemp}(); \\ & \mathbf{gen}(t = E_1.place + E_2.place); \\ & E.place = t; \} \\ E \rightarrow id & \{ E.place = \mathbf{id}.place; \} \end{array}$$

Here, \mathbf{gen} is a function that generates the output code, and $\mathbf{newtemp}$ is a function that returns the name of a new temporary variable on every call. Assume that t 's are the temporary variable names generated by $\mathbf{newtemp}$. For the statement ' $X := Y + Z$ ', the 3-address code sequence generated by this definition is

- A. $X = Y + Z$
- B. $t_1 = Y + Z; X = t_1$
- C. $t_1 = Y; t_2 = t_1 + Z; X = t_2$

- D. $t_1 = Y; t_2 = Z; t_3 = t_1 + t_2; X = t_3$

gate2003 compiler-design target-code-generation normal

[Answer](#)

2.19.3 Target Code Generation: GATE2004-10 [top](#)

<https://gateoverflow.in/4069>

Consider the grammar rule $E \rightarrow E1 - E2$ for arithmetic expressions. The code generated is targeted to a CPU having a single user register. The subtraction operation requires the first operand to be in the register. If $E1$ and $E2$ do not have any common sub expression, in order to get the shortest possible code

- A. $E1$ should be evaluated first
- B. $E2$ should be evaluated first
- C. Evaluation of $E1$ and $E2$ should necessarily be interleaved
- D. Order of evaluation of $E1$ and $E2$ is of no consequence

gate2004 compiler-design target-code-generation normal

[Answer](#)

2.19.4 Target Code Generation: GATE2010-37 [top](#)

<https://gateoverflow.in/2338>

The program below uses six temporary variables a, b, c, d, e, f .

```
a = 1
b = 10
c = 20
d = a + b
e = c + d
f = c + e
b = c + e
e = b + f
d = 5 + e
return d + f
```

Assuming that all operations take their operands from registers, what is the minimum number of registers needed to execute this program without spilling?

- A. 2
- B. 3
- C. 4
- D. 6

gate2010 compiler-design target-code-generation register-allocation normal

[Answer](#)

Answers: Target Code Generation

2.19.1 Target Code Generation: GATE1997-4.9 [top](#)

<https://gateoverflow.in/2250>



Selected Answer

A)

\uparrow has higher precedence than $\{*, +, -, /\}$

So, if $op = \uparrow$ implies, we need to evaluate the right hand side of \uparrow first and then do the lhs part, which would definitely require us to store the value of lhs

but if its a '+' or '*', we don't need to store the values evaluated, and on the go can do the operation directly on one register.

17 votes

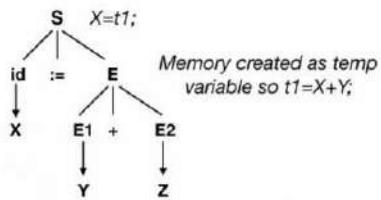
-- confused_luck (901 points)

2.19.2 Target Code Generation: GATE2003-59 top

<https://gateoverflow.in/947>



Selected Answer



Answer (B)

using given semantic rules.

18 votes

-- Prashant Singh (59.8k points)

2.19.3 Target Code Generation: GATE2004-10 top

<https://gateoverflow.in/4069>



Selected Answer

$E2$ should be evaluated first

After evaluating $E2$ first and then $E1$, we will have $E1$ in the register and thus we can simply do SUB operation with $E2$ which will be in memory (as we have only a single register). If we do $E1$ first and then $E2$, we must move $E2$ to memory and $E1$ back to register before doing SUB, which will increase the code size.

40 votes

-- Arjun Suresh (350k points)

2.19.4 Target Code Generation: GATE2010-37 top

<https://gateoverflow.in/2338>



Selected Answer

Here in these types of compiler questions, idea is "map/assign multiple temporaries to one registers."

here a, b , and c all are having 3 different values so i need atleast 3 registers $r1, r2$ and $r3$. a is mapped to $r1$, b to $r2$ and c to $r3$.

$d = a + b$, after this line if u notice ' a ' is never present on right hand side, so I can map (register of a which is $r1$) d to $r1$.

$e = c + d$, after this line ' d ' is never present on rhs, so I can map (register of d which is $r1$) e to $r1$.

at this time mapping is

$r1 --- e$
 $r2 --- b$
 $r3 --- c$

(at this moment I have registers for e, b and c . if I introduce new variable then I may need different register)

now at this point if u see

$f = c + e$
 $b = c + e$

these two are essentially doing same thing, after these two line ' b ' and ' f ' are same so I can skip computing ' f '. and wherever f is present I will replace it with ' b '. (because neither of ' f ' and ' b ' are changing after these two lines, so value of these will be ' $c + e$ ' forever)

(seems like I introduced one more variable f , and register is needed for that, but actually I did not really introduce ' f '. I am skipping computation of ' f ')

now at second last line " $d = 5 + e$ "

here I introduced ' d ', I can map it to any of the register $r1$ or $r3$, because after this line neither of ' e ' or ' c ' is required. (value of ' b ' is required because I need to return ' $d + f$ ', and ' f ' is essentially equal to ' b ')

finally code becomes

$r1 = 1$
 $r2 = 10$
 $r3 = 20$
 $r1 = r1 + r2$
 $r1 = r3 + r1$

(skipping ' f ' computation)

$r2 = r3 + r1$
 $r2 = r3 + r1$
 $r1 = r2 + r2$
 $r3 = 5 + r1$
return $r3 + r2$

Therefore minimum 3 registers needed.

👍 29 votes

-- Sachin Mittal (15.7k points)

2.20

Variable Scope(2) top

2.20.1 Variable Scope: GATE1987-1-xix top

<https://gateoverflow.in/80373>

Study the following program written in a block-structured language:

```
Var x, y:integer;
procedure P(n:integer);
begin
    x:=(n+2) / (n-3);
end;

procedure Q
Var x, y:integer;
begin
```

```

x:=3;
y:=4;
P(y);
Write(x) _____ (1)
end;

begin
  x:=7;
  y:=8;
  Q;
Write(x); _____ (2)
end.

```

What will be printed by the write statements marked (1) and (2) in the program if the variables are statically scoped?

- A. 3,6
- B. 6,7
- C. 3,7
- D. None of the above.

gate1987 compiler-design variable-scope

Answer

2.20.2 Variable Scope: GATE1987-1-XX top

<https://gateoverflow.in/80374>

For the program given below what will be printed by the write statements marked (1) and (2) in the program if the variables are dynamically scoped?

```

Var x, y:integer;
procedure P(n:integer);
begin
  x := (n+2) / (n-3);
end;

procedure Q
Var x, y:integer;
begin
  x:=3;
  y:=4;
  P(y);
  Write(x); _____ (1)
end;

begin
  x:=7;
  y:=8;
  Q;
  Write(x); _____ (2)
end.

```

- A. 3,6
- B. 6,7
- C. 3,7
- D. None of the above

gate1987 compiler-design variable-scope

Answer

Answers: Variable Scope

2.20.1 Variable Scope: GATE1987-1-xix top

<https://gateoverflow.in/80373>



Selected Answer

Using Static Scoping:

First, procedure **Q** is called from the **main** procedure. **Q** has local variables **x** and **y** with values **3** and **4** respectively. This local variable **y** (value **4**) is being passed to procedure **P** during call, and received in local variable **n** inside procedure **P**. Now, as **P** does not have any local definition for variable **x**, it will assign the evaluated value of $(n+2)/(n-3)$ i.e. $(4+2)/(4-3)=6$ to the global variable **x**, which was previously **7**. After the call of procedure **P**, procedure **Q** writes the value of local variable **x** which is still **3**. Lastly, the **main** procedure writes the value of global variable **x** which has been changed to **6** inside procedure **P**. So, the output will be **3, 6**.

Using Dynamic Scoping:

The same sequence of statements will be executed using dynamic scoping. However, as there is no local definition of variable **x** in procedure **P**, it will consider the recent definition in the calling sequence; as **P** is being called from procedure **Q**, definition of **x** from **Q** will be used, and value of **x** will be changed to **6** from **3**. Now, when **Q** writes local variable **x**, **6** will be printed. The write global variable **x** from **main** procedure will print **7** (as value of the global variable **x** has not been changed). So, the output will be **6, 7**.

1 3 votes

-- Sutanay Bhattacharjee (1.2k points)

2.20.2 Variable Scope: GATE1987-1-XX top

<https://gateoverflow.in/80374>



Selected Answer

Using Static Scoping:

First, procedure **Q** is called from the **main** procedure. **Q** has local variables **x** and **y** with values **3** and **4** respectively. This local variable **y** (value **4**) is being passed to procedure **P** during call, and received in local variable **n** inside procedure **P**. Now, as **P** does not have any local definition for variable **x**, it will assign the evaluated value of $(n+2)/(n-3)$ i.e. $(4+2)/(4-3)=6$ to the global variable **x**, which was previously **7**. After the call of procedure **P**, procedure **Q** writes the value of local variable **x** which is still **3**. Lastly, the **main** procedure writes the value of global variable **x** which has been changed to **6** inside procedure **P**. So, the output will be **3, 6**.

Using Dynamic Scoping:

The same sequence of statements will be executed using dynamic scoping. However, as there is no local definition of variable **x** in procedure **P**, it will consider the recent definition in the calling sequence; as **P** is being called from procedure **Q**, definition of **x** from **Q** will be used, and value of **x** will be changed to **6** from **3**. Now, when **Q** writes local variable **x**, **6** will be printed. The write global variable **x** from **main** procedure will print **7** (as value of the global variable **x** has not been changed). So, the output will be **6, 7**.

1 1 votes

-- Sutanay Bhattacharjee (1.2k points)

2.21

Viable Prefix(1) top

2.21.1 Viable Prefix: GATE2015-1-13 top

<https://gateoverflow.in/8187>

Which one of the following is TRUE at any valid state in shift-reduce parsing?

- A. Viable prefixes appear only at the bottom of the stack and not inside
- B. Viable prefixes appear only at the top of the stack and not inside
- C. The stack contains only a set of viable prefixes
- D. The stack never contains viable prefixes

gate2015-1 compiler-design parsing normal viable-prefix

Answer

Answers: Viable Prefix

2.21.1 Viable Prefix: GATE2015-1-13 top

<https://gateoverflow.in/8187>

 Selected Answer

Answer - C

Explanation:

A handle is actually the one which is always on the top of the stack. A viable prefix(prefix of the Right-hand side of a production or productions), is actually a prefix of the handle and so can never extend past the right end of the handle(i.e. the top of the stack).

The structure of the stack can be considered as a set of viable prefixes -

$Stack = \{Prefix_1 Prefix_2 Prefix_3 \dots Prefix_{n-1} Prefix_n\}$ and so it is not wrong to say that the stack contains a set of viable prefixes.

 40 votes -- Ravi Ranjan (3.2k points)

3 Programming & DS: DS (208)top

3.1 Abstract Data Type(1) top

3.1.1 Abstract Data Type: GATE2005-2 top

<https://gateoverflow.in/1344>

An Abstract Data Type (ADT) is:

- A. same as an abstract class
- B. a data type that cannot be instantiated
- C. a data type for which only the operations defined on it can be used, but none else
- D. all of the above

gate2005 data-structure normal abstract-data-type

[Answer](#)

Answers: Abstract Data Type

3.1.1 Abstract Data Type: GATE2005-2 top

<https://gateoverflow.in/1344>



Selected Answer

An abstract data type (ADT) supports only the operations which are defined.

Abstract class is one that may not have definitions of all the objects it have. Moreover it can not be instantiated. To instantiate we have to create a subclass then instantiate the class.

Abstract Data Type is like data structure eg. *STACK* where we have *PUSH()* *POP()* operation defined .

Hence, they are not the same thing.

<http://www.devx.com/tips/Tip/5681>

30 votes

-- Manali (2.6k points)

3.2 Arrays(14) top

3.2.1 Arrays: GATE1993-12 top

<https://gateoverflow.in/2309>

The following Pascal program segments finds the largest number in a two-dimensional integer array $A[0 \dots n-1, 0 \dots n-1]$ using a single loop. Fill up the boxes to complete the program and write against \boxed{A} , \boxed{B} , \boxed{C} and \boxed{D} in your answer book Assume that max is a variable to store the largest value and i, j are the indices to the array.

```
begin
  max:=|A|, i:=0, j:=0;
  while |B| do
  begin
    if A[i, j]>max then max:=A[i, j];
```

```

        if |C| then j:=j+1;
        else begin
            j:=0;
            i:=|D|
        end
    end
end

```

gate1993 data-structure arrays normal

Answer

3.2.2 Arrays: GATE1994-1.11 [top](#)

<https://gateoverflow.in/2452>

In a compact single dimensional array representation for lower triangular matrices (i.e all the elements above the diagonal are zero) of size $n \times n$, non-zero elements, (i.e elements of lower triangle) of each row are stored one after another, starting from the first row, the index of the $(i, j)^{th}$ element of the lower triangular matrix in this new representation is:

- A. $i + j$
- B. $i + j - 1$
- C. $(j - 1) + \frac{i(i-1)}{2}$
- D. $i + \frac{j(j-1)}{2}$

gate1994 data-structure arrays normal

Answer

3.2.3 Arrays: GATE1994-25 [top](#)

<https://gateoverflow.in/2521>

An array A contains n integers in non-decreasing order, $A[1] \leq A[2] \leq \dots \leq A[n]$. Describe, using Pascal like pseudo code, a linear time algorithm to find i, j , such that $A[i] + A[j] = a$ given integer M , if such i, j exist.

gate1994 data-structure arrays normal

Answer

3.2.4 Arrays: GATE1997-17 [top](#)

<https://gateoverflow.in/2277>

An array A contains $n \geq 1$ positive integers in the locations $A[1], A[2], \dots, A[n]$. The following program fragment prints the length of a shortest sequence of consecutive elements of A , $A[i], A[i+1], \dots, A[j]$ such that the sum of their values is $\geq M$, a given positive number. It prints ' $n+1$ ' if no such sequence exists. Complete the program by filling in the boxes. In each case use the simplest possible expression. Write only the line number and the contents of the box.

```

begin
i:=1;j:=1;
sum := □
min:=n; finish:=false;
while not finish do
    if □ then
        if j=n then finish:=true
        else
            begin
                j:=j+1;
                sum:= □
            end
    else
        begin

```

```

        if(j-i) < min then min:=j-i;
        sum:=sum -A[i];
        i:=i+1;
    end
    writeln (min +1);
end.

```

gate1997 data-structure arrays normal

Answer

3.2.5 Arrays: GATE1998-2.14 [top](#)

<https://gateoverflow.in/1686>

Let A be a two dimensional array declared as follows:

```
A: array [1 ... 10] [1 .... 15] of integer;
```

Assuming that each integer takes one memory location, the array is stored in row-major order and the first element of the array is stored at location 100, what is the address of the element $A[i][j]$?

- A. $15i + j + 84$
- B. $15j + i + 84$
- C. $10i + j + 89$
- D. $10j + i + 89$

gate1998 data-structure arrays easy

Answer

3.2.6 Arrays: GATE2000-1.2 [top](#)

<https://gateoverflow.in/625>

An $n \times n$ array v is defined as follows:

$$v[i, j] = i - j \text{ for all } i, j, i \leq n, 1 \leq j \leq n$$

The sum of the elements of the array v is

- A. 0
- B. $n - 1$
- C. $n^2 - 3n + 2$
- D. $n^2 \frac{(n+1)}{2}$

gate2000 data-structure arrays easy

Answer

3.2.7 Arrays: GATE2000-15 [top](#)

<https://gateoverflow.in/686>

Suppose you are given arrays $p[1.....N]$ and $q[1.....N]$ both uninitialized, that is, each location may contain an arbitrary value), and a variable count, initialized to 0. Consider the following procedures *set* and *is_set*:

```

set(i) {
    count = count + 1;
    q[count] = i;
    p[i] = count;
}

```

```

is_set(i) {
    if (p[i] ≤ 0 or p[i] > count)
        return false;
    if (q[p[i]] ≠ i)
        return false;
    return true;
}

```

- A. Suppose we make the following sequence of calls:

set(7); set(3); set(9);

After these sequence of calls, what is the value of count, and what do $q[1], q[2], q[3], p[7], p[3]$ and $p[9]$ contain?

- B. Complete the following statement "The first count elements of _____ contain values i such that $\text{set}(i)$ has been called".
- C. Show that if $\text{set}(i)$ has not been called for some i , then regardless of what $p[i]$ contains, $\text{is_set}(i)$ will return false.

gate2000 | data-structure | arrays | easy | descriptive

Answer

3.2.8 Arrays: GATE2005-5 top

<https://gateoverflow.in/1347>

A program P reads in 500 integers in the range $[0, 100]$ representing the scores of 500 students. It then prints the frequency of each score above 50. What would be the best way for P to store the frequencies?

- A. An array of 50 numbers
- B. An array of 100 numbers
- C. An array of 500 numbers
- D. A dynamically allocated array of 550 numbers

gate2005 | data-structure | arrays | easy

Answer

3.2.9 Arrays: GATE2013-50 top

<https://gateoverflow.in/1557>

The procedure given below is required to find and replace certain characters inside an input character string supplied in array A . The characters to be replaced are supplied in array $oldc$, while their respective replacement characters are supplied in array $newc$. Array A has a fixed length of five characters, while arrays $oldc$ and $newc$ contain three characters each. However, the procedure is flawed.

```

void find_and_replace (char *A, char *oldc, char *newc) {
    for (int i=0; i<5; i++)
        for (int j=0; j<3; j++)
            if (A[i] == oldc[j])
                A[i] = newc[j];
}

```

The procedure is tested with the following four test cases.

1. $oldc = "abc"$, $newc = "dab"$
2. $oldc = "cde"$, $newc = "bcd"$
3. $oldc = "bca"$, $newc = "cda"$
4. $oldc = "abc"$, $newc = "bac"$

The tester now tests the program on all input strings of length five consisting of characters '*a*', '*b*', '*c*', '*d*' and '*e*' with duplicates allowed. If the tester carries out this testing with the four test cases given above, how many test cases will be able to capture the flaw?

- A. Only one
- B. Only two
- C. Only three
- D. All four

gate2013 data-structure arrays normal

[Answer](#)

3.2.10 Arrays: GATE2013-51 [top](#)

<https://gateoverflow.in/43291>

The procedure given below is required to find and replace certain characters inside an input character string supplied in array *A*. The characters to be replaced are supplied in array *oldc*, while their respective replacement characters are supplied in array *newc*. Array *A* has a fixed length of five characters, while arrays *oldc* and *newc* contain three characters each. However, the procedure is flawed.

```
void find_and_replace (char *A, char *oldc, char *newc) {
    for (int i=0; i<5; i++)
        for (int j=0; j<3; j++)
            if (A[i] == oldc[j])
                A[i] = newc[j];
}
```

The procedure is tested with the following four test cases.

1. *oldc* = “abc”, *newc* = “dab”
2. *oldc* = “cde”, *newc* = “bcd”
3. *oldc* = “bca”, *newc* = “cda”
4. *oldc* = “abc”, *newc* = “bac”

If array *A* is made to hold the string “*abcde*”, which of the above four test cases will be successful in exposing the flaw in this procedure?

- A. None
- B. 2 only
- C. 3 and 4 only
- D. 4 only

gate2013 data-structure arrays normal

[Answer](#)

3.2.11 Arrays: GATE2014-3-42 [top](#)

<https://gateoverflow.in/2076>

Consider the C function given below. Assume that the array *listA* contains *n*(> 0) elements, sorted in ascending order.

```
int ProcessArray(int *listA, int x, int n)
{
    int i, j, k;
    i = 0; j = n-1;
    do {
        k = (i+j)/2;
        if (x <= listA[k]) j = k-1;
    }
```

```

        if (listA[k] <= x) i = k+1;
    }
    while (i <= j);
    if (listA[k] == x) return(k);
    else    return -1;
}

```

Which one of the following statements about the function *ProcessArray* is **CORRECT**?

- A. It will run into an infinite loop when x is not in *listA*.
- B. It is an implementation of binary search.
- C. It will always find the maximum element in *listA*.
- D. It will return -1 even when x is present in *listA*.

gate2014-3 data-structure arrays easy

Answer

3.2.12 Arrays: GATE2015-2-31 [top](#)

<https://gateoverflow.in/8148>

A Young tableau is a $2D$ array of integers increasing from left to right and from top to bottom. Any unfilled entries are marked with ∞ , and hence there cannot be any entry to the right of, or below a ∞ . The following Young tableau consists of unique entries.

1	2	5	14
3	4	6	23
10	12	18	25
31	∞	∞	∞

When an element is removed from a Young tableau, other elements should be moved into its place so that the resulting table is still a Young tableau (unfilled entries may be filled with a ∞). The minimum number of entries (other than 1) to be shifted, to remove 1 from the given Young tableau is _____.

gate2015-2 databases arrays normal numerical-answers

Answer

3.2.13 Arrays: ISI 2015 PCB C2 B [top](#)

<https://gateoverflow.in/120886>

You are given a array A of size n . You are told that A comprises three consecutive runs - first a run of a 's, then a run of b 's and finally a run of c 's. Moreover, you are provided an index of i such that $A[i] = b$. Design an $O(\log n)$ time algorithm to determine the number of b 's (i.e., length of the second run) in A .

data-structure arrays isi2015

Answer

3.2.14 Arrays: TIFR2011-B-30 [top](#)

<https://gateoverflow.in/20611>

Consider an array $A[1...n]$. It consists of a permutation of numbers $1....n$. Now compute another array $B[1...n]$ as follows: $B[A[i]] := i$ for all i . Which of the following is true?

- A. B will be a sorted array.
- B. B is a permutation of array A .
- C. Doing the same transformation twice will not give the same array.
- D. B is not a permutation of array A .
- E. None of the above.

tifr2011 data-structure arrays

Answer

Answers: Arrays

3.2.1 Arrays: GATE1993-12 [top](#)

<https://gateoverflow.in/2309>

Selected Answer

We have to traverse all elements in array. The code is doing this row wise.

```

begin
    max:=A[0,0], i:=0, j:=0;
    while (i < n) do
        begin
            if A[i, j]>max then max:=A[i, j];
            if (j < n-1) then j:=j+1;
            else begin
                j:=0;
                i:=i++;
            end
        end
    end
end

```

13 votes

-- Arjun Suresh (350k points)

3.2.2 Arrays: GATE1994-1.11 [top](#)

<https://gateoverflow.in/2452>

Selected Answer

$j - 1 + i(i - 1)/2$... because if you form a lower triangular matrix it contains elements in rows 1, 2, 3, ...

So, **C** is the correct answer.

PS: Though not mentioned in question, from options it is clear that **array index starts from 1 and not 0**.

Explanation :

In a lower triangular matrix, i^{th} row contains $(i + 1)$ number of non zero elements.

If we assume Array index starting from 1 then, i^{th} row contains i number of non zero elements.

before i^{th} row there are $i - 1$ rows (row 1 to $i - 1$), and in total these rows has $1 + 2 + 3 + \dots + (i - 1) = i((i - 1)/2)$ elements (row 1 has 1 element, row 2 has 2 elements, ..., row $i - 1$ has $i - 1$ elements etc.)

Now, at i^{th} row, before j^{th} element there are $(j - 1)$ elements (starting from $arr[i, 1]$ to $arr[i, j - 1]$).

Hence in total before $arr[i, j]$ there are $(i(i - 1)/2 + j - 1)$ elements and those elements will have indexes .

S, o the index of the $(i, j)^{th}$ element of the lower triangular matrix in this new representation is $(j - 1) + i(i - 1)/2$ which is option **C**.

32 votes

-- Bhagirathi Nayak (14.1k points)

3.2.3 Arrays: GATE1994-25 top

<https://gateoverflow.in/2521>



Selected Answer

```
i = 1;
j = n;
while(i != j) {
    if(A[i] + A[j] == M) break;
    else if(A[i] + A[j] < M) i++;
    else j--;
}
```

22 votes

-- Ankit Rokde (9k points)

3.2.4 Arrays: GATE1997-17 top

<https://gateoverflow.in/2277>



Selected Answer

```
begin
i:=1; j:=1;
sum := A[1]
min:=n; finish:=false;
while not finish do
    if sum < M then
        if j=n then finish:=true
        else
            begin
                j:=j+1;
                sum:= sum + A[j]
            end
    else
        begin
            if(j-i) < min then min:=j-i;
            sum:=sum - A[i];
            i:=i+1;
        end
    writeln (min +1);
end.
```

Algorithm

'*i*' indicates the starting marker and '*j*' acts as ending marker for the sum sequence. 'sum' is initialised as the first element in the array because the algorithm proceeds by taking the sum of remaining elements. 'finish' is a boolean variable that indicates exit from the loop.

After entering the loop for the first time with 'finish' as false, the sum is checked if it's strictly less than "*M*". If that's the case *j* is incremented and the sum is modified to sum + *A[j]*. When 'sum' becomes greater than or equal to '*M*', '*min*' is modified to the latest number of elements

that make the sum greater than or equal to ' M ' and then, the first element is stripped off from the sum and ' i ' is incremented by one to move the initial marker of the sum sequence. The loop runs till ' j ' reaches the end of the array.

The algorithm keeps track of ' min ' i.e. the number of elements in the minimum sum sequence. This is very similar to the way we find the minimum value in an array by modifying the min value whenever a lesser value is encountered.

12 votes

-- krish__ (4.3k points)

3.2.5 Arrays: GATE1998-2.14 top

<https://gateoverflow.in/1686>



Selected Answer

$$A[LB_1 \dots UB_1, LB_2 \dots UB_2]$$

BA = Base address.

C = size of each element.

Row major order.

$$Loc(a[i][j]) = BA + [(i - LB_1)(UB_2 - LB_2 + 1) + (j - LB_2)] * C .$$

Column Major order

$$Loc(a[i][j]) = BA + [(j - LB_2)(UB_1 - LB_1 + 1) + (i - LB_1)] * C .$$

Substituting the values.

Answer is **A**.

25 votes

-- Gate Keeda (19.6k points)

3.2.6 Arrays: GATE2000-1.2 top

<https://gateoverflow.in/625>



Selected Answer

The sum of the i^{th} row and i^{th} column is 0 as shown below. Since, the numbers of rows = no. of columns, the total sum will be 0.

0	-1	-2	-3	-4
1	0	-1	-2	-3
2	1	0	-1	-2
3	2	1	0	-1
4	3	2	1	0

28 votes

-- Arjun Suresh (350k points)

3.2.7 Arrays: GATE2000-15 top

<https://gateoverflow.in/686>



Selected Answer

a.

Initially $count = 0$;

When we call $set(7) - count = 1, q[1] = 7, p[7] = 1$;
 when we call $set(3) - count = 2, q[2] = 3, p[3] = 2$;
 when we call $set(9) - count = 3, q[3] = 9, p[9] = 3$;

b. Ans- "The first count elements of array q contain values i such that $set(i)$ has been called".

c. If $set(i)$ has not been called for some i , then regardless of what $p[i]$ contains, When we call $is_set(i)$ then

```
if (q[p[i]] != i)
    return false;
```

will always execute, because if $set(i)$ is not called then $p[i] \neq count$ (any) and for then same $count$ $q[count] \neq i$. So if statement will be true and will return false.

10 votes

-- Dhananjay Kumar Sharma (25.7k points)

3.2.8 Arrays: GATE2005-5 top

<https://gateoverflow.in/1347>



Selected Answer

As we our area of interest is only the 50 numbers so take An array of 50 numbers where $A[0]$ corresponds to 51... $A[49]$ corresponds to 100 then after reading an input just increment the counter in correct position as said above.

28 votes

-- Bhagirathi Nayak (14.1k points)

3.2.9 Arrays: GATE2013-50 top

<https://gateoverflow.in/1557>



Selected Answer

A	a	b	c	d	e
oldc	a	b	c		

oldc	a	b	c
newc	d	a	b

for (1)

Here, when the element of array A and $oldc$ match , we replace that array element of A with array element of $newc$. For every element of A array update occurs maximum one time.

Similarly for (2) array element of A has updated with array element of $newc$ less than or equal to one time,

A	a	b	c	d	e
oldc	b	c	a		

oldc	b	c	a
newc	c	d	a

for (2)

Now, for (3) when $i = 0$, value of A match with $oldc[2]$ i.e.'a' , and replace with $newc[2]$ i.e. also 'a'. So, no changes

When $i = 1$ value of array $A[1] = 'b'$

Match with $oldc[0] = 'b'$ and replace with $newc[0] = 'c'$.

Now, $A[1] = 'c'$ which equal with next element of $oldc[1] = 'c'$.

So, replace again with $newc[1] = 'd'$.

Now, we can say here in array $A[1]$ value replace with $newc[0]$ value , and that $newc[0]$ value replace with next $newc[1]$ value.

	$oldc$	$newc$
	a b c	b c
	for [4]	

Similarly for (4) here 2 times replacement for $A[0]$ with element $newc[0]$ and $newc[1]$

Updating of $newc$ value with another $newc$ value is calling flaw here

So, answer is **B**.

16 votes

-- srestha (87k points)

3.2.10 Arrays: GATE2013-51 top

<https://gateoverflow.in/43291>

	Selected Answer
$oldc$	
$newc$	

Here when the element of array A and $oldc$ match , we replace that array element of A with array element of $newc$. For every element of A array update occurs maximum one time.

Similarly for (2) array element of A has updated with array element of $newc$ less than or equal to one time,

	$oldc$	$newc$
	b c a	c d a
	for [3]	

Now, for (3) when $i = 0$, value of A match with $oldc[2]$ i.e.'a' , and replace with $newc[2]$ i.e. also 'a'. So, no changes

when $i = 1$ value of array $A[1] = 'b'$

match with $oldc[0] = 'b'$ and replace with $newc[0] = 'c'$.

Now, $A[1] = 'c'$ which equal with next element of $oldc[1] = 'c'$.

So, replace again with $newc[1] = 'd'$.

Now, we can say here in array $A[1]$ value replace with $newc[0]$ value , and that $newc[0]$ value

replace with next $newc[1]$ value.

x	x	c	d	e
a	b	c		
b	a	c		

for [4]

Similarly for (4) here 2 times replacement for $A[0]$ with element $newc[0]$ and $newc[1]$

Updating of $newc$ value with another $newc$ value is calling flaw here

So Ans (C)

17 votes

-- srestha (87k points)

3.2.11 Arrays: GATE2014-3-42 top

<https://gateoverflow.in/2076>



This is an implementation of the Binary search algorithm.

Note that the loop will be terminated when we have found x . In that case both the if conditions will be true making condition inside the while as false i.e., $i > j$.

9 votes

-- Monanshi Jain (9.3k points)

3.2.12 Arrays: GATE2015-2-31 top

<https://gateoverflow.in/8148>



The answer should be 5.

1. We first need to shift 2 in place of 1 keeping 5 AND 14 intact as it isn't mentioned in the question that the entire row elements move.
2. 4 is shifted up,next to 2 (keeping 12 and infinity intact in column 2).
3. Now in second row 6 is shifted left.
4. 18 shifts up to the second row
5. And finally 25 is shifted left to the third column.

So this takes 5 moves and still maintains the tableau property. Also infinity is placed to the right of 25 and below 23 (unfilled entries to be filled with ∞). The final table would look as follows.

2	4	5	14
3	6	18	23
10	12	25	∞
31	∞	∞	∞

39 votes

-- Aman verma (261 points)

3.2.13 Arrays: ISI 2015 PCB C2 B top

<https://gateoverflow.in/120886>



Hint: Modify Binary Search.

Solution:

You should use binary search twice first to find the first occurrence of b . Then apply it again to find the last occurrence.

The following code does this.

BinarySearchForFirst

```
int binarySearchForFirst(int l, int r) {
    int mid;
    while(l <= r) {
        mid = l+(r-l)/2;
        if(array[mid]=='b' && array[mid-1]=='a') { //the desired case
            return mid;
        }
        if(array[mid-1]=='b') { //we're to the right of required index
            r = mid;
            continue;
        }
        if(array[mid]=='a') { //we're to the left or required index
            l = mid;
            continue;
        }
    }
}
```

This returns the first index say i

Similarly find the second index j .

Then the answer is $j - i + 1$

11 votes

-- Akshay Arora (2.5k points)

3.2.14 Arrays: TIFR2011-B-30 top

<https://gateoverflow.in/20611>



Selected Answer

Option b)

B is a permutation of array A .

Infact, B gives the reverse index of all the elements of array A . Since the array A contains numbers $[1..n]$ mapped to the locations $[1..n]$ and A is a permutation of the numbers $[1..n]$, the array B will also be a permutation of the numbers $[1..n]$.

For example:

index	1	2	3	4	5	6	7	8
A	5	1	3	7	6	2	8	4
B	2	6	3	8	1	5	4	7

To see that option c is incorrect, let array C be the array attained from doing the same

transformation twice, that is, $C[B[i]] = i, \forall i \in [1..n]$. We get,

index	1	2	3	4	5	6	7	8
A	5	1	3	7	6	2	8	4
B	2	6	3	8	1	5	4	7
C	5	1	3	7	6	2	8	4

We can see that $C = A$, which makes option c incorrect.

20 votes

-- Pragy Agarwal (20.6k points)

3.3

Binary Search Tree(29) top

3.3.1 Binary Search Tree: GATE1996-2.14 top

<https://gateoverflow.in/2743>

A binary search tree is generated by inserting in order the following integers:

50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24

The number of nodes in the left subtree and right subtree of the root respectively is

- A. (4, 7)
- B. (7, 4)
- C. (8, 3)
- D. (3, 8)

[gate1996](#) [data-structure](#) [binary-search-tree](#) [normal](#)

Answer

3.3.2 Binary Search Tree: GATE1996-4 top

<https://gateoverflow.in/2756>

A binary search tree is used to locate the number 43. Which of the following probe sequences are possible and which are not? Explain.

- (a) 61 52 14 17 40 43
- (b) 2 3 50 40 60 43
- (c) 10 65 31 48 37 43
- (d) 81 61 52 14 41 43
- (e) 17 77 27 66 18 43

[gate1996](#) [data-structure](#) [binary-search-tree](#) [normal](#)

Answer

3.3.3 Binary Search Tree: GATE2001-14 top

<https://gateoverflow.in/755>

- A. Insert the following keys one by one into a binary search tree in the order specified.

15, 32, 20, 9, 3, 25, 12, 1

Show the final binary search tree after the insertions.

- B. Draw the binary search tree after deleting 15 from it.
- C. Complete the statements $S1$, $S2$ and $S3$ in the following function so that the function computes the depth of a binary tree rooted at t .

```
typedef struct tnode{
    int key;
    struct tnode *left, *right;
} *Tree;

int depth (Tree t)
{
    int x, y;
    if (t == NULL) return 0;
    x = depth (t -> left);
S1:   _____;
S2:   if (x > y) return _____;
S3:   else return _____;
}
```

gate2001 data-structure binary-search-tree normal descriptive

[Answer](#)

3.3.4 Binary Search Tree: GATE2003-19, ISRO2009-24 [top](#)

<https://gateoverflow.in/909>

Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?

- A. 7 5 1 0 3 2 4 6 8 9
- B. 0 2 4 3 1 6 5 9 8 7
- C. 0 1 2 3 4 5 6 7 8 9
- D. 9 8 6 4 2 3 0 1 5 7

gate2003 binary-search-tree easy isro2009

[Answer](#)

3.3.5 Binary Search Tree: GATE2003-6 [top](#)

<https://gateoverflow.in/897>

Let $T(n)$ be the number of different binary search trees on n distinct elements.

Then $T(n) = \sum_{k=1}^n T(k-1)T(x)$, where x is

- A. $n - k + 1$
- B. $n - k$
- C. $n - k - 1$
- D. $n - k - 2$

gate2003 normal binary-search-tree

[Answer](#)

3.3.6 Binary Search Tree: GATE2003-63, ISRO2009-25 [top](#)

<https://gateoverflow.in/950>

A data structure is required for storing a set of integers such that each of the following operations can be done in $O(\log n)$ time, where n is the number of elements in the set.

- I. Deletion of the smallest element
- II. Insertion of an element if it is not already present in the set

Which of the following data structures can be used for this purpose?

- A. A heap can be used but not a balanced binary search tree
- B. A balanced binary search tree can be used but not a heap
- C. Both balanced binary search tree and heap can be used
- D. Neither balanced search tree nor heap can be used

[gate2003](#) [data-structure](#) [easy](#) [isro2009](#) [binary-search-tree](#)

[Answer](#)

3.3.7 Binary Search Tree: GATE2004-4, ISRO2009-26 [top](#)

<https://gateoverflow.in/1001>

The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, 16. What is the height of the binary search tree (the height is the maximum distance of a leaf node from the root)?

- A. 2
- B. 3
- C. 4
- D. 6

[gate2004](#) [data-structure](#) [binary-search-tree](#) [easy](#) [isro2009](#)

[Answer](#)

3.3.8 Binary Search Tree: GATE2004-85 [top](#)

<https://gateoverflow.in/1079>

A program takes as input a balanced binary search tree with n leaf nodes and computes the value of a function $g(x)$ for each node x . If the cost of computing $g(x)$ is:

$$\min \left(\frac{\text{number of leaf-nodes}}{\text{in left-subtree of } x}, \frac{\text{number of leaf-nodes}}{\text{in right-subtree of } x} \right)$$

Then the worst-case time complexity of the program is?

- A. $\Theta(n)$
- B. $\Theta(n \log n)$
- C. $\Theta(n^2)$
- D. $\Theta(n^2 \log n)$

[gate2004](#) [binary-search-tree](#) [normal](#) [data-structure](#)

[Answer](#)

3.3.9 Binary Search Tree: GATE2005-IT-12 [top](#)

<https://gateoverflow.in/3757>

The numbers $1, 2, \dots, n$ are inserted in a binary search tree in some order. In the resulting tree, the right subtree of the root contains p nodes. The first number to be inserted in the tree must be

- A. p
- B. $p + 1$
- C. $n - p$
- D. $n - p + 1$

[gate2005-it](#) [data-structure](#) [normal](#) [binary-search-tree](#)

[Answer](#)

3.3.10 Binary Search Tree: GATE2005-IT-55 [top](#)

<https://gateoverflow.in/3816>

A binary search tree contains the numbers $1, 2, 3, 4, 5, 6, 7, 8$. When the tree is traversed in pre-order and the values in each node printed out, the sequence of values obtained is $5, 3, 1, 2, 4, 6, 8, 7$. If the tree is traversed in post-order, the sequence obtained would be

- A. $8, 7, 6, 5, 4, 3, 2, 1$
- B. $1, 2, 3, 4, 8, 7, 6, 5$
- C. $2, 1, 4, 3, 6, 7, 8, 5$
- D. $2, 1, 4, 3, 7, 8, 6, 5$

[gate2005-it](#) [data-structure](#) [binary-search-tree](#) [normal](#)

[Answer](#)

3.3.11 Binary Search Tree: GATE2006-IT-45 [top](#)

<https://gateoverflow.in/3588>

Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 55. Which of the following sequences CANNOT be the sequence of nodes examined?

- A. $\{10, 75, 64, 43, 60, 57, 55\}$
- B. $\{90, 12, 68, 34, 62, 45, 55\}$
- C. $\{9, 85, 47, 68, 43, 57, 55\}$
- D. $\{79, 14, 72, 56, 16, 53, 55\}$

[gate2006-it](#) [data-structure](#) [binary-search-tree](#) [normal](#)

[Answer](#)

3.3.12 Binary Search Tree: GATE2007-IT-29 [top](#)

<https://gateoverflow.in/3462>

When searching for the key value 60 in a binary search tree, nodes containing the key values $10, 20, 40, 50, 70, 80, 90$ are traversed, not necessarily in the order given. How many different orders are possible in which these key values can occur on the search path from the root to the node containing the value 60?

- A. 35
- B. 64
- C. 128
- D. 5040

[gate2007-it](#) [data-structure](#) [binary-search-tree](#) [normal](#)

Answer**3.3.13 Binary Search Tree: GATE2008-46** [top](#)<https://gateoverflow.in/458>

You are given the postorder traversal, P , of a binary search tree on the n elements $1, 2, \dots, n$. You have to determine the unique binary search tree that has P as its postorder traversal. What is the time complexity of the most efficient algorithm for doing this?

- A. $\Theta(\log n)$
- B. $\Theta(n)$
- C. $\Theta(n \log n)$
- D. None of the above, as the tree cannot be uniquely determined

[gate2008](#) [data-structure](#) [binary-search-tree](#) [normal](#)**Answer****3.3.14 Binary Search Tree: GATE2008-IT-12** [top](#)<https://gateoverflow.in/3272>

Which of the following is TRUE?

- A. The cost of searching an AVL tree is $\Theta(\log n)$ but that of a binary search tree is $O(n)$
- B. The cost of searching an AVL tree is $\Theta(\log n)$ but that of a complete binary tree is $\Theta(n \log n)$
- C. The cost of searching a binary search tree is $O(\log n)$ but that of an AVL tree is $\Theta(n)$
- D. The cost of searching an AVL tree is $\Theta(n \log n)$ but that of a binary search tree is $O(n)$

[gate2008-it](#) [data-structure](#) [binary-search-tree](#) [easy](#)**Answer****3.3.15 Binary Search Tree: GATE2008-IT-71** [top](#)<https://gateoverflow.in/3385>

A Binary Search Tree (BST) stores values in the range 37 to 573. Consider the following sequence of keys.

- I. 81, 537, 102, 439, 285, 376, 305
- II. 52, 97, 121, 195, 242, 381, 472
- III. 142, 248, 520, 386, 345, 270, 307
- IV. 550, 149, 507, 395, 463, 402, 270

Suppose the BST has been unsuccessfully searched for key 273. Which all of the above sequences list nodes in the order in which we could have encountered them in the search?

- A. II and III only
- B. I and III only
- C. III and IV only
- D. III only

[gate2008-it](#) [data-structure](#) [binary-search-tree](#) [normal](#)**Answer****3.3.16 Binary Search Tree: GATE2008-IT-72** [top](#)<https://gateoverflow.in/3386>

A Binary Search Tree (BST) stores values in the range 37 to 573. Consider the following sequence of

keys.

- I. 81, 537, 102, 439, 285, 376, 305
- II. 52, 97, 121, 195, 242, 381, 472
- III. 142, 248, 520, 386, 345, 270, 307
- IV. 550, 149, 507, 395, 463, 402, 270

Which of the following statements is TRUE?

- A. I, II and IV are inorder sequences of three different BSTs
- B. I is a preorder sequence of some BST with 439 as the root
- C. II is an inorder sequence of some BST where 121 is the root and 52 is a leaf
- D. IV is a postorder sequence of some BST with 149 as the root

[gate2008-it](#) [data-structure](#) [binary-search-tree](#) [easy](#)

[Answer](#)

3.3.17 Binary Search Tree: GATE2008-IT-73 [top](#)

<https://gateoverflow.in/3387>

How many distinct BSTs can be constructed with 3 distinct keys?

- A. 4
- B. 5
- C. 6
- D. 9

[gate2008-it](#) [data-structure](#) [binary-search-tree](#) [normal](#)

[Answer](#)

3.3.18 Binary Search Tree: GATE2009-37,ISRO-DEC2017-55 [top](#)

<https://gateoverflow.in/1323>

What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0.

- A. 2
- B. 3
- C. 4
- D. 5

[gate2009](#) [data-structure](#) [binary-search-tree](#) [normal](#) [isrodec2017](#)

[Answer](#)

3.3.19 Binary Search Tree: GATE2012-5 [top](#)

<https://gateoverflow.in/37>

The worst case running time to search for an element in a balanced binary search tree with $n2^n$ elements is

- A. $\Theta(n \log n)$
- B. $\Theta(n2^n)$
- C. $\Theta(n)$
- D. $\Theta(\log n)$

[gate2012](#) [data-structure](#) [normal](#) [binary-search-tree](#)

[Answer](#)

3.3.20 Binary Search Tree: GATE2013-43 [top](#)

<https://gateoverflow.in/1554>

The preorder traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42 . Which one of the following is the postorder traversal sequence of the same tree?

- A. 10, 20, 15, 23, 25, 35, 42, 39, 30
- B. 15, 10, 25, 23, 20, 42, 35, 39, 30
- C. 15, 20, 10, 23, 25, 42, 35, 39, 30
- D. 15, 10, 23, 25, 20, 35, 42, 39, 30

gate2013 data-structure binary-search-tree normal

[Answer](#)

3.3.21 Binary Search Tree: GATE2013-7 [top](#)

<https://gateoverflow.in/1416>

Which one of the following is the tightest upper bound that represents the time complexity of inserting an object into a binary search tree of n nodes?

- A. $O(1)$
- B. $O(\log n)$
- C. $O(n)$
- D. $O(n \log n)$

gate2013 data-structure easy binary-search-tree

[Answer](#)

3.3.22 Binary Search Tree: GATE2014-3-39 [top](#)

<https://gateoverflow.in/2073>

Suppose we have a balanced binary search tree T holding n numbers. We are given two numbers L and H and wish to sum up all the numbers in T that lie between L and H . Suppose there are m such numbers in T . If the tightest upper bound on the time to compute the sum is $O(n^a \log^b n + m^c \log^d n)$, the value of $a + 10b + 100c + 1000d$ is _____.

gate2014-3 data-structure binary-search-tree numerical-answers normal

[Answer](#)

3.3.23 Binary Search Tree: GATE2015-1-10 [top](#)

<https://gateoverflow.in/8129>

Which of the following is/are correct in order traversal sequence(s) of binary search tree(s)?

- I. 3, 5, 7, 8, 15, 19, 25
- II. 5, 8, 9, 12, 10, 15, 25
- III. 2, 7, 10, 8, 14, 16, 20
- IV. 4, 6, 7, 9, 18, 20, 25

- A. I and IV only
- B. II and III only
- C. II and IV only
- D. II only

gate2015-1 data-structure binary-search-tree easy

[Answer](#)

3.3.24 Binary Search Tree: GATE2015-1-23 [top](#)

<https://gateoverflow.in/8221>

What are the worst-case complexities of insertion and deletion of a key in a binary search tree?

- A. $\Theta(\log n)$ for both insertion and deletion
- B. $\Theta(n)$ for both insertion and deletion
- C. $\Theta(n)$ for insertion and $\Theta(\log n)$ for deletion
- D. $\Theta(\log n)$ for insertion and $\Theta(n)$ for deletion

gate2015-1 data-structure binary-search-tree easy

[Answer](#)

3.3.25 Binary Search Tree: GATE2015-3-13 [top](#)

<https://gateoverflow.in/8409>

While inserting the elements 71, 65, 84, 69, 67, 83 in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is

- A. 65
- B. 67
- C. 69
- D. 83

gate2015-3 data-structure binary-search-tree easy

[Answer](#)

3.3.26 Binary Search Tree: GATE2016-2-40 [top](#)

<https://gateoverflow.in/39586>

The number of ways in which the numbers 1, 2, 3, 4, 5, 6, 7 can be inserted in an empty binary search tree, such that the resulting tree has height 6, is _____.

Note: The height of a tree with a single node is 0.

gate2016-2 data-structure binary-search-tree normal numerical-answers

[Answer](#)

3.3.27 Binary Search Tree: GATE2017-1-6 [top](#)

<https://gateoverflow.in/118286>

Let T be a binary search tree with 15 nodes. The minimum and maximum possible heights of T are:

Note: The height of a tree with a single node is 0.

- A. 4 and 15 respectively.
- B. 3 and 14 respectively.
- C. 4 and 14 respectively.
- D. 3 and 15 respectively.

gate2017-1 data-structure binary-search-tree easy

[Answer](#)

3.3.28 Binary Search Tree: GATE2017-2-36 [top](#)

<https://gateoverflow.in/118378>

The pre-order traversal of a binary search tree is given by 12, 8, 6, 2, 7, 9, 10, 16, 15, 19, 17, 20 . Then the post-order traversal of this tree is

- A. 2, 6, 7, 8, 9, 10, 12, 15, 16, 17, 19, 20
- B. 2, 7, 6, 10, 9, 8, 15, 17, 20, 19, 16, 12
- C. 7, 2, 6, 8, 9, 10, 20, 17, 19, 15, 16, 12
- D. 7, 6, 2, 10, 9, 8, 15, 16, 17, 20, 19, 12

[gate2017-2](#)
[data-structure](#)
[binary-search-tree](#)
Answer

3.3.29 Binary Search Tree: TIFR2010-B-26 [top](#)

<https://gateoverflow.in/18749>

Suppose there is a balanced binary search tree with n nodes, where at each node, in addition to the key, we store the number of elements in the sub tree rooted at that node.

Now, given two elements a and b , such that $a < b$, we want to find the number of elements x in the tree that lie between a and b , that is, $a \leq x \leq b$. This can be done with (choose the best solution).

- A. $O(\log n)$ comparisons and $O(\log n)$ additions.
- B. $O(\log n)$ comparisons but no further additions.
- C. $O(\sqrt{n})$ comparisons but $O(\log n)$ additions.
- D. $O(\log n)$ comparisons but a constant number of additions.
- E. $O(n)$ comparisons and $O(n)$ additions, using depth-first- search.

[tifr2010](#)
[binary-search-tree](#)
Answer

Answers: Binary Search Tree

3.3.1 Binary Search Tree: GATE1996-2.14 [top](#)

<https://gateoverflow.in/2743>


Selected Answer

B.

Root will be 50. now insert one by one, greater to 50 in the right sub tree, lesser in left sub tree.

Or you can simply count the number looking at the i/p. less than 50 are 7. more than 50 are 4.

 21 votes
-- Gate Keeda (19.6k points)

3.3.2 Binary Search Tree: GATE1996-4 [top](#)

<https://gateoverflow.in/2756>


Selected Answer

(b) and **(e)** are not possible.

rest all i/p's will have binary trees with only one child. but (b) and (e) will have two childs at a point. therefore the probe sequence will not be possible.

For better clarification, make BST's for the given i/p's and probe for 43.

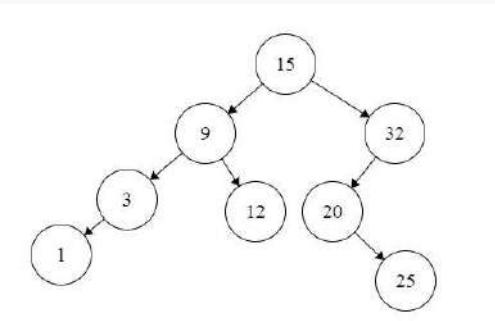
 23 votes
-- Gate Keeda (19.6k points)

3.3.3 Binary Search Tree: GATE2001-14 [top](#)

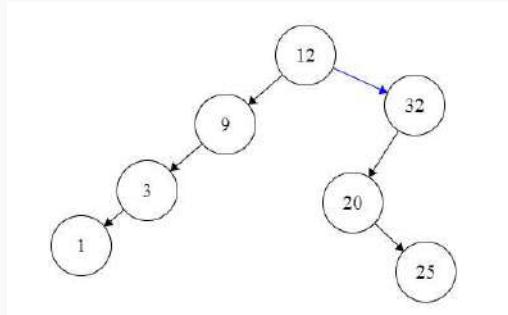
<https://gateoverflow.in/755>


Selected Answer

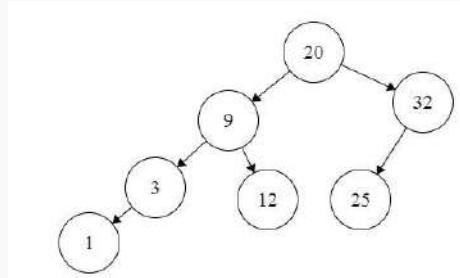
Binary search tree will be:



After delete of root: use inorder predecessor



Or After delete of root: use inorder successor



```

typedef struct tnode{
    int key;
    struct tnode *left, *right;
} *Tree;

int depth (Tree t)
{
    int x, y;
    if (t == NULL) return 0;
    x = depth (t -> left);
    S1: y = depth (t -> right);

    S2:     if (x > y) return (1+x); one for root
    S3:     else return (1+y); one for root
}
  
```

13 votes

-- Anu007 (17k points)

3.3.4 Binary Search Tree: GATE2003-19, ISRO2009-24 top



In-order traversal returns the elements in sorted order.

Therefore, it's option **C**

21 votes

-- Gate_15_isHere (583 points)

3.3.5 Binary Search Tree: GATE2003-6 top

<https://gateoverflow.in/897>



The summation is for each node, if that node happens to be the root. When a node is root, it will have $(k - 1)$ nodes on the left sub tree (k being any number) and correspondingly $(n - k)$ elements on the right sub tree. So, we can write recurrence $T(k - 1) * T(n - k)$ for the number of distinct binary search trees, as the numbers on left and right sub trees form BSTs independent of each other and only a difference in one of the sub trees produces a difference in the tree. Hence, answer is **B**.

Knowing the direct formula can also help in getting the answer but is not recommended.

http://gatetcse.in/wiki/Number_of_Binary_trees_possible_with_n_nodes

35 votes

-- Arjun Suresh (350k points)

3.3.6 Binary Search Tree: GATE2003-63, ISRO2009-25 top

<https://gateoverflow.in/950>



Balanced search tree have height $\log n$

Deletion of smallest element will take $O(\log n)$ time

Finding a element is present/not and doing insertion: $O(\log n)$

Heap(MIN) is also an almost complete binary tree have height $\log n$

Deletion of smallest element will take $O(\log n)$ time (root element removal, replace with last element +balancing)

Finding a element is present/not and insertion: Finding only takes $O(n)$, insertion then balancing take $O(\log n)$. So, total $O(n) + O(\log n) = O(n)$.

Answer is **B**.

(even if its maxheap our ans does not change only time for deletion of min will increase $O(n)$)

40 votes

-- Anurag Semwal (8k points)

3.3.7 Binary Search Tree: GATE2004-4, ISRO2009-26 top

<https://gateoverflow.in/1001>



Selected Answer

```

graph TD
    0010((0010)) --> 0001((0001))
    0010 --> 0015((0015))
    0001 --> 0003((0003))
    0001 --> 0005((0005))
    0015 --> 0012((0012))
    0015 --> 0016((0016))
  
```

Height is 3

22 votes -- Prashant Singh (59.8k points)

3.3.8 Binary Search Tree: GATE2004-85 top

<https://gateoverflow.in/1079>

Selected Answer

B. At the root node (first level) the cost would be $\Theta\left(\frac{n}{2}\right)$ as the tree is **balanced**.
At next level, we have 2 nodes and for each of them cost of computing $g(x)$ will be $\Theta\left(\frac{n}{4}\right)$. So, total cost at second level = $\Theta\left(\frac{n}{2}\right)$. Similarly at **each level** (total cost per level and not the cost per node in a level) the cost would be $\Theta\left(\frac{n}{2}\right)$ and so for $\log n$ levels it would be $\Theta(n \log n)$.
PS: Even if we change **min** to **max** in the definition of $g(x)$ we get the same answer.

43 votes -- Shaun Patel (6.9k points)

3.3.9 Binary Search Tree: GATE2005-IT-12 top

<https://gateoverflow.in/3757>

Selected Answer

Option is C.

```

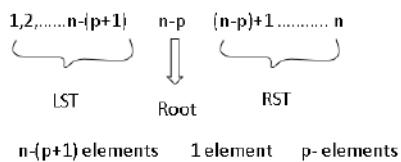
graph TD
    x((x)) --> LST1(( ))
    x --> RST1(( ))
    LST1 --- LST_label[LST]
    RST1 --- RST_label[RST]
  
```

P: # elements in RST
 \Rightarrow Depending on X, some number will go LST \notin RST
 \Rightarrow

```

graph TD
    x((x)) --> ?1(( ))
    x --> P_element1(( ))
    ?1 --- ?_label[?]
    P_element1 --- P_element_label[P-element]
    x --> 1_element_label[1 element]
  
```

Remaining elements for LST: $n - (p + 1)$



33 votes

-- pC (22k points)

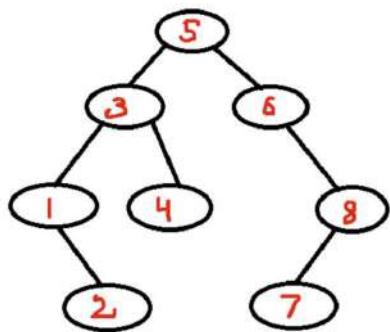
3.3.10 Binary Search Tree: GATE2005-IT-55 top

<https://gateoverflow.in/3816>



Selected Answer

Answer is D.



23 votes

-- Gate Keeda (19.6k points)

3.3.11 Binary Search Tree: GATE2006-IT-45 top

<https://gateoverflow.in/3588>



Selected Answer

In option C search sequence progress in ...47, 68, 43,..

At 47 we see that search key 55 is greater and it will be on right side of 47. so in further comparison a value less than 47 will not come

Hence, option C is wrong.

25 votes

-- Sankaranarayanan P.N (11.5k points)

3.3.12 Binary Search Tree: GATE2007-IT-29 top

<https://gateoverflow.in/3462>



Selected Answer

10, 20, 40, 50, 70, 80, 90

In BST search we if we go from say 10 to 40 while searching for 60, we will never encounter 20. So, 10, 20, 40 and 50 visited, means they are visited in order. Similarly, 90, 80 and 70 are visited in order. So, our required answer will be

No. of possible permutations of 7 numbers
No. of possible permutations of numbers smaller than 60 × No. of possible permutations of numbers larger than 60

(Since only one permutation is valid for both the smaller set of numbers as well as larger set of numbers)

$$= \frac{7!}{4!3!}$$

$$= 35$$

74 votes

-- Arjun Suresh (350k points)

3.3.13 Binary Search Tree: GATE2008-46 top

<https://gateoverflow.in/458>



Last element in post order is the root of tree- find this element in inorder- $\log n$ time.
 Now as in quick sort consider this as pivot and split the post order array into 2- possible because all elements smaller than pivot goes to left and all elements larger than pivot goes to right and suppose we have x elements smaller than pivot, these elements will be same in both inorder as well as postorder (order may change). We already got the root, now left child is the left split and right child is the right split.

So, doing this recursively gives time complexity of this approach as

$$T(n) = T(k) + T(n - k - 1) + \log n$$

Solving would give $T(n) = O(n \log n)$ in worst case, by putting $k = 0$ and shown at bottom.

But searching for an element in the inorder traversal of given BST can be done in $O(1)$ because we have n elements from $1..n$ so there is no need to search for an element- if last element in post order is say 5 we take it as root and since 4 elements ($1..4$) are smaller we split the post order array in to two- (first 4 elements), (6th element onward) and solve recursively. Time complexity for this would be

$$T(n) = T(k) + T(n - k - 1) + O(1)$$

which gives $T(n) = O(n)$.

Since we know that all elements must be traversed at least once, $T(n) = \Omega(n)$ also and so

$$T(n) = \Theta(n).$$

The following code is doing this.

```
//Install graphviz (sudo apt-get install graphviz on Ubuntu) to view output tree
#include<stdio.h>
#include<stdlib.h>
struct tree
{
    struct tree* left;
    struct tree* right;
    int x;
};
struct tree* makenode(int x)
{
    struct tree * root = malloc(sizeof(struct tree));
    root -> x = x;
    root -> left = root -> right = NULL;
    return root;
}
```

```

}

struct tree* makeBST(int *post, int start, int n, int inorder) {
    if(n <= 0)
        return NULL;
    int pivot = post[start + n - 1];
    struct tree * root = makenode(pivot);
    root -> left = makeBST(post, start, pivot-1 - inorder, inorder );
    root -> right = makeBST(post, pivot - inorder - 1, n - (pivot - inorder), pivot);
    return root;
}
void preorder(struct tree* node)
{
    if(node == NULL)
        return;
    printf("%d ", node->x);
    preorder(node->left);
    preorder(node->right);
}
void printdot(struct tree* node, FILE * f)
{
    if(node == NULL)
        return;
    if(node-> left != NULL)
    {
        fprintf(f, "%d -- %d;\n", node->x, node->left->x);
    }
    if(node-> right != NULL)
    {
        fprintf(f, "%d -- %d;\n", node->x, node->right->x);
    }
    printdot(node->left, f);
    printdot(node->right, f);
}

int main(){
    int i, n, *a;
    printf("Enter n: ");
    scanf("%d", &n);
    a = malloc(n * sizeof(int));
    printf ("Enter post order traversal: ");
    for(i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    struct tree * tree = makeBST(a, 0, n, 0);
    printf("Pre order traversal is : ");
    preorder(tree);
    printf("\n");
    FILE * f = fopen("tree.dot", "w");
    fprintf(f, "graph tree { \n");
    printdot(tree, f);
    fprintf(f, "}\n");
    fclose(f);

    #if defined(__linux__)
        system("dot -Tpng tree.dot -o output.png; eog output.png");
    #endif
}

```

$$T(n) = T(k) + T(n-k-1) + \log n$$

Solving would give $T(n) = O(n \log n)$, by putting $k = 0$,
 $T(n) = T(0) + T(n - 1) + \log n \implies T(n) = O(1) + \log n + \log(n - 1) + \log(n - 2) + \dots + \log 1$
[\(Stirling's Approximation\)](#)

PS: Even for a general BST, given a post order traversal, we can construct the BST in $O(n)$ more
stricter than $O(n \log n)$ derived above and this matches $\Omega(n)$ and hence we do have an $\Theta(n)$
algorithm. This algorithm can be seen at below link

<https://www.geeksforgeeks.org/construct-a-binary-search-tree-from-given-postorder/>

1 50 votes

-- Arjun Suresh (350 points)

3.3.14 Binary Search Tree: GATE2008-IT-12 top

<https://gateoverflow.in/3272>



Selected Answer

A) is true as AVL tree is a balanced search tree that has time complexity of searching $\Theta(\log n)$, but in binary search tree, we can have a completely left/right skewed tree, in which search is $O(n)$.

1 22 votes

-- Happy Mittal (11.2k points)

3.3.15 Binary Search Tree: GATE2008-IT-71 top

<https://gateoverflow.in/3385>

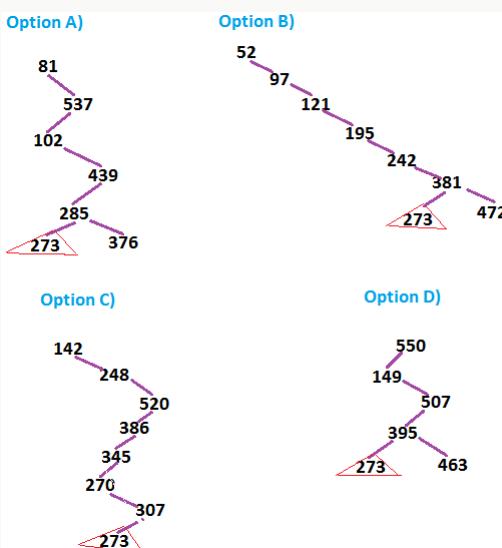


Selected Answer

Option is D.

Which all of the above sequences list nodes in the order in which we could have encountered them in the search?

Question goes like this . IF there had been **273** in the actual sequence then which of the following search sequences would have been successful.



In sequence 1 no need to go from 285 to 376 as 273 is less than 285.

In sequence 2 no need to go from 381 to 472 as 273 is less than 381.

In sequence 4 no need to go from 395 to 463 as 273 is less than 395.

In sequence 3 number 273 might have been to left of 307 and search would have been successful.

Hence **Option D**

25 votes

-- pC (22k points)

Answer: D

I. no need to go from 285 to 376 as 273 is less than 285.

II. no need to go from 381 to 472 as 273 is less than 381.

IV. no need to go from 395 to 463 as 273 is less than 395.

25 votes

-- Rajarshi Sarkar (34.1k points)

3.3.16 Binary Search Tree: GATE2008-IT-72 [top](#)

<https://gateoverflow.in/3386>



A. Incorrect because I & IV are not in ascending order.(Inorder sequence of BST is in increasing order)

B. I is a preorder sequence of some BST with 439 as the root . False because if 439 is root, it should be first element in preorder.

C. This is correct.

D. IV is a postorder sequence of some BST with 149 as the root, False because if 149 is root, it should be last element in postorder

20 votes

-- Akash Kanase (42.5k points)

3.3.17 Binary Search Tree: GATE2008-IT-73 [top](#)

<https://gateoverflow.in/3387>



For number of distinct BSTs with n nodes we apply the formula

$$\frac{C(2n,n)}{n+1}$$

$n = 3$ here, so $C(6,3) = 20$

$$\text{So, } \frac{C(2n,n)}{n+1} = 20/4 = 5$$

Answer is 5

REF :- <https://gatecse.in/number-of-binary-trees-possible-with-n-nodes/>

26 votes

-- Abhimanyu Kumar (207 points)

3.3.18 Binary Search Tree: GATE2009-37,ISRO-DEC2017-55 [top](#)



<https://gateoverflow.in/1323>

Answer is **B**.

With 1 node height is 0.

Max height will come when each level contain min nodes.

Minimum Nodes in an AVL tree with height n is $H(n) = H(n-1) + H(n-2) + 1$.

$$H(0) = 1.$$

$$H(1) = 2$$

$$H(2) = H(1) + H(0) + 1 = 2 + 1 + 1 = 4$$

$$H(3) = H(2) + H(1) + 1 = 4 + 2 + 1 = 7 .$$

So, the max height with 7 nodes is 3.

44 votes

-- Gate Keeda (19.6k points)

3.3.19 Binary Search Tree: GATE2012-5 [top](#)

<https://gateoverflow.in/37>



Binary search takes $\Theta(\log n)$ for n elements in the worst case. So, with $(n2^n)$ elements, the worst case time will be

$$\Theta(\log(n2^n))$$

$$= \Theta(\log n + \log 2^n)$$

$$= \Theta(\log n + n)$$

$$= \Theta(n)$$

42 votes

-- Arjun Suresh (350k points)

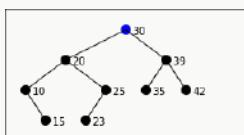
3.3.20 Binary Search Tree: GATE2013-43 [top](#)

<https://gateoverflow.in/1554>



Since it is a binary search tree, its inorder traversal produces a sorted sequence i.e. 10, 15, 20, 23, 25, 30, 35, 39, 42 .

Now given inorder and preorder traversals, we get following tree :



From this, we can give postorder traversal sequence as 15,10,23,25,20,35,42,39,30 i.e. **option (D)**.

26 votes

-- Happy Mittal (11.2k points)

3.3.21 Binary Search Tree: GATE2013-7 top

<https://gateoverflow.in/1416>

Selected Answer

Option (C) is True .

Suppose that we need to insert a node z such that $k = \text{key}[z]$. Using binary search we find a nil such that replacing it by z does not break the BST-property

BST-Insert

 (x, z, k)

1. : if $x = \text{nil}$ then return "Error"
2. : $y \leftarrow x$
3. : while true do {
4. : if $\text{key}[y] < k$
5. : then $z \leftarrow \text{left}[y]$
6. : else $z \leftarrow \text{right}[y]$
7. : if $z = \text{nil}$ break
8. : }
9. : if $\text{key}[y] > k$ then $\text{left}[y] \leftarrow z$
10. : else $\text{right}[y] \leftarrow z$

Time Complexity Analysis :

1. Best Case = $O(1)$, When it is smallest/greatest element and BST contains only all greater/smaller element than inserting element respectively.
2. Avg Case = $O(\log n)$, When it belongs between some elements .
3. Worst Case = $O(n)$, When it is smallest/greatest element and BST contains only all smaller/greater element than inserting element respectively.

28 votes

-- Bhagirathi Nayak (14.1k points)

3.3.22 Binary Search Tree: GATE2014-3-39 top

<https://gateoverflow.in/2073>

Selected Answer

In worst case for finding L and H it will take $O(\log n)$ time as the given tree is **balanced** binary search tree.

Now there are m elements between L and H . So to traverse m element it will take $O(m)$ time (traversal algorithm given below). So, total

$$O(m + \log n) \implies a = 0, b = 1, c = 1, d = 0 \\ \therefore 0 + (10 \times 1) + (100 \times 1) + (1000 \times 0) = 110 .$$

To find all the numbers from L to H we can do an inorder traversal from root and discard all elements before L and after H . But this has $O(n)$ time complexity. So, we can do a modification to inorder traversal and combine with binary search as follows:

1. Find L using binary search and keep all nodes encountered in the search in a stack.
2. After finding L add it to stack as well and initialize $sum = 0$.
3. Now, for all nodes in stack, do an inorder traversal starting from their right node and adding the

node value to sum. If H is found, stop the algorithm.

61 votes

-- Kalpish Singhal (2.1k points)

3.3.23 Binary Search Tree: GATE2015-1-10 top

<https://gateoverflow.in/8129>



In order traversal of key are always in ascending order.

So, here I & IV th sequence are in ascending order so Option A is Answer.

16 votes

-- Rajesh Pradhan (22.6k points)

3.3.24 Binary Search Tree: GATE2015-1-23 top

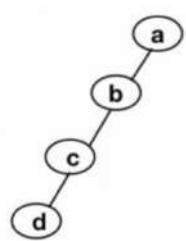
<https://gateoverflow.in/8121>



Option B, both happens when the BST is skewed.

What is [Skewed Tree](#) :

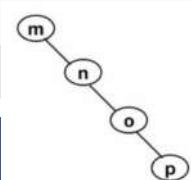
A binary tree which is dominated solely by left child nodes or right child nodes is called a **skewed binary tree** or more specifically **left skewed binary tree** or **right skewed binary tree** respectively.



Left Skewed

28 votes

-- GATERush (1.1k points)



Right Skewed

3.3.25 Binary Search Tree: GATE2015-3-13 top

<https://gateoverflow.in/8409>



On last level 67, hence option B is True.

22 votes

-- Bran Stark (399 points)

3.3.26 Binary Search Tree: GATE2016-2-40 top

<https://gateoverflow.in/39586>



We need to fill 7 levels with 7 elements. So, at each level we have exactly 2 possible options like 1 and 7 for root- one corresponding to making it left skewed and other right skewed. And this is the same for all levels up to 6 giving $2^6 = 64$ possible ways.

85 votes

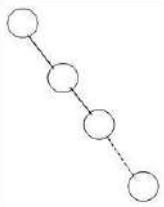
-- Arjun Suresh (350k points)

3.3.27 Binary Search Tree: GATE2017-1-6 top

<https://gateoverflow.in/118286>



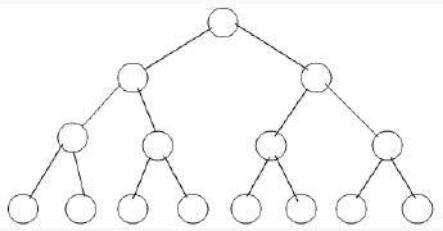
The maximum height of a binary search tree will be when the tree is fully skewed



Maximum height

$$= n - 1 = 15 - 1 = 14$$

The minimum height of a binary search tree will be when the tree is the full complete tree



$$\text{Minimum height} = \log_2(n + 1) - 1 = \log_2(15 + 1) - 1 = \log_2(16) - 1 = 4 - 1 = 3$$

2 votes

-- Pankaj Kumar (8.5k points)

3.3.28 Binary Search Tree: GATE2017-2-36 top

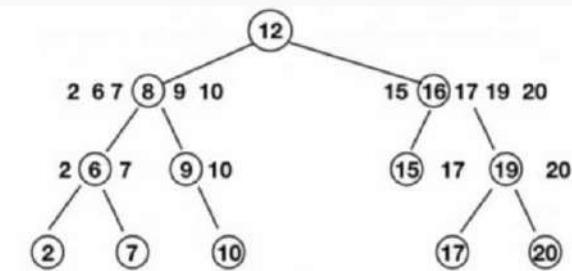
<https://gateoverflow.in/118378>



Preorder: 12 ⑧ 6 2 7 9 10 16 15 19 17 20

Inorder of BST must be sorted in increasing order!

Inorder: {2, 6, 7} 8 {9, 10} 12 {15, 16, 17, 19, 20}



So, Postorder: 2 7 6 10 9 8 15 17 20 19 16 12.

Answer is **B**.

13 votes

-- 2018 (6.7k points)

3.3.29 Binary Search Tree: TIFR2010-B-26 [top](#)

<https://gateoverflow.in/18749>

$O(\log n)$ comparisons and $O(\log n)$ additions. The algorithm is :

1. Find a and b : This will take $O(\log n)$ comparisons as tree is balanced BST.
2. Follow path from a to b , and along the path, keep adding the required number of nodes to result by looking at number stored at each node. Path length is $O(\log n)$, hence number of additions will also be $O(\log n)$.

1 like 7 votes

-- Happy Mittal (11.2k points)

3.4

Binary Tree(50) [top](#)

3.4.1 Binary Tree: GATE1987-2C [top](#)

<https://gateoverflow.in/80579>

State whether the following statements are TRUE or FALSE:

It is possible to construct a binary tree uniquely whose pre-order and post-order traversals are given?

gate1987 binary-tree data-structure normal

Answer

3.4.2 Binary Tree: GATE1987-2g [top](#)

<https://gateoverflow.in/80588>

State whether the following statements are TRUE or FALSE:

If the number of leaves in a tree is not a power of 2, then the tree is not a binary tree.

gate1987 data-structure binary-tree

Answer

3.4.3 Binary Tree: GATE1987-7b [top](#)

<https://gateoverflow.in/82427>

Construct a binary tree whose preorder traversal is

- K L N M P R Q S T

and inorder traversal is

- N L K P R M S Q T

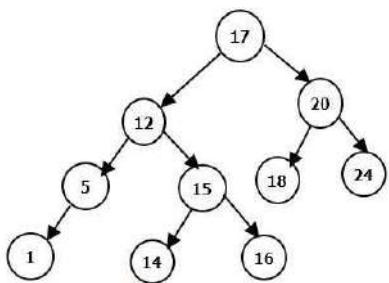
gate1987 data-structure binary-tree

Answer

3.4.4 Binary Tree: GATE1988-7iii [top](#)

<https://gateoverflow.in/94368>

Consider the tree given in the below figure, insert 13 and show the new balance factors that would arise if the tree is not rebalanced. Finally, carry out the required rebalancing of the tree and show the new tree with the balance factors on each mode.



gate1988 normal descriptive data-structure binary-tree

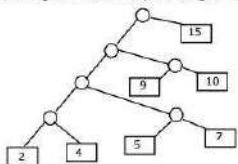
Answer

3.4.5 Binary Tree: GATE1991-01,viii top

<https://gateoverflow.in/506>

The weighted external path length of the binary tree in figure is _____

(viii) The weighted external path length of the binary tree in figure is _____



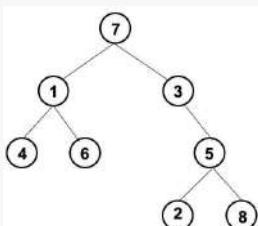
gate1991 binary-tree data-structure normal

Answer

3.4.6 Binary Tree: GATE1991-1,ix top

<https://gateoverflow.in/502>

If the binary tree in figure is traversed in inorder, then the order in which the nodes will be visited is _____



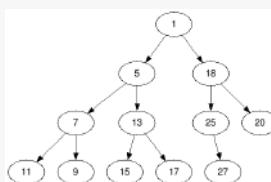
gate1991 binary-tree easy data-structure

Answer

3.4.7 Binary Tree: GATE1991-14,a top

<https://gateoverflow.in/541>

Consider the binary tree in the figure below:



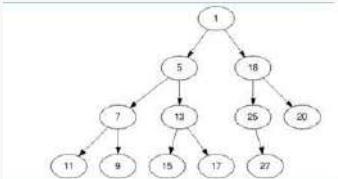
(a). What structure is represented by the binary tree?

[gate1991](#) [data-structure](#) [binary-tree](#) [time-complexity](#) [normal](#)
Answer

3.4.8 Binary Tree: GATE1991-14,b [top](#)

<https://gateoverflow.in/43026>

Consider the binary tree in the figure below:



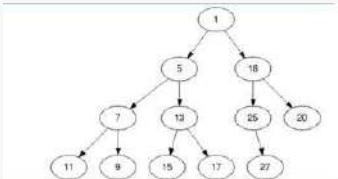
Give different steps for deleting the node with key 5 so that the structure is preserved.

[gate1991](#) [data-structure](#) [binary-tree](#) [normal](#)
Answer

3.4.9 Binary Tree: GATE1991-14,c [top](#)

<https://gateoverflow.in/43027>

Consider the binary tree in the figure below:



Outline a procedure in Pseudo-code to delete an arbitrary node from such a binary tree with n nodes that preserves the structures. What is the worst-case-time-complexity of your procedure?

[gate1991](#) [normal](#) [data-structure](#) [binary-tree](#) [time-complexity](#)
Answer

3.4.10 Binary Tree: GATE1993-16 [top](#)

<https://gateoverflow.in/2313>

Prove by the principle of mathematical induction that for any binary tree, in which every non-leaf node has 2-descendants, the number of leaves in the tree is one more than the number of non-leaf nodes.

[gate1993](#) [data-structure](#) [binary-tree](#) [normal](#)
Answer

3.4.11 Binary Tree: GATE1994-8 [top](#)

<https://gateoverflow.in/2504>

A rooted tree with 12 nodes has its nodes numbered 1 to 12 in pre-order. When the tree is traversed in post-order, the nodes are visited in the order 3, 5, 4, 2, 7, 8, 6, 10, 11, 12, 9, 1 .

Reconstruct the original tree from this information, that is, find the parent of each node, and show the tree diagrammatically.

[gate1994](#) [data-structure](#) [binary-tree](#) [normal](#)
Answer

3.4.12 Binary Tree: GATE1995-1.17 [top](#)

<https://gateoverflow.in/2604>

A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is

- A. $\log_2 n$
- B. $n - 1$
- C. n
- D. 2^n

gate1995 data-structure binary-tree normal

[Answer](#)

3.4.13 Binary Tree: GATE1995-6 top

<https://gateoverflow.in/2667>

What is the number of binary trees with 3 nodes which when traversed in post-order give the sequence A, B, C ? Draw all these binary trees.

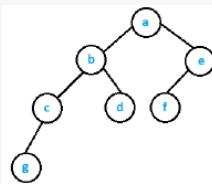
gate1995 data-structure binary-tree normal

[Answer](#)

3.4.14 Binary Tree: GATE1996-1.14 top

<https://gateoverflow.in/2718>

In the balanced binary tree in the below figure, how many nodes will become unbalanced when a node is inserted as a child of the node "g"?



- A. 1
- B. 3
- C. 7
- D. 8

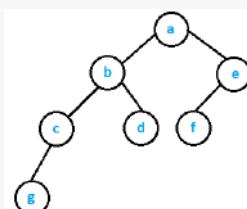
gate1996 data-structure binary-tree normal

[Answer](#)

3.4.15 Binary Tree: GATE1996-1.15 top

<https://gateoverflow.in/2719>

Which of the following sequences denotes the post order traversal sequence of the below tree?



- A. $f \ e \ g \ c \ d \ b \ a$

- B. $g c b d a f e$
 C. $g c d b f e a$
 D. $f e d g c b a$

gate1996 data-structure binary-tree easy

Answer

3.4.16 Binary Tree: GATE1997-16 top

<https://gateoverflow.in/2276>

A size-balanced binary tree is a binary tree in which for every node the difference between the number of nodes in the left and right subtree is at most 1. The distance of a node from the root is the length of the path from the root to the node. The height of a binary tree is the maximum distance of a leaf node from the root.

- A. Prove, by using induction on h , that a size-balance binary tree of height h contains at least 2^h nodes.
 B. In a size-balanced binary tree of height $h \geq 1$, how many nodes are at distance $h - 1$ from the root? Write only the answer without any explanations.

gate1997 data-structure binary-tree normal

Answer

3.4.17 Binary Tree: GATE1997-4.5 top

<https://gateoverflow.in/2246>

A binary search tree contains the value 1, 2, 3, 4, 5, 6, 7, 8. The tree is traversed in pre-order and the values are printed out. Which of the following sequences is a valid output?

- A. 5 3 1 2 4 7 8 6
 B. 5 3 1 2 6 4 8 7
 C. 5 3 2 4 1 6 7 8
 D. 5 3 1 2 4 7 6 8

gate1997 data-structure binary-tree normal

Answer

3.4.18 Binary Tree: GATE1998-20 top

<https://gateoverflow.in/1734>

Draw the binary tree with node labels a, b, c, d, e, f and g for which the inorder and postorder traversals result in the following sequences:

Inorder: a f b c d g e

Postorder: a f c g e d b

gate1998 data-structure binary-tree descriptive

Answer

3.4.19 Binary Tree: GATE2000-1.14 top

<https://gateoverflow.in/637>

Consider the following nested representation of binary trees: $(X Y Z)$ indicates Y and Z are the left and right subtrees, respectively, of node X . Note that Y and Z may be *NULL*, or further nested. Which of the following represents a valid binary tree?

- A. $(1 2 (4 5 6 7))$

- B. $(1 (2 3 4) 5 6) 7)$
 C. $(1 (2 3 4) (5 6 7))$
 D. $(1 (2 3 \text{NULL}) (4 5))$

gate2000 data-structure binary-tree easy

Answer

3.4.20 Binary Tree: GATE2000-2.16 [top](#)

<https://gateoverflow.in/663>

Let LASTPOST, LASTIN and LASTPRE denote the last vertex visited in a postorder, inorder and preorder traversal respectively, of a complete binary tree. Which of the following is always true?

- A. LASTIN = LASTPOST
 B. LASTIN = LASTPRE
 C. LASTPRE = LASTPOST
 D. None of the above

gate2000 data-structure binary-tree normal

Answer

3.4.21 Binary Tree: GATE2002-2.12 [top](#)

<https://gateoverflow.in/842>

A weight-balanced tree is a binary tree in which for each node, the number of nodes in the left sub tree is at least half and at most twice the number of nodes in the right sub tree. The maximum possible height (number of nodes on the path from the root to the furthest leaf) of such a tree on n nodes is best described by which of the following?

- A. $\log_2 n$
 B. $\log_{\frac{4}{3}} n$
 C. $\log_3 n$
 D. $\log_{\frac{3}{2}} n$

gate2002 data-structure binary-tree normal

Answer

3.4.22 Binary Tree: GATE2002-6 [top](#)

<https://gateoverflow.in/859>

Draw all binary trees having exactly three nodes labeled A, B and C on which preorder traversal gives the sequence C, B, A .

gate2002 data-structure binary-tree easy descriptive

Answer

3.4.23 Binary Tree: GATE2004-35 [top](#)

<https://gateoverflow.in/1032>

Consider the label sequences obtained by the following pairs of traversals on a labeled binary tree. Which of these pairs identify a tree uniquely?

- I. preorder and postorder
- II. inorder and postorder
- III. preorder and inorder
- IV. level order and postorder

- A. I only
 B. II, III
 C. III only
 D. IV only

gate2004 data-structure binary-tree normal

Answer

3.4.24 Binary Tree: GATE2004-43 [top](#)

<https://gateoverflow.in/1040>

Consider the following C program segment

```
struct CellNode{
    struct CellNode *leftChild;
    int element;
    struct CellNode *rightChild;
};

int Dosomething (struct CellNode *ptr)
{
    int value = 0;
    if(ptr != NULL)
    {
        if (ptr -> leftChild != NULL)
            value = 1 + Dosomething (ptr -> leftChild);
        if (ptr -> rightChild != NULL)
            value = max(value, 1 + Dosomething (ptr -> rightChild));
    }
    return (value);
}
```

The value returned by the function **Dosomething** when a pointer to the root of a non-empty tree is passed as argument is

- A. The number of leaf nodes in the tree
 B. The number of nodes in the tree
 C. The number of internal nodes in the tree
 D. The height of the tree

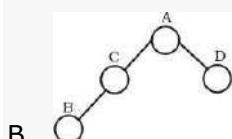
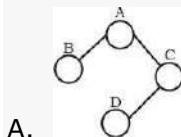
gate2004 data-structure binary-tree normal

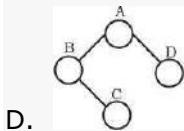
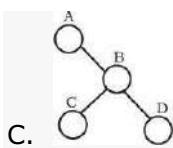
Answer

3.4.25 Binary Tree: GATE2004-IT-54 [top](#)

<https://gateoverflow.in/3697>

Which one of the following binary trees has its inorder and preorder traversals as *BCAD* and *ABCD*, respectively?





gate2004-it binary-tree easy data-structure

[Answer](#)

3.4.26 Binary Tree: GATE2005-33 [top](#)

<https://gateoverflow.in/1369>

Postorder traversal of a given binary search tree, T produces the following sequence of keys

10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29

Which one of the following sequences of keys can be the result of an in-order traversal of the tree T ?

- A. 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95
- B. 9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 27, 29
- C. 29, 15, 9, 10, 25, 22, 23, 27, 40, 60, 50, 95
- D. 95, 50, 60, 40, 27, 23, 22, 25, 10, 9, 15, 29

gate2005 data-structure binary-tree easy

[Answer](#)

3.4.27 Binary Tree: GATE2005-IT-50 [top](#)

<https://gateoverflow.in/3811>

In a binary tree, for every node the difference between the number of nodes in the left and right subtrees is at most 2. If the height of the tree is $h > 0$, then the minimum number of nodes in the tree is

- A. 2^{h-1}
- B. $2^{h-1} + 1$
- C. $2^h - 1$
- D. 2^h

gate2005-it data-structure binary-tree normal

[Answer](#)

3.4.28 Binary Tree: GATE2006-13 [top](#)

<https://gateoverflow.in/974>

A scheme for storing binary trees in an array X is as follows. Indexing of X starts at 1 instead of 0. the root is stored at $X[1]$. For a node stored at $X[i]$, the left child, if any, is stored in $X[2i]$ and the right child, if any, in $X[2i + 1]$. To be able to store any binary tree on n vertices the minimum size of X should be

- A. $\log_2 n$

- B. n
 C. $2n + 1$
 D. $2^n - 1$

gate2006 data-structure binary-tree normal

Answer

3.4.29 Binary Tree: GATE2006-IT-71 top

<https://gateoverflow.in/3615>

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0. The index of the parent of element $X[i], i \neq 0$, is?

- A. $\left\lfloor \frac{i}{2} \right\rfloor$
 B. $\left\lceil \frac{i-1}{2} \right\rceil$
 C. $\left\lceil \frac{i}{2} \right\rceil$
 D. $\left\lceil \frac{i}{2} \right\rceil - 1$

gate2006-it data-structure binary-tree normal

Answer

3.4.30 Binary Tree: GATE2006-IT-73 top

<https://gateoverflow.in/3617>

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0. If the root node is at level 0, the level of element $X[i], i \neq 0$, is

- A. $\lfloor \log_2 i \rfloor$
 B. $\lceil \log_2(i+1) \rceil$
 C. $\lfloor \log_2(i+1) \rfloor$
 D. $\lceil \log_2 i \rceil$

gate2006-it data-structure binary-tree normal

Answer

3.4.31 Binary Tree: GATE2006-IT-9 top

<https://gateoverflow.in/3548>

In a binary tree, the number of internal nodes of degree 1 is 5, and the number of internal nodes of degree 2 is 10. The number of leaf nodes in the binary tree is

- A. 10
 B. 11
 C. 12
 D. 15

gate2006-it data-structure binary-tree normal

Answer

3.4.32 Binary Tree: GATE2007-12 [top](#)

<https://gateoverflow.in/1210>

The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is:

- A. $2^h - 1$
- B. $2^{h-1} - 1$
- C. $2^{h+1} - 1$
- D. 2^{h+1}

gate2007 data-structure binary-tree easy

[Answer](#)

3.4.33 Binary Tree: GATE2007-13 [top](#)

<https://gateoverflow.in/1211>

The maximum number of binary trees that can be formed with three unlabeled nodes is:

- A. 1
- B. 5
- C. 4
- D. 3

gate2007 data-structure binary-tree normal

[Answer](#)

3.4.34 Binary Tree: GATE2007-39, UGCNET-June2015-II-22 [top](#)

<https://gateoverflow.in/1237>

The inorder and preorder traversal of a binary tree are

d b e a f c g and a b d e c f g respectively

The postorder traversal of the binary tree is:

- A. d e b f g c a
- B. e d b g f c a
- C. e d b f g c a
- D. d e f g b c a

gate2007 data-structure binary-tree normal ugcnetjune2015ii

[Answer](#)

3.4.35 Binary Tree: GATE2007-46 [top](#)

<https://gateoverflow.in/1244>

Consider the following C program segment where *CellNode* represents a node in a binary tree:

```
struct CellNode {
    struct CellNode *leftChild;
    int element;
    struct CellNode *rightChild;
};
```

```

int GetValue (struct CellNode *ptr) {
    int value = 0;
    if (ptr != NULL) {
        if ((ptr->leftChild == NULL) &&
            (ptr->rightChild == NULL))
            value = 1;
        else
            value = value + GetValue(ptr->leftChild)
                + GetValue(ptr->rightChild);
    }
    return (value);
}

```

The value returned by *GetValue* when a pointer to the root of a binary tree is passed as its argument is:

- A. the number of nodes in the tree
- B. the number of internal nodes in the tree
- C. the number of leaf nodes in the tree
- D. the height of the tree

[gate2007](#) [data-structure](#) [binary-tree](#) [normal](#)

Answer

3.4.36 Binary Tree: GATE2008-IT-46 [top](#)

<https://gateoverflow.in/3356>

The following three are known to be the preorder, inorder and postorder sequences of a binary tree. But it is not known which is which.

- I. *MBCAFHPYK*
- II. *KAMCBYPFH*
- III. *MABCKYFPH*

Pick the true statement from the following.

- A. I and II are preorder and inorder sequences, respectively
- B. I and III are preorder and postorder sequences, respectively
- C. II is the inorder sequence, but nothing more can be said about the other two sequences
- D. II and III are the preorder and inorder sequences, respectively

[gate2008-it](#) [data-structure](#) [normal](#) [binary-tree](#)

Answer

3.4.37 Binary Tree: GATE2008-IT-76 [top](#)

<https://gateoverflow.in/3390>

A binary tree with $n > 1$ nodes has n_1 , n_2 and n_3 nodes of degree one, two and three respectively. The degree of a node is defined as the number of its neighbours.

n_3 can be expressed as

- A. $n_1 + n_2 - 1$
- B. $n_1 - 2$
- C. $\lceil((n_1 + n_2)/2)\rceil$
- D. $n_2 - 1$

[gate2008-it](#)
[data-structure](#)
[binary-tree](#)
[normal](#)
Answer**3.4.38 Binary Tree: GATE2008-IT-77** [top](#)<https://gateoverflow.in/3391>

A binary tree with $n > 1$ nodes has n_1 , n_2 and n_3 nodes of degree one, two and three respectively. The degree of a node is defined as the number of its neighbours.

Starting with the above tree, while there remains a node v of degree two in the tree, add an edge between the two neighbours of v and then remove v from the tree. How many edges will remain at the end of the process?

- A. $2 * n_1 - 3$
- B. $n_2 + 2 * n_1 - 2$
- C. $n_3 - n_2$
- D. $n_2 + n_1 - 2$

[gate2008-it](#)
[data-structure](#)
[binary-tree](#)
[normal](#)
Answer**3.4.39 Binary Tree: GATE2010-10** [top](#)<https://gateoverflow.in/2183>

In a binary tree with n nodes, every node has an odd number of descendants. Every node is considered to be its own descendant. What is the number of nodes in the tree that have exactly one child?

- A. 0
- B. 1
- C. $\frac{(n-1)}{2}$
- D. $n - 1$

[gate2010](#)
[data-structure](#)
[binary-tree](#)
[normal](#)
Answer**3.4.40 Binary Tree: GATE2011-29** [top](#)<https://gateoverflow.in/2131>

We are given a set of n distinct elements and an unlabeled binary tree with n nodes. In how many ways can we populate the tree with the given set so that it becomes a binary search tree?

- A. 0
- B. 1
- C. $n!$
- D. $\frac{1}{n+1} \cdot {}^{2n}C_n$

[gate2011](#)
[binary-tree](#)
[normal](#)
Answer**3.4.41 Binary Tree: GATE2012-47** [top](#)<https://gateoverflow.in/2163>

The height of a tree is defined as the number of edges on the longest path in the tree. The function shown in the pseudo-code below is invoked as height (root) to compute the height of a binary tree rooted at the tree pointer root.

```

int height (treeptr n)
{ if (n == NULL) return -1;
  if (n->left == NULL)
    if (n->right == NULL) return 0;
  else return [B1]; // Box 1
  else { h1 = height (n->left);
          if (n->right == NULL) return (1+h1);
          else { h2 = height (n->right);
                  }
          }
  return [B2]; // Box 2
}

```

The appropriate expressions for the two boxes **B1** and **B2** are:

- A. **B1**: $(1 + \text{height}(n \rightarrow \text{right}))$; **B2**: $(1 + \max(h1, h2))$
- B. **B1**: $(\text{height}(n \rightarrow \text{right}))$; **B2**: $(1 + \max(h1, h2))$
- C. **B1**: $\text{height}(n \rightarrow \text{right})$; **B2**: $\max(h1, h2)$
- D. **B1**: $(1 + \text{height}(n \rightarrow \text{right}))$; **B2**: $\max(h1, h2)$

gate2012 data-structure binary-tree normal

[Answer](#)

3.4.42 Binary Tree: GATE2014-1-12 [top](#)

<https://gateoverflow.in/1776>

Consider a rooted n node binary tree represented using pointers. The best upper bound on the time required to determine the number of subtrees having exactly 4 nodes is $O(n^a \log^b n)$. Then the value of $a + 10b$ is _____.

gate2014-1 data-structure binary-tree numerical-answers normal

[Answer](#)

3.4.43 Binary Tree: GATE2015-1-25 [top](#)

<https://gateoverflow.in/8223>

The height of a tree is the length of the longest root-to-leaf path in it. The maximum and minimum number of nodes in a binary tree of height 5 are

- A. 63 and 6, respectively
- B. 64 and 5, respectively
- C. 32 and 6, respectively
- D. 31 and 5, respectively

gate2015-1 data-structure binary-tree easy

[Answer](#)

3.4.44 Binary Tree: GATE2015-2-10 [top](#)

<https://gateoverflow.in/8059>

A binary tree T has 20 leaves. The number of nodes in T having two children is _____.

gate2015-2 data-structure binary-tree normal numerical-answers

[Answer](#)

3.4.45 Binary Tree: GATE2015-3-25 [top](#)

<https://gateoverflow.in/8428>

Consider a binary tree T that has 200 leaf nodes. Then the number of nodes in T that have exactly two children are _____.

gate2015-3 data-structure binary-tree normal numerical-answers

[Answer](#)

3.4.46 Binary Tree: GATE2016-2-36 [top](#)

<https://gateoverflow.in/39597>

Consider the following New-order strategy for traversing a binary tree:

- Visit the root;
- Visit the right subtree using New-order;
- Visit the left subtree using New-order;

The New-order traversal of the expression tree corresponding to the reverse polish expression

`3 4 * 5 - 2 ^ 6 7 * 1 + -`

is given by:

- A. $+ - 1 6 7 * 2 \wedge 5 - 3 4 *$
 B. $- + 1 * 6 7 \wedge 2 - 5 * 3 4$
 C. $- + 1 * 7 6 \wedge 2 - 5 * 4 3$
 D. $1 7 6 * + 2 5 4 3 * - \wedge -$

gate2016-2 data-structure binary-tree normal

Answer

3.4.47 Binary Tree: GATE2018-20 [top](#)

<https://gateoverflow.in/204094>

The postorder traversal of a binary tree is $8, 9, 6, 7, 4, 5, 2, 3, 1$. The inorder traversal of the same tree is $text{8, 6, 9, 4, 7, 2, 5, 1, 3}$. The height of a tree is the length of the longest path from the root to any leaf. The height of the binary tree above is _____

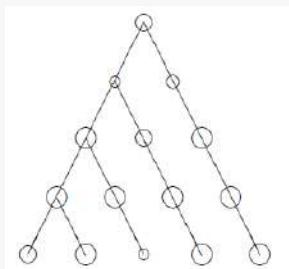
gate2018 data-structure binary-tree numerical-answers

Answer

3.4.48 Binary Tree: TIFR2013-B-13 [top](#)

<https://gateoverflow.in/25775>

Given a binary tree of the following form and having n nodes, the height of the tree is



- a. $\Theta(\log n)$
 b. $\Theta(n)$
 c. $\Theta(\sqrt{n})$
 d. $\Theta(n/\log n)$
 e. None of the above.

tifr2013 binary-tree data-structure

Answer

3.4.49 Binary Tree: TIFR2014-B-1 [top](#)

<https://gateoverflow.in/27133>

Let T be a rooted binary tree whose vertices are labelled with symbols $a, b, c, d, e, f, g, h, i, j, k$. Suppose the in-order (visit left subtree, visit root, visit right subtree) and post-order (visit left subtree, visit right subtree, visit root) traversals of T produce the following

sequences.

in-order: $a, b, c, d, e, f, g, h, i, j, k$

post-order: $a, c, b, e, f, h, j, k, i, g, d$

How many leaves does the tree have?

- a. THREE.
- b. FOUR.
- c. FIVE.
- d. SIX.
- e. Cannot be determined uniquely from the given information.

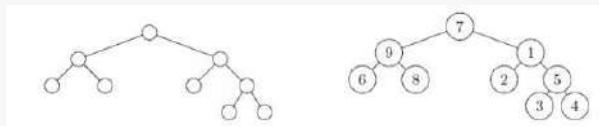
[tifr2014](#) [binary-tree](#) [data-structure](#) [easy](#)

Answer

3.4.50 Binary Tree: TIFR2015-B-4 [top](#)

<https://gateoverflow.in/29849>

First, consider the tree on the left.



On the right, the nine nodes of the tree have been assigned numbers from the set $\{1, 2, \dots, 9\}$ so that for every node, the numbers in its left subtree and right subtree lie in disjoint intervals (that is, all numbers in one subtree are less than all numbers in the other subtree). How many such assignments are possible? Hint: Fix a value for the root and ask what values can then appear in its left and right subtrees.

- A. $2^9 = 512$
- B. $2^4 \cdot 3^2 \cdot 5 \cdot 9 = 6480$
- C. $2^3 \cdot 3 \cdot 5 \cdot 9 = 1080$
- D. $2^4 = 16$
- E. $2^3 \cdot 3^3 = 216$

[tifr2015](#) [binary-tree](#) [permutations-and-combinations](#)

Answer

Answers: Binary Tree

3.4.1 Binary Tree: GATE1987-2C [top](#)

<https://gateoverflow.in/80579>



Selected Answer

Yes it is possible since we can create Binary search tree , we know every Binary search tree is binary tree also but there are many binary tree possible , so we know that there is many binary tree possible without inorder.

So, answer is **NO** for this question

Refer: <http://www.geeksforgeeks.org/if-you-are-given-two-traversal-sequences-can-you-construct-the-binary-tree/>

14 votes

-- Prashant Singh (59.8k points)

3.4.2 Binary Tree: GATE1987-2g top

<https://gateoverflow.in/80588>



Selected Answer

Condition for binary tree is atmost two immediate child for every internal node

No, it is not the condition for binary tree.

12 votes

-- Prashant Singh (59.8k points)

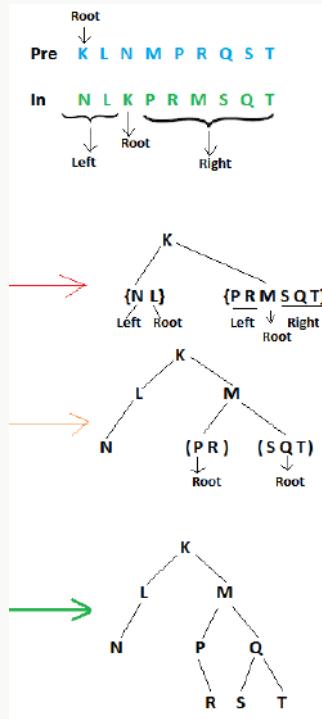
3.4.3 Binary Tree: GATE1987-7b top

<https://gateoverflow.in/82427>



Selected Answer

We can do as follows:



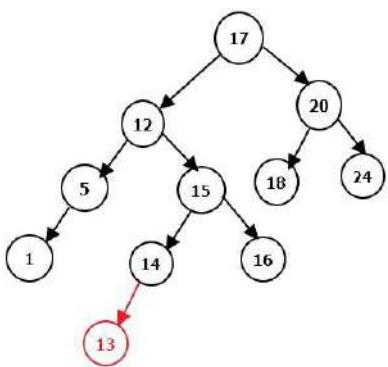
12 votes

-- kirti singh (3.8k points)

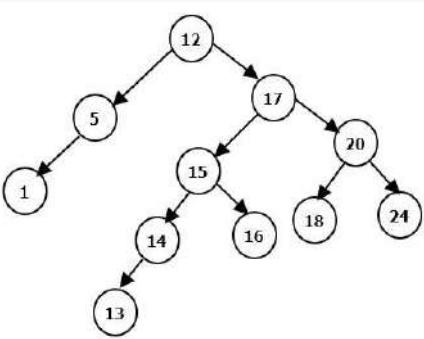
3.4.4 Binary Tree: GATE1988-7iii top

<https://gateoverflow.in/94368>

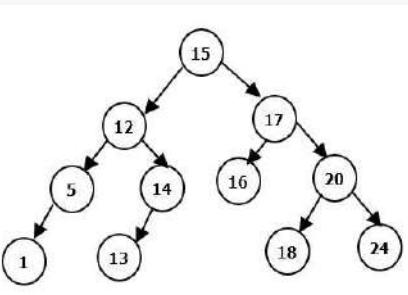
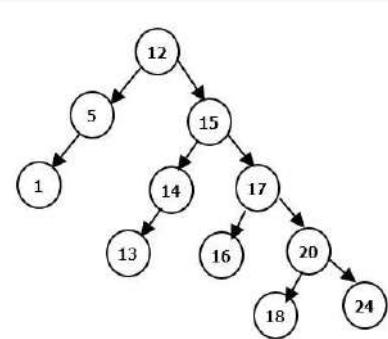
13 is less than 14, so it is placed on the left side of 14



Now need to be rebalanced. Here 17 is unbalanced, So, in the 2nd tree we need 1 right rotation to 17.



Now, 12 is unbalanced. To balance it we need 1 right rotation and 1 left rotation



2 votes

-- srestha (87k points)

3.4.5 Binary Tree: GATE1991-01,viii top

<https://gateoverflow.in/506>



Selected Answer

This is straightforward. The nodes of the given tree are given in square boxes. The weights associated with the nodes are the numbers example 15, 9, 10 etc.

Weighted path length = $\sum_{i=1}^n PathLength_i * WeightOfNode_i$

$$= \sum_{i=1}^n PathLength_i * WeightOfNode_i$$

So answer (written in **path_length*weight** form)
 $= 4*2 + 4*4 + 4*5 + 4*7 + 3*9 + 3*10 + 1*15 = 144$

31 votes

-- arvchamp (171 points)

3.4.6 Binary Tree: GATE1991-1,ix top

<https://gateoverflow.in/502>



Selected Answer

4 1 6 7 3 2 5 8

15 votes

-- Keith Kr (6.1k points)

3.4.7 Binary Tree: GATE1991-14,a top

<https://gateoverflow.in/541>



Selected Answer

(A) This is min heap. It is obvious looking at tree.

(B) Assuming that we have stored this heap in array structure, Procedure ->

1. Search for element 5 using sequential search in array.
2. Swap it with last element in this case 27.
3. Bubble down it so that min heap property is satisfied.

(C) Deleting element from min heap , $O(\log n)$, but for searching in Heap we need $O(N)$

So, time complexity of sequential search + Delete = $> O(N) + O(\log N) = O(N)$.

We can't use binary search as it is heap.

22 votes

-- Akash Kanase (42.5k points)

3.4.8 Binary Tree: GATE1991-14,b top

<https://gateoverflow.in/43026>



Selected Answer

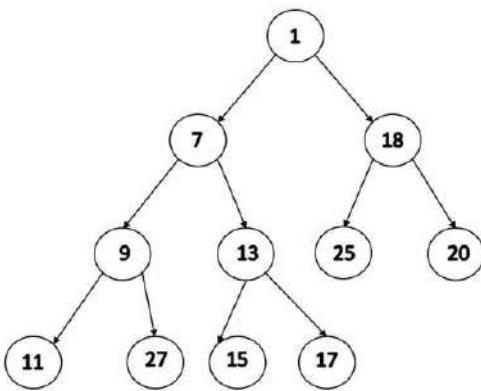
Since given Binary tree is minheap tree.

First swap 27 and 5

Then delete 5

Apply Minheapify

And Structure will be



13 votes

-- Manoj Kumar (38.6k points)

3.4.9 Binary Tree: GATE1991-14, C top

<https://gateoverflow.in/43027>



Selected Answer

By looking at the values it is clear that It is a **Min-Heap** Data structures. We know that, Heap Data structures are stored in the array.

==> **Delete procedure for Min-Heap Data Structure (If you already know value and position of the node):**

1. Replace that node with the last element of that tree.
2. Apply Heapify property on that node.

For Example, Let If I want to delete 1, then I will replace that with 27. and apply heapify on that node. Or if i want to delete 5 then also I will replace that with 27, and apply heapify on that node.

Time Complexity: In this case time complexity will not be more than $O(\log n)$.

==> **Delete procedure for Min-Heap Data Structure (If you know the value but not position) :**

1. Find the position of the number by sequential search. (In worst case it will take $O(n)$ time).
2. Replace that node with the last element of that tree.
3. Apply heapify property at that node. }

Time Complexity: Worst time complexity of this algorithm will be **$O(n + \log n)$ i.e. $O(n)$** .

Note: This is a standard problem of Minimum element deletion from Min-heap tree. Minimum element always resides at top (Root node). We just replace that value with the last element of the tree and apply heapify at the root node. Time complexity of that algorithm is **$O(\log n)$** .

Here I have written second method only to show that if we have to delete any of the node, and we just know the value but not the position. Since in question it is mentioned that **Arbitrary node**.

28 votes

-- Muktinath Vishwakarma (35.4k points)

3.4.10 Binary Tree: GATE1993-16 top

<https://gateoverflow.in/2313>



Selected Answer

Base Case :- When we have just root then, there are no non leaf nodes. So No of leaves = 1, No of non leaf nodes is = 0. Base case holds.

Induction Hypothesis :- Assume that now for k internal nodes we will have $k + 1$ leaves.

Inducting on no of leaves, Now we add 2 more leaves to this tree. One of $k + 1$ leaf will become internal node. So now we will have $k + 1$ internal node. No of leafs will be $K + 1 - 1$ (1 leaf just became internal node) + 2 (New leafs). So we proved that for any binary tree, in which every non-leaf node has 2-descendants, the number of leaves in the tree is one more than the number of non-leaf nodes.

12 votes

-- Akash Kanase (42.5k points)

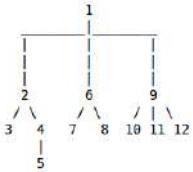
3.4.11 Binary Tree: GATE1994-8 top

<https://gateoverflow.in/2504>



Selected Answer

Answer: The tree is a ternary tree.



12 votes

-- Rajarshi Sarkar (34.1k points)

3.4.12 Binary Tree: GATE1995-1.17 top

<https://gateoverflow.in/2604>



Selected Answer

In Binary Tree a node can have at most 2 children.

Total number of nodes $\Rightarrow N = \text{node with 0 child} + \text{node with 1 child} + \text{node with 2 child}$.

$N = n_0 + n_1 + n_2$ (here, in question it is given that no. of leaf nodes i.e no. of nodes with 0 children is n so $n_0 = n$)

$$N = n + n_1 + n_2$$

Total number of edges

$$e = N - 1$$

$$\text{and also } e = n * 0 + n_1 * 1 + n_2 * 2$$

Therefore,

$$N - 1 = n * 0 + n_1 * 1 + n_2 * 2$$

$$n + n_1 + n_2 - 1 = n_1 * 1 + n_2 * 2$$

therefore

$$n_2 = n - 1$$

Option B is answer.

NOTE - For tree, **Degree** of a **node** is defined as the number of sub-trees of the **node or no of children of a node ...**

36 votes

-- Umang Raman (15.7k points)

3.4.13 Binary Tree: GATE1995-6 top

<https://gateoverflow.in/2667>



Selected Answer

There are only Five such binary trees. This is given by 3rd [Catalan number](#) as here we are finding the number of structurally similar binary trees with 3 nodes.

1. One with C as root and left child as A and right child B .
2. Second with C as root, B as left child and A as again left child of B .
3. Third with C as root, B as left child and A as right child of B .
4. Fourth with C as root, B as right child and A as right child of B .
5. Fifth with C as root, B as right child and A as left child of B .

👍 23 votes

-- Gate Keeda (19.6k points)

3.4.14 Binary Tree: GATE1996-1.14 top

<https://gateoverflow.in/2718>



Selected Answer

(B).

a, b, c will become unbalanced with Balance factor as +2, +2, +2 respectively. Balance factor should be -1, 0, +1.

Balance factor = Height(LST) - Height(RST)

Or Balance factor = | Height(LST) - Height(RST) |

👍 19 votes

-- Gate Keeda (19.6k points)

3.4.15 Binary Tree: GATE1996-1.15 top

https://gateoverflow.in/2718/gate1996_1-14



Selected Answer

C.

Left → Right → Root.

Ref: https://gateoverflow.in/2718/gate1996_1-14

👍 16 votes

-- Gate Keeda (19.6k points)

3.4.16 Binary Tree: GATE1997-16 top

<https://gateoverflow.in/2276>



Selected Answer

- a. Prove, by using induction on h , that a size-balanced binary tree of height h contains at least 2^h nodes.

When

$$h = 0 \dots \text{least no. of nodes} = 2^0 = 1$$

$$h = 1 \dots \text{least no. of nodes} = 2^1 = 2$$

$h = 2$ least no. of nodes $= 2^2 = 4$

Assume that the rule is true for $h = k$

Then the min no. of nodes $= 2^k$ nodes

If we increase the height by 1 by adding a node, we must also add nodes to fill the (max level -1) level.

This would mean doubling the nodes

Thus 2^{k+1}

Hence, proved

- b. In a size-balanced binary tree of height $h \geq 1$, how many nodes are at distance $h - 1$ from the root? Write only the answer without any explanation
 2^{h-1}

1 6 votes

-- Sachin Mittal (15.7k points)

3.4.17 Binary Tree: GATE1997-4.5 top

<https://gateoverflow.in/2246>



Selected Answer

Note: PreOrder means Root-Left-Right Traversal of tree.

By Option Elimination:-

B. False. Because 5 is root so in preorder sequence first 5 elements must contain 1 to 5 But 6 comes here. So false.

So, now common things in option A,C,D is 5,3 means 5 root then LHS subtree root is 3 . now 3's LHS is 1,2 so they should come first rather than 4. So option C is False.

Now Check for Option A,D.

Root 5's RHS is 6,7,8 now as per Option A,D 7 is Root so 6 should be Left and 8 should be Right so pre order for Right sub tree is 7,6,8 (Root-L-R). This satisfies option D.

1 19 votes

-- Rajesh Pradhan (22.6k points)

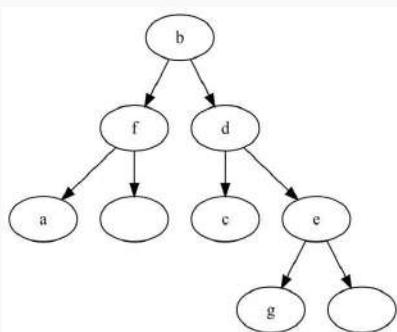
3.4.18 Binary Tree: GATE1998-20 top

<https://gateoverflow.in/1734>



Selected Answer

The binary tree will be



16 votes

-- Anu (5.9k points)

3.4.19 Binary Tree: GATE2000-1.14 top

<https://gateoverflow.in/637>



- A. → (4 5 6 7) this part of answer is not correct. We have (X Y Z) not (W X Y Z). So, this is wrong.
 B. → 3 closing parenthesis, 2 opening parenthesis. This is wrong.
 C. CORRECT
 D. → Here in (1 (2 3 NULL) (4 5)), (4 5) this is not allowed. So this is wrong. (It should be (4,5,NULL))

28 votes

-- Akash Kanase (42.5k points)

3.4.20 Binary Tree: GATE2000-2.16 top

<https://gateoverflow.in/663>



Inorder : Left → Root → Right

Preorder : Root → Left → Right

Postorder: Left → Right → Root

If the binary tree is full (last level is fully filled), the last visited node in Inorder and Preorder must be the rightmost one in the last level. But for a complete binary tree this need not be the case (in a complete binary tree last level need not be fully filled) and LASTPRE will be from the second last level in case the complete binary tree is not full. So, choice (D).

26 votes

-- Arjun Suresh (350k points)

3.4.21 Binary Tree: GATE2002-2.12 top

<https://gateoverflow.in/842>



Let n_l and n_r nodes be present in the left and right sub-trees respectively.

We have, $\frac{n_r}{2} \leq n_l \leq 2n_r$. Without loss of generality, let the left sub-tree have greater number of nodes ($2n_r$ nodes). Then, $n_r + 2n_r + 1 = n$. Thus we get, $n_l = \frac{2(n-1)}{3}$ and $n_r = \frac{n-1}{3}$.

Total number of nodes can be described by the recurrence:

$$T(n) = T\left(\frac{n-1}{3}\right) + T\left(\frac{2(n-1)}{3}\right) + 1 \text{ and } T(1) = 1$$

As this makes maximum nodes go to one subtree and that is what we want to get the maximum height with a given number of nodes.

Now, the height of the tree will be: $H(n) = H(\frac{2}{3}(n - 1)) + 1$ and $H(1) = 0$

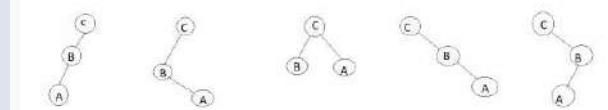
We can draw a recurrence tree and the cost at each level is 1, and the height will be $\log_{\frac{3}{2}}(n)$.
So, **D** option is the answer.

45 votes

-- Arjun Suresh (350k points)

3.4.22 Binary Tree: GATE2002-6 top

<https://gateoverflow.in/859>



5 Binary trees

14 votes

-- Anu (5.9k points)

3.4.23 Binary Tree: GATE2004-35 top

<https://gateoverflow.in/1032>



Following combination can uniquely identify a tree.

- Inorder and Preorder.
- Inorder and Postorder.
- Inorder and Level-order.

And following do not.

- Postorder and Preorder.
- Preorder and Level-order.
- Postorder and Level-order.

Answer: **B**

17 votes

-- Shikhar Vashishth (4.1k points)

3.4.24 Binary Tree: GATE2004-43 top

<https://gateoverflow.in/1040>



(D) It calculates Height of tree.

Easy way to get this answer .

Draw a tree where all 4 parameters are different.

Get a Tree for which Height, No of Internal Nodes & No of Leafs are different & Trace out this algorithm.

19 votes

-- Akash Kanase (42.5k points)

3.4.25 Binary Tree: GATE2004-IT-54 [top](#)

<https://gateoverflow.in/3697>



Selected Answer

Answer is **D**.

Inorder traversal is left node right.

Preorder is node left right.

19 votes

-- Sankaranarayanan P.N (11.5k points)

3.4.26 Binary Tree: GATE2005-33 [top](#)

<https://gateoverflow.in/1369>



Selected Answer

In order traversal of b binary search tree returns the element in sorted order - ascending (inorder is left parent then right. in a bst left is less than parent and right is greater than parent). In this option 1 is the only sorted list. hence it is the only possibility.

24 votes

-- Sankaranarayanan P.N (11.5k points)

3.4.27 Binary Tree: GATE2005-IT-50 [top](#)

<https://gateoverflow.in/3811>



Selected Answer

It should be **B**)

Since the difference between the nodes in left and right subtree must hold for every node, until the last to last to last level, all levels must be fully filled. So, we get $2^{h-1} - 1$ nodes (No. of nodes in a complete binary tree of height $h - 2$). Now, our aim is to increase two more levels by adding minimum no. of nodes- just add two in nodes one below other to any of the nodes. So, we get $2^{h-1} + 1$ nodes - **B** option.

33 votes

-- Sneha Goel (1.1k points)

3.4.28 Binary Tree: GATE2006-13 [top](#)

<https://gateoverflow.in/974>



Selected Answer

Answer is **D**.

To be able to store " **any** " binary tree on n vertices the minimum size of X should be

" Any Binary Tree and Size should be minimum ".

So We must consider worst case binary tree for this situation and find the minimum space required

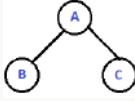
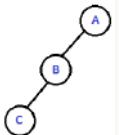
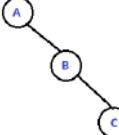
.

Minimum size for any binary tree

⇒ Minimum size of worst case binary tree

$$X[i] = \text{node}X[2i] = \text{Left child}X[2i + 1] = \text{Right child}$$

Let $n = 3$

																														
<table border="1"><tr><td>A</td><td>B</td><td>C</td></tr><tr><td>1</td><td>2</td><td>3</td></tr></table>	A	B	C	1	2	3	<table border="1"><tr><td>A</td><td>B</td><td>Null</td><td>C</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	A	B	Null	C	1	2	3	4	<table border="1"><tr><td>A</td><td>Null</td><td>B</td><td>Null</td><td>Null</td><td>Null</td><td>C</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr></table>	A	Null	B	Null	Null	Null	C	1	2	3	4	5	6	7
A	B	C																												
1	2	3																												
A	B	Null	C																											
1	2	3	4																											
A	Null	B	Null	Null	Null	C																								
1	2	3	4	5	6	7																								
$X[1]$ $= A$ $X[2]$ $= B$ $X[3]$ $= C$	$X[1] = A$ $X[2] = B$ $X[4] = C$	$X[1]$ $= A$ $X[3]$ $= B$ $X[7]$ $= C$																												
n	2^{n-1}	$2^n - 1$																												
Minimum size Best Case binary tree		Minimum size Worst Case binary tree																												

32 votes

-- pC (22k points)

Answer should be (D).

Since binary tree can be of any form, the worst case happens for right skewed binary tree. Now, root goes to index 1, its child goes to index 3, its child goes to index 7 and so on the nth vertex goes to $2^n - 1$ th index of array.

32 votes

-- Shaun Patel (6.9k points)

3.4.29 Binary Tree: GATE2006-IT-71 top

<https://gateoverflow.in/3615>



Option is (D).

Left child of ith element will be at $2 * i + 1$ and right child at $2(i + 1)$

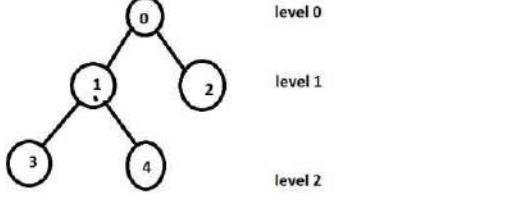
24 votes

-- Sankaranarayanan P.N (11.5k points)

3.4.30 Binary Tree: GATE2006-IT-73 top

<https://gateoverflow.in/3617>

 Selected Answer



level 0
level 1
level 2

PS : the value inside the node is array index

- A. **FALSE** , $\lfloor \log_2 i \rfloor = \lfloor \log_2 1 \rfloor = 0$ but $X[1]$ is at level 1
- B. **FALSE** , $\lceil \log_2(i+1) \rceil = \lceil \log_2(4+1) \rceil = \lceil \log_2(5) \rceil = 3$ but $X[4]$ is at level 2
- C. **Correct**
- D. **FALSE** , $\lceil \log_2 i \rceil = \lceil \log_2 1 \rceil = 0$ but $X[1]$ is at level 1

10 votes

-- Gate Ranker18 (2.3k points)

3.4.31 Binary Tree: GATE2006-IT-9 [top](#)

<https://gateoverflow.in/3548>

 Selected Answer

A node in a binary tree has degree 0,1 or 2.

Ref: <http://faculty.cs.niu.edu/~mcmahon/CS241/Notes/bintree.html>

We are given no. of 1 degree node = 5, no. of 2 degree nodes = 10.

Total no. of edges = $1 * 5 + 2 * 10 = 25$ (In tree degree is for outgoing edges only, and hence each degree corresponds to an edge)

So, total no. of nodes = $25 + 1 = 26$ (No. of nodes in a tree is 1 more than no. of edges).

Now, no. of leaf nodes (nodes with 0 degree) = $26 - 5 - 10 = 11$.

40 votes

-- Arjun Suresh (350k points)

3.4.32 Binary Tree: GATE2007-12 [top](#)

<https://gateoverflow.in/1210>

 Selected Answer

$2^{h+1} - 1$ just try this taking a small complete binary

never try to remember these formulae as remembering formulae is an overhead try to take examples in such cases.

23 votes

-- Bhagirathi Nayak (14.1k points)

3.4.33 Binary Tree: GATE2007-13 [top](#)

<https://gateoverflow.in/1211>



Can be found with formula... $(2nCn/n + 1)$... n being the number of nodes. for the given question... where $n = 3$... answer is 5. Let me also specify here.. that number of Binary Search Trees with n nodes is equal to number of unlabeled Binary trees.

http://gatecse.in/wiki/Number_of_Binary_trees_possible_with_n_nodes

1 21 votes

-- Gate Keeda (19.6k points)

3.4.34 Binary Tree: GATE2007-39, UGCNET-June2015-II-22 [top](#)



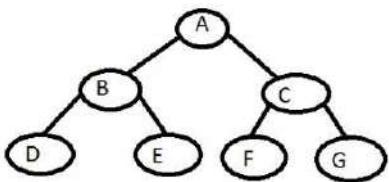
<https://gateoverflow.in/1237>

The answer is A.

Take the first node in preorder traversal - a will be the root of the tree
All nodes to the left of 'a' in inorder traversal will be in the left subtree of 'a' and all elements on the right will be in the right subtree of 'a'.

Take the second element from preorder traversal - 'b' - goes to left subtree of 'a' as it is in the left of 'a' in inorder list.

Proceeding likewise we can construct the binary tree as:



1 16 votes

-- Gate Keeda (19.6k points)

3.4.35 Binary Tree: GATE2007-46 [top](#)

<https://gateoverflow.in/1244>



Answer: C

As the function returns 1 if and only if any node has both left & right children as *NULL* (that node is a leaf node). Hence, `value` gets incremented at each leaf node.

1 19 votes

-- Rajarshi Sarkar (34.1k points)

3.4.36 Binary Tree: GATE2008-IT-46 [top](#)

<https://gateoverflow.in/3356>



In preorder, root comes at the beginning of the traversal sequence and in postorder, root comes at the last of the traversal sequence. So, out of the given sequences only 1 and 2 are having such kind of order i.e K at the beginning and at the last.

Therefore, 2 is the preorder and 1 is postorder and the left sequence i.e 3 will definitely be inorder.

So, option **D** is correct.

27 votes

-- Vivek sharma (2.4k points)

3.4.37 Binary Tree: GATE2008-IT-76 top

<https://gateoverflow.in/3390>



Selected Answer

Given definition of degree: no of neighbours of a node.

$$\text{total nodes} = n = n_1 + n_2 + n_3$$

Apply handshaking lemma:

$$\text{Sum of degrees} = 2 * \text{no of edges}$$

$$1 * n_1 + 2 * n_2 + 3 * n_3 = 2(n - 1)$$

Total number of edges in graph will always be $(n_1 + n_2 + n_3 - 1)$.

$$1 * n_1 + 2 * n_2 + 3 * n_3 = 2(n_1 + n_2 + n_3 - 1)$$

$$n_1 + 2n_2 + 3n_3 = 2n_1 + 2n_2 + 2n_3 - 2$$

$$n_3 = n_1 - 2 \quad \textbf{Option B}$$

25 votes

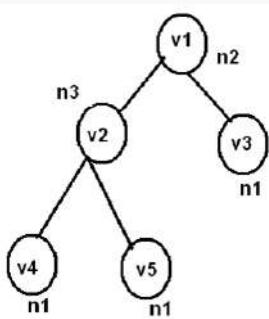
-- swagnikd (723 points)

3.4.38 Binary Tree: GATE2008-IT-77 top

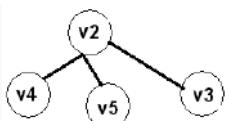
<https://gateoverflow.in/3391>



Selected Answer



From above tree we will get the tree below



Now, check with the options we will get (A) as answer.

25 votes

-- Shreyans Dhankhar (2.5k points)

3.4.39 Binary Tree: GATE2010-10 top

<https://gateoverflow.in/2183>



Selected Answer

0 because every node has an odd number of descendants so least odd number 1 and every node is considered to be its own descendant so all nodes have even number of descendants (0,2,4,6...) so every node has either 0 children or 2 children...

26 votes

-- Murali (505 points)

3.4.40 Binary Tree: GATE2011-29 top

<https://gateoverflow.in/2131>



Selected Answer

With n nodes, there are $\frac{2nCn}{(n+1)}$ distinct tree structures possible.

Corresponding to each structure, only one binary search tree (BST) can be formed because inorder is fixed.

Here, we are already given one such structure therefore only one tree possible.

If binary trees would have been asked, $n!$ trees would have been possible corresponding to each distinct tree structure. Here, tree structure is fixed and hence, we can have only one possibility for BST as elements are distinct. For general cases:

http://gatecse.in/wiki/Number_of_Binary_trees_possible_with_n_nodes

29 votes

-- Anurag Semwal (8k points)

Given binary tree is unlabeled . So as it is given we are not allowed to change the formation of tree. Then To make it BST we can use atmost 1 way . As for particular structure we can not use n! arrangement of nodes (Becasue they are labeled and it is BST not BT)

26 votes

-- Palash Nandi (1.5k points)

3.4.41 Binary Tree: GATE2012-47 top

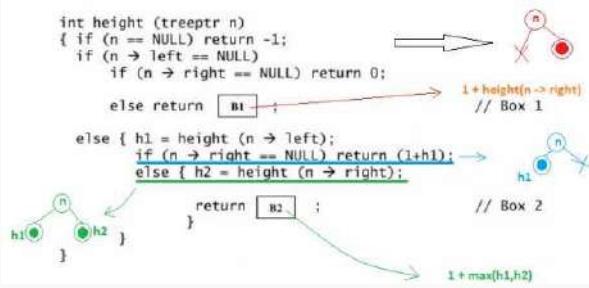
<https://gateoverflow.in/2163>



Selected Answer

Answer is **option A**.

From the diagram below we are able to see how this works :



20 votes

-- Amar Vashishth (30.5k points)

3.4.42 Binary Tree: GATE2014-1-12 top

<https://gateoverflow.in/1776>



Answer: 1..

Explanation:

1. Come to the 4th level up from the leaf node of the given binary tree, which can be done using tree traversal in $O(n)$.
2. For each node present in the level check whether it's subtree having exactly 4 nodes.. which can be done in constant time for each node, since it's subtree having constant number of nodes..
3. nodes in the level is less than n.. so its complexity is $O(n)$

Therefore, $a = 1$ and $b = 0$

$a + 10b = 1$... <- Answer

31 votes

-- Vicky Bajoria (5k points)

We need to traverse all nodes at least once, and we need only one traversal. If $\text{num}(\text{child1}) + \text{num}(\text{child2}) + 1 = 4$, then output yes.

So, a must be 1 and $b = 0 \Rightarrow a + 10b = 1$.

33 votes

-- Arjun Suresh (350k points)

3.4.43 Binary Tree: GATE2015-1-25 top

<https://gateoverflow.in/8223>



Option A is correct because height 5 means level 6 so maximum node = $2^l - 1 = 2^6 - 1 = 63$ and for minimum, at each level only single node so total 6.

30 votes

-- Anoop Sonkar (5k points)

3.4.44 Binary Tree: GATE2015-2-10 top

<https://gateoverflow.in/8059>



19

In Binary tree If there are N leaf nodes then the number of Nodes having two children will be $N - 1$. So in this case answer will be $20 - 1$, means 19.

4 votes

-- Raghuvveer Dhakad (1.6k points)

3.4.45 Binary Tree: GATE2015-3-25 top

<https://gateoverflow.in/8428>


Let number of nodes with exactly two children be x , and with exactly one children be y .

Total degree = $200 + 3x + 2y - 1$ (As all nodes with 2 children have degree 3 except the root)

No. of nodes = $x + y + 200$

No. of edges = Total degree/2 = $(200 + 3x + 2y - 1)/2$ [Handshaking Theorem]

No. of edges in a tree = No. of nodes - 1

So, $(200 + 3x + 2y - 1) = 2x + 2y + 400 - 2$

$x = 199$

46 votes

-- Arjun Suresh (350k points)

3.4.46 Binary Tree: GATE2016-2-36 top

<https://gateoverflow.in/39597>


Expression given in reverse polish notation (i,e in Post-order)

convert first it into In-order

$3 \ 4 \ * \ 5 \ - \ 2 \ \wedge \ 6 \ 7 \ * \ 1 \ + \ -$

$(3 * 4) \ 5 \ - \ 2 \ \wedge \ 6 \ 7 \ * \ 1 \ + \ -$

$((3 * 4) - 5) \ 2 \ \wedge \ 6 \ 7 \ * \ 1 \ + \ -$

$((((3 * 4) - 5) \wedge 2) \ 6 \ 7 \ * \ 1 \ + \ -$

$((((3 * 4) - 5) \wedge 2) \ (6 * 7) \ 1 \ + \ -$

$((((3 * 4) - 5) \wedge 2) \ ((6 * 7) + 1) \ -$

$(((((3 * 4) - 5) \wedge 2) - ((6 * 7) + 1))$

so Inorder expression is $(((((3 * 4) - 5) \wedge 2) - ((6 * 7) + 1))$

New-Order traversal is as by ROOT RIGHT LEFT

$(((((3 * 4) - 5) \wedge 2) - ((6 * 7) + 1))$

$-(6 * 7) + 1(((3 * 4) - 5) \wedge 2)$
 $- + 1(6 * 7)((3 * 4) - 5) \wedge 2)$
 $- + 1 * 7 6((3 * 4) - 5) \wedge 2)$
 $- + 1 * 7 6 \wedge 2((3 * 4) - 5)$
 $- + 1 * 7 6 \wedge 2 - 5(3 * 4)$
 $- + 1 * 7 6 \wedge 2 - 5 * 4 3$

option C is correct

👍 42 votes

-- Praveen Saini (54.6k points)

It is quite simple actually.

Postorder: left, right, root

Neworder: Root, right, left

(left, right, root) and (Root, right, left) are reverse of each other, right ?

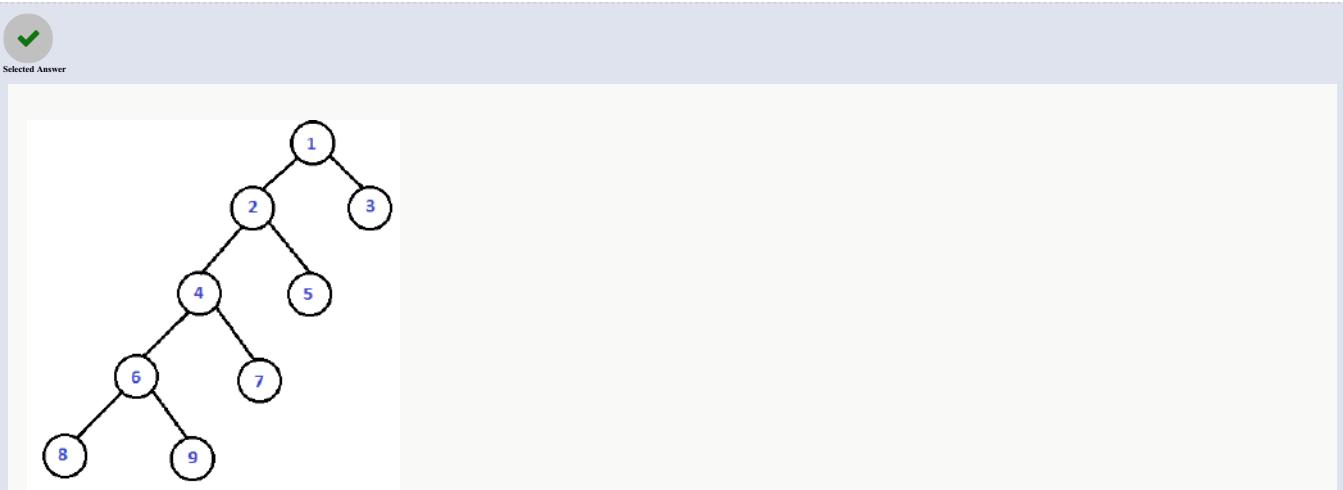
So, these two traversals will be exact reverse of each other! Option (c)!

👍 60 votes

-- Ashish Deshmukh (1.5k points)

3.4.47 Binary Tree: GATE2018-20 top

<https://gateoverflow.in/204094>



Nodes in longest path from Root to Leaf
 $= (1 - 2, 2 - 4, 4 - 6, 6 - 8) \text{ or } (1 - 2, 2 - 4, 4 - 6, 6 - 9)$
 $|Longest Path| = |(1 - 2, 2 - 4, 4 - 6, 6 - 8)| = 4$

👍 13 votes

-- Digvijay (54.9k points)

3.4.48 Binary Tree: TIFR2013-B-13 top

<https://gateoverflow.in/25775>



The correct answer is option c, $\Theta(\sqrt{n})$.

$$n = 1 + 2 + 3 + \dots + (h + 1)$$

$$= \frac{(h+1)(h+2)}{2}$$

$$2n = h^2 + 3h + 2$$

$$0 = h^2 + 3h + (2 - 2n)$$

$$\Rightarrow h = \frac{-3 + \sqrt{3^2 - 4 \cdot (2 - 2n)}}{2} \quad \because h \geq 0$$

$$= \frac{-3 + \sqrt{8n+1}}{2}$$

$$\Rightarrow h = \Theta(\sqrt{n})$$

40 votes

-- Pragy Agarwal (20.6k points)

3.4.49 Binary Tree: TIFR2014-B-1 top

<https://gateoverflow.in/27133>



Selected Answer

We can construct binary tree by postorder and inorder traversal

There we get 5 leaves of the tree are a, c, e, h, j

So, answer (C) 5

9 votes

-- srestha (87k points)

3.4.50 Binary Tree: TIFR2015-B-4 top

<https://gateoverflow.in/29849>



Selected Answer

Option is B.

for every node - all numbers in one subtree are less than all numbers in the other subtree .

Firstly **choose** a value for **root** — 9 elements = **9 ways**

Now, we hv **8 elements left** - we hv to **choose 3 for left subtree & 5 for right subtree**.

Note: Here we can either choose 3 nodes from beginning or end out of 8 elements we have ! = **2**

ways

Now, we hv 3 elements for left subtree & 5 for right (**Consider subtrees of subtree**).

Left Subtree :

whatever way we place , always one side is smaller than other {6 is smaller than 8 in above example given in question} so, total ways = **3!** {three places put one by one} = **6 ways**

Right Subtree :

Right subtree has two more sub-trees ,so that elements on one side should be smaller than other
**

Steps :

1. Select one element for root = **5 ways**
2. 4 elements left ,Select one element for left = **2 ways** {Either we can chose from left or right}
3. 3 elements left, for right subtree = **3!** ways = **6 ways**

$$\text{Total ways} = 9 * 2 * 3! * 5 * 2 * 3! = 2^4 * 3^2 * 5 * 9 = 6480 = \mathbf{B (Ans)}$$

31 votes

-- **Himanshu Agarwal** (15.3k points)

3.5**Graph Search(1)****3.5.1 Graph Search: GATE1989-3-ixa**

<https://gateoverflow.in/87143>

Answer the following:

Which one of the following statements (s) is/are FALSE?

- A. Overlaying is used to run a program, which is longer than the address space of the computer.
- B. Optimal binary search tree construction can be performed efficiently by using dynamic programming.
- C. Depth first search cannot be used to find connected components of a graph.
- D. Given the prefix and postfix walls over a binary tree, the binary tree can be uniquely constructed.

[normal](#) [gate1989](#) [overlay](#) [binary-tree](#) [graph-search](#)

Answer

Answers: Graph Search**3.5.1 Graph Search: GATE1989-3-ixa**

<https://gateoverflow.in/87143>

- A. **FALSE** according to definition of address space given in the link. whatever memory used by the overlay comes under the address space of computer "https://en.wikipedia.org/wiki/Address_space".
- B. **TRUE** Optimal binary search tree construction can be performed efficiently by using dynamic programming.
ref: <http://www.geeksforgeeks.org/dynamic-programming-set-24-optimal-binary-search-tree/>
- C. **FALSE** Depth first search can be used to find connected components of a graph.

- D. **FALSE** Infix + (postfix or prefix) is req. to construct the binary tree uniquely.

7 votes

-- Lokesha Dafale (11.2k points)

3.6

Graphs(6) top

3.6.1 Graphs: GATE1992-03,iii top

<https://gateoverflow.in/580>

How many edges can there be in a forest with p components having n vertices in all?

gate1992 data-structure graphs easy

Answer

3.6.2 Graphs: GATE1997-6.2 top

<https://gateoverflow.in/2258>

Let G be the graph with 100 vertices numbered 1 to 100. Two vertices i and j are adjacent if $|i - j| = 8$ or $|i - j| = 12$. The number of connected components in G is

- A. 8
- B. 4
- C. 12
- D. 25

gate1997 data-structure normal graphs

Answer

3.6.3 Graphs: GATE2008-42 top

<https://gateoverflow.in/1872>

G is a graph on n vertices and $2n - 2$ edges. The edges of G can be partitioned into two edge-disjoint spanning trees. Which of the following is NOT true for G ?

- A. For every subset of k vertices, the induced subgraph has at most $2k - 2$ edges.
- B. The minimum cut in G has at least 2 edges.
- C. There are at least 2 edge-disjoint paths between every pair of vertices.
- D. There are at least 2 vertex-disjoint paths between every pair of vertices.

gate2008 data-structure graphs normal

Answer

3.6.4 Graphs: GATE2008-IT-4 top

<https://gateoverflow.in/3264>

What is the size of the smallest MIS (Maximal Independent Set) of a chain of nine nodes?

- A. 5
- B. 4
- C. 3
- D. 2

gate2008-it data-structure normal graphs

Answer

3.6.5 Graphs: GATE2014-1-3 top

<https://gateoverflow.in/1754>

Let $G = (V, E)$ be a directed graph where V is the set of vertices and E the set of edges. Then

which one of the following graphs has the same strongly connected components as G ?

- A. $G_1 = (V, E_1)$ where $E_1 = \{(u, v) \mid (u, v) \notin E\}$
- B. $G_2 = (V, E_2)$ where $E_2 = \{(u, v) \mid (v, u) \in E\}$
- C. $G_3 = (V, E_3)$ where $E_3 = \{(u, v) \mid \text{there is a path of length } \leq 2 \text{ from } u \text{ to } v \text{ in } E\}$
- D. $G_4 = (V_4, E)$ where V_4 is the set of vertices in G which are not isolated

gate2014-1 data-structure graphs ambiguous

Answer

3.6.6 Graphs: GATE2016-1-38 top

<https://gateoverflow.in/39731>

Consider the weighted undirected graph with 4 vertices, where the weight of edge $\{i, j\}$ is given by the entry W_{ij} in the matrix W .

$$W = \begin{bmatrix} 0 & 2 & 8 & 5 \\ 2 & 0 & 5 & 8 \\ 8 & 5 & 0 & x \\ 5 & 8 & x & 0 \end{bmatrix}$$

The largest possible integer value of x , for which at least one shortest path between some pair of vertices will contain the edge with weight x is _____.

gate2016-1 data-structure graphs normal numerical-answers

Answer

Answers: Graphs

3.6.1 Graphs: GATE1992-03,iii top

<https://gateoverflow.in/580>



Answer: $n - p$.

There are p components and n vertices. A component can have any no. of vertices but it must be a tree.

Let the no. of vertices in 1st component be x_1 , in 2nd component x_2 , in 3rd component x_3 , and so on. For the p th component, there will be x_p vertices.

In a tree with

n vertices, there are exactly $n - 1$ edges.

So, In 1st component $\Rightarrow (x_1 - 1)$ edges.

2nd component $\Rightarrow (x_2 - 1)$ edges.

:

In p th component $\Rightarrow (x_p - 1)$ edges.

Therefore, total no. of edges $= [(x_1 + x_2 + x_3 + \dots + x_p) - p]$

$(x_1 + x_2 + x_3 + \dots + x_p) = n$ (total no. of vertices).

So, total no. of edges is $(n - p)$.

20 votes

-- Hemant Parihar (14.8k points)

3.6.2 Graphs: GATE1997-6.2 top

<https://gateoverflow.in/2258>

Selected Answer

From the description it is clear that vertices are connected as follows:

1 – 9 – 17 – ... – 97
 2 – 10 – 18 – ... – 98
 3 – 11 – 19 – ... – 99
 4 – 12 – 20 – ... – 100
 5 – 13 – 21 – ... – 93
 6 – 14 – 22 – ... – 94
 7 – 15 – 23 – ... – 95
 8 – 16 – 24 – ... – 96

We have covered all vertices using 8 vertex sets considering only $|i - j| = 8$. Using $|i - j| = 12$ we can see the vertex 1 is connected to 13, 2 – 14, 3 – 15 and 4 – 16, so the top 4 vertex sets are in fact connected to the bottom 4 sets, thus reducing the connected components to 4.

35 votes

-- Arjun Suresh (350k points)

3.6.3 Graphs: GATE2008-42 top

<https://gateoverflow.in/1872>

Selected Answer

There are 2 spanning trees (a spanning tree connects all n vertices) for G which are edge disjoint. A spanning tree for n nodes require $n - 1$ edges and so 2 edge-disjoint spanning trees requires $2n - 2$ edges. As G has only $2n - 2$ edges, it is clear that it doesn't have any edge outside that of the two spanning trees. Now lets see the cases:

Lets take any subgraph of G with k vertices. The remaining subgraph will have $n - k$ vertices. Between these two subgraphs (provided both has at least one vertex) there should be at least 2 edges, as we have 2 spanning trees in G . So, (B) is TRUE. Also, in the subgraph with k vertices, we cannot have more than $2k - 2$ edges as this would mean that in the other subgraph with $n - k$ vertices, we would have less than $2n - 2k$ edges, making 2 spanning trees impossible in it. So, (A) is also TRUE.

A spanning tree covers all the vertices. So, 2 edge-disjoint spanning trees in G means, between every pair of vertices in G we have two edge-disjoint paths (length of paths may vary). So, (C) is also TRUE.

So, that leaves option (D) as answer. It is not quite hard to give a counter example for (D).

29 votes

-- Arjun Suresh (350k points)

3.6.4 Graphs: GATE2008-IT-4 top

<https://gateoverflow.in/3264>

Selected Answer

Answer: C

1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

(2,5,8) is the maximal independent set for a chain of 9 nodes. If we add any other node to the set then it will not be MIS.

26 votes

-- Rajarshi Sarkar (34.1k points)

3.6.5 Graphs: GATE2014-1-3 top

<https://gateoverflow.in/1754>



Selected Answer

(A) is false. Consider just two vertices connected to each other. So, we have one SCC. The new graph won't have any edges and so 2 SCC.

(B) is true. In a directed graph an SCC will have a path from each vertex to every other vertex. So, changing the direction of all the edges, won't change the SCC.

(D) is false. Consider any graph with isolated vertices- we loose those components.

(C) is a bit tricky. Any edge is a path of length 1. So, the new graph will have all the edges from old one. Also, we are adding new edges (u, v) . So, does this modify any SCC? No, because we add an edge (u, v) , only if there is already a path of length ≤ 2 from u to v - so we do not create a new path. So, both (B) and (C) must answer, though GATE key says only B.

36 votes

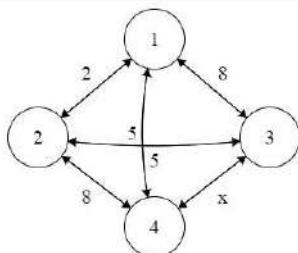
-- Arjun Suresh (350k points)

3.6.6 Graphs: GATE2016-1-38 top

<https://gateoverflow.in/39731>



Selected Answer



Let us list down the shortest edge between each pair of vertices x, y in the graph

x	y	shortest path from x to y
1	2	2
1	3	there is a direct path from 1 to 3 of weight 8. We can also chose to go via node 4 with total path weight $5 + x$. If $5 + x < 8$ ($x < 3$) then shortest path is $5 + x$ otherwise shortest path is the weight of the direct path i.e 8
1	4	5
2	3	5

		shortest path from to
2	4	there is a direct path from 2 to 4 of weight 8. We can also choose to go via node 3 with total path weight $5 + x$. If $5 + x < 8$ ($x < 3$) then shortest path is $5 + x$ otherwise shortest path is the weight of the direct path i.e 8
3	4	We can choose to go through the direct path of weight x or via node 2 with weight $5 + 8 = 13$. If $x < 13$ then we will choose x to be the shortest path otherwise 13 is the shortest path

- **Case 1:** Let us say $x < 3$. Say $x = 2$.

When we put $x = 2$ the above table is modified as

x	y	shortest path from x to y
1	2	2
1	3	7
1	4	5
2	3	5
2	4	7
3	4	2 // Note that the shortest path between node 3 and 4 is $x = 2$

- **Case 2:** $3 \leq x < 13$. Let's say $x = 12$. The table is now modified as

x	y	shortest path from x to y
1	2	2
1	3	8
1	4	5
2	3	5
2	4	8
3	4	12 // Note that the shortest path between node 3 and 4 is $x = 12$

Now the question asks you to find the largest possible integer value of x such that shortest path between atleast one pair of nodes in the graph is equal to x . for values $x = 2, 3, 4, \dots, 12$ the shortest path between node 3 and 4 is equal to x .

The largest among this is $x = 12$. So the answer is 12

40 votes

-- janakyMurthy (1k points)

3.7

Hashing(17) top

3.7.1 Hashing: GATE1989-1-vii, ISRO2015-14 top

<https://gateoverflow.in/10905>

A hash table with ten buckets with one slot per bucket is shown in the following figure. The symbols S_1 to S_7 initially entered using a hashing function with linear probing. The maximum number of comparisons needed in searching an item that is not present is

0	S_7
1	S_1
2	
3	S_4
4	S_2
5	
6	S_5
7	
8	S_6
9	S_3

- A. 4
- B. 5
- C. 6
- D. 3

hashing isro2015 gate1989 data-structure normal

[Answer](#)

3.7.2 Hashing: GATE1996-1.13 [top](#)

<https://gateoverflow.in/2717>

An advantage of chained hash table (external hashing) over the open addressing scheme is

- A. Worst case complexity of search operations is less
- B. Space used is less
- C. Deletion is easier
- D. None of the above

gate1996 data-structure hashing normal

[Answer](#)

3.7.3 Hashing: GATE1996-15 [top](#)

<https://gateoverflow.in/2767>

Insert the characters of the string $K R P C S N Y T J M$ into a hash table of size 10.

Use the hash function

$$h(x) = (\text{ord}(x) - \text{ord}("a")) + 1 \mod 10$$

and linear probing to resolve collisions.

- A. Which insertions cause collisions?
- B. Display the final hash table.

gate1996 data-structure hashing normal

[Answer](#)

3.7.4 Hashing: GATE1997-12 [top](#)

<https://gateoverflow.in/2272>

Consider a hash table with n buckets, where external (overflow) chaining is used to resolve collisions. The hash function is such that the probability that a key value is hashed to a particular bucket is $\frac{1}{n}$. The hash table is initially empty and K distinct values are inserted in the table.

- A. What is the probability that bucket number 1 is empty after the K^{th} insertion?
- B. What is the probability that no collision has occurred in any of the K insertions?
- C. What is the probability that the first collision occurs at the K^{th} insertion?

gate1997 | data-structure | hashing | probability | normal

[Answer](#)

3.7.5 Hashing: GATE2004-7 [top](#)

<https://gateoverflow.in/1004>

Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function $x \bmod 10$, which of the following statements are true?

- I. 9679, 1989, 4199 hash to the same value
 - II. 1471, 6171 hash to the same value
 - III. All elements hash to the same value
 - IV. Each element hashes to a different value
- A. I only
 - B. II only
 - C. I and II only
 - D. III or IV

gate2004 | data-structure | hashing | easy

[Answer](#)

3.7.6 Hashing: GATE2005-IT-16 [top](#)

<https://gateoverflow.in/3761>

A hash table contains 10 buckets and uses linear probing to resolve collisions. The key values are integers and the hash function used is key \% 10 . If the values 43, 165, 62, 123, 142 are inserted in the table, in what location would the key value 142 be inserted?

- A. 2
- B. 3
- C. 4
- D. 6

gate2005-it | data-structure | hashing | easy

[Answer](#)

3.7.7 Hashing: GATE2006-IT-20 [top](#)

<https://gateoverflow.in/3559>

Which of the following statement(s) is TRUE?

- I. A hash function takes a message of arbitrary length and generates a fixed length code.
 - II. A hash function takes a message of fixed length and generates a code of variable length.
 - III. A hash function may give the same hash value for distinct messages.
- A. I only
 - B. II and III only

- C. I and III only
D. II only

gate2006-it data-structure hashing normal

[Answer](#)

3.7.8 Hashing: GATE2007-40 [top](#)

<https://gateoverflow.in/1238>

Consider a hash table of size seven, with starting index zero, and a hash function $(3x + 4) \bmod 7$. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing? Note that – denotes an empty location in the table.

- A. 8, –, –, –, –, –, 10
B. 1, 8, 10, –, –, –, 3
C. 1, –, –, –, –, –, 3
D. 1, 10, 8, –, –, –, 3

gate2007 data-structure hashing easy

[Answer](#)

3.7.9 Hashing: GATE2007-IT-28 [top](#)

<https://gateoverflow.in/3461>

Consider a hash function that distributes keys uniformly. The hash table size is 20. After hashing of how many keys will the probability that any new key hashed collides with an existing one exceed 0.5.

- A. 5
B. 6
C. 7
D. 10

gate2007-it data-structure hashing probability normal

[Answer](#)

3.7.10 Hashing: GATE2008-IT-48 [top](#)

<https://gateoverflow.in/3358>

Consider a hash table of size 11 that uses open addressing with linear probing. Let $h(k) = k \bmod 11$ be the hash function used. A sequence of records with keys

43 36 92 87 11 4 71 13 14

is inserted into an initially empty hash table, the bins of which are indexed from zero to ten. What is the index of the bin into which the last record is inserted?

- A. 3
B. 4
C. 6
D. 7

gate2008-it data-structure hashing normal

[Answer](#)

3.7.11 Hashing: GATE2009-36 [top](#)

<https://gateoverflow.in/1322>

The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function $h(k) = k \bmod 10$ and linear probing. What is the resultant hash table?

A
0
1
2
22
323
4
515
6
7
818
9

B
0
1
2
12
3
13
4
55
6
7
818
9

C
0
1
2
12
3
13
4
42
5
3
6
23
7
75
8
18
9
15

D
0
1
2
12
3
13, 3, 23
4
5, 15
6
7
818
9

[gate2009](https://gateoverflow.in/1322) [data-structure](#) [hashing](#) [normal](#)

Answer

3.7.12 Hashing: GATE2010-52 [top](#)

<https://gateoverflow.in/2360>

A hash table of length 10 uses open addressing with hash function $h(k) = k \bmod 10$, and linear probing. After inserting 6 values into an empty hash table, the table is shown as below

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

- A. 46, 42, 34, 52, 23, 33
- B. 34, 42, 23, 52, 33, 46
- C. 46, 34, 42, 23, 52, 33
- D. 42, 46, 33, 23, 34, 52

[gate2010](https://gateoverflow.in/2360) [data-structure](#) [hashing](#) [normal](#)

Answer

3.7.13 Hashing: GATE2010-53 [top](#)

<https://gateoverflow.in/43327>

A hash table of length 10 uses open addressing with hash function $h(k) = k \bmod 10$, and linear probing. After inserting 6 values into an empty hash table, the table is shown as below

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

How many different insertion sequences of the key values using the same hash function and linear probing will result in the hash table shown above?

- A. 10
- B. 20
- C. 30
- D. 40

data-structure hashing difficult gate2010

[Answer](#)

3.7.14 Hashing: GATE2014-1-40 [top](#)

<https://gateoverflow.in/1918>

Consider a hash table with 9 slots. The hash function is $h(k) = k \bmod 9$. The collisions are resolved by chaining. The following 9 keys are inserted in the order: 5, 28, 19, 15, 20, 33, 12, 17, 10. The maximum, minimum, and average chain lengths in the hash table, respectively, are

- A. 3, 0, and 1
- B. 3, 3, and 3
- C. 4, 0, and 1
- D. 3, 0, and 2

gate2014-1 data-structure hashing normal

[Answer](#)

3.7.15 Hashing: GATE2014-3-40 [top](#)

<https://gateoverflow.in/2074>

Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 3 slots are unfilled after the first 3 insertions?

- A. $(97 \times 97 \times 97)/100^3$
- B. $(99 \times 98 \times 97)/100^3$
- C. $(97 \times 96 \times 95)/100^3$
- D. $(97 \times 96 \times 95)/(3! \times 100^3)$

gate2014-3 data-structure hashing probability normal

[Answer](#)

3.7.16 Hashing: GATE2015-2-33 top

<https://gateoverflow.in/8152>

Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2020?

- A. $h(i) = i^2 \bmod 10$
- B. $h(i) = i^3 \bmod 10$
- C. $h(i) = (11 * i^2) \bmod 10$
- D. $h(i) = (12 * i^2) \bmod 10$

gate2015-2 data-structure hashing normal

Answer

3.7.17 Hashing: GATE2015-3-17 top

<https://gateoverflow.in/8414>

Given that hash table T with 25 slots that stores 2000 elements, the load factor a for T is _____.

gate2015-3 data-structure hashing normal numerical-answers

Answer

Answers: Hashing

3.7.1 Hashing: GATE1989-1-vii, ISRO2015-14 top

<https://gateoverflow.in/10905>



Selected Answer

No of comparison in worst case for an element not in hash table is size of largest cluster +1. This is because the probe stops as soon as an empty slot is found (we r using linear probing here).

Size of largest cluster is 4 (S_6, S_3, S_7, S_1)

No of comparison is $4 + 1 = 5$

Like 43 votes

-- Digvijay (54.9k points)

3.7.2 Hashing: GATE1996-1.13 top

<https://gateoverflow.in/2717>



Selected Answer

- A. False :- search operation can go worst in chaining if all elements are stored under a single bucket.
- B. False . Pointer space is overhead in chaining.
- C. is true BCZ in Open Addressing sometimes though element is present we cant delete it if Empty Bucket comes in between while searching for that element ;Such Limitation is not there in Chaining.

Like 13 votes

-- Rajesh Pradhan (22.6k points)

3.7.3 Hashing: GATE1996-15 top

<https://gateoverflow.in/2767>



Selected Answer

Here $Order(x) - Order('a')$ means count difference between that character & a .

Assuming $a = 0, b = 1$ & so on.

- J & M cause collision.
- Final Hash Table

Index	Key
0	T
1	K
2	J
3	C
4	N
5	Y
6	P
7	M
8	R
9	S

16 votes

-- Akash Kanase (42.5k points)

3.7.4 Hashing: GATE1997-12 top

<https://gateoverflow.in/2272>



- A. probability that other buckets are selected $= (n - 1)/n$
 this should happen k times and each of k events are independent so $(n - 1)^k / n^k$
- B. when $k = 1$ prob of no collision $= n/n$
 for $k = 2$ prob of no collision $= n/n * (n - 1)/n$
 for $k = k$ prob of no collision $= n/n * (n - 1)/n * (n - 2)/n ... * (n - k + 1)/n$ for $k \leq n$
 for $k > n$ prob = 0
- C. prob of collision at $k = 1 = (k - 1)/n$
 prob of collision at $k = 2 = n/n * (k - 1)/n$
 prob of collision at $k = 3 = n/n * (n - 1)/n * (k - 1)/n$
 prob of collision at $k = k = n/n * (n - 1)/n * (n - 2)/n ... * 2/n * (k - 1)/n$ for $k \leq n$
 for $k > n$ prob = 1

19 votes

-- Danish (3.8k points)

3.7.5 Hashing: GATE2004-7 top

<https://gateoverflow.in/1004>



Option C is correct answer because the last digit of every digit given is equal in I and II.

12 votes

-- Bhagirathi Nayak (14.1k points)

3.7.6 Hashing: GATE2005-IT-16 top

<https://gateoverflow.in/3761>



43 in loc 3

165 in loc 5

62 in loc 2

123 in loc 4 (collision and next free space)

142 in loc 6 (collision in 2, and 3,4,5 already occupied)

hence answer **D**

17 votes

-- Sankaranarayanan P.N (42.5k points)

3.7.7 Hashing: GATE2006-IT-20 top

<https://gateoverflow.in/3559>



Answer is (**C**).

I. A hash function takes a message of arbitrary length and generates a fixed length code.. This is correct, this is directly from definition of hash function.

Ref: https://en.wikipedia.org/wiki/Hash_function

II. As I is correct II is wrong !

III. This is true. example: Hash function $N \% 10$, this will generate same values for 1 as well as 11!

(Even in cryptographic hash functions collision happens, just that it is not easy to find colliding instances!)

20 votes

-- Akash Kanase (42.5k points)

3.7.8 Hashing: GATE2007-40 top

<https://gateoverflow.in/1238>



The answer is (**B**).

1 will occupy location 0, 3 will occupy location 6, 8 hashed to location 0 which is already occupied so, it will be hashed to one location next to it. i.e. to location 1.

Since 10 also clashes, so it will be hashed to location 2.

17 votes

-- Gate Keeda (19.6k points)

3.7.9 Hashing: GATE2007-IT-28 top

<https://gateoverflow.in/3461>



The question is a bit ambiguous.

After hashing of how many keys, will the probability that any new key hashed collides with an existing one exceed 0.5.

Here, 'new key hashed' is the ambiguity. It can mean the probability of a collision in the next 'hash', or the probability of a collision in any of the hashes of the 'new keys' starting from the first insertion. For the first case answer must be 10 to get probability equal to 0.5, and so 11 must be the answer for probability > 0.5 . Thus we can conclude from given choices, it is the second case.

So, we need to find n such that after n hashes, probability of collision (in any of the n hashes) > 0.5 .

Probability that there will be a collision after n hashes (a collision happened in at least one of those n hashes) = $1 - \text{Probability that there was no collision in the first } n \text{ hashes}$

$$= 1 - 1 \cdot \frac{19}{20} \cdot \frac{18}{20} \cdots \frac{20-n+1}{20} .$$

So, we need,

$$0.5 < 1 - 1 \cdot \frac{19}{20} \cdot \frac{18}{20} \cdots \frac{20-n+1}{20} .$$

$$\implies \frac{19}{20} \cdot \frac{18}{20} \cdots \frac{20-n+1}{20} < 0.5 .$$

For $n = 5$, we get, 0.5814 and for $n = 6$, we get 0.43605. So, answer should be $n = 6$.

1 31 votes

-- Arjun Suresh (350k points)

3.7.10 Hashing: GATE2008-IT-48 top

<https://gateoverflow.in/3358>



Selected Answer

Index	Key
0	87
1	11
2	13
3	36
4	92
5	4
6	71
7	14
8	
9	
10	43

(D) is answer

1 12 votes

-- Prashant Singh (59.8k points)

3.7.11 Hashing: GATE2009-36 top

<https://gateoverflow.in/1322>



Selected Answer

(C) is the correct option ..directly from the definition of linear probing. In linear probing, when a hashed location is already filled, locations are linearly probed until a free one is found.

<http://courses.cs.washington.edu/courses/cse326/00wi/handouts/lecture16/sld015.htm>

Upvote (25 votes)

-- Bhagirathi Nayak (14.1k points)

3.7.12 Hashing: GATE2010-52 top

<https://gateoverflow.in/2360>



Selected Answer

Option (C)

46, 34, 42, 23, 52, 33

46 – position 6

34 position 4

42 position 2

23 position 3

52 position 2 – collision next empty is 5

33 position 3 – collision next empty is 7

Upvote (15 votes)

-- Sankaranarayanan P.N (11.5k points)

3.7.13 Hashing: GATE2010-53 top

<https://gateoverflow.in/43327>



Selected Answer

53 - option (C).

Slots 3, 4, 5 and 6 must be filled before 33 comes. Similarly slots 2, 3 and 4 must be filled before 52 comes. And 52 must come before 33, as it is not occupying slot 2. So, 33 must be at the end and 52 can come at position 4 or 5.

Let 52 come at position 4. Slots 2, 3 and 4 must be filled before 52 leaving only slot 6 left for the element coming at position 5 which should be 46. So, the first 3 elements can come in any order giving $3! = 6$ ways.

Let 52 come at position 5. Now, the first four elements can come in any order. So, $4! = 24$ ways.

So, total number of different insertion sequences possible = $24 + 6 = 30$

Upvote (47 votes)

-- Arjun Suresh (350k points)

answer = **option C**

element **33** has managed to hold position at slot #7 it means elements should occupy slot #3 to slot #6 before it in the sequence. Currently it seems like all element except **42** should come before **33** in the sequence.

But, element **52** requires that **42** comes before it. and **33** requires that **52** comes before it. This means that **42** has to come before **33**. this makes element **33** to occupy last position in the sequence.

now for element **52** to occupy its place these can be two cases :

- Case 1 : $\boxed{\{42, 23, 34\}}, 52$
- Case 2: $\boxed{\{42, 23, 34, 46\}}, 52$

Case 1 means that those three elements comes before **52** = $3! = 6$ ways
Case 2 means that those four elements comes before **52** = $4! = 24$ ways

Combining all info we get,

Total number of sequences possible that will form the same hash table as above
 $= (6 + 24) \times 1 = 30$

28 votes

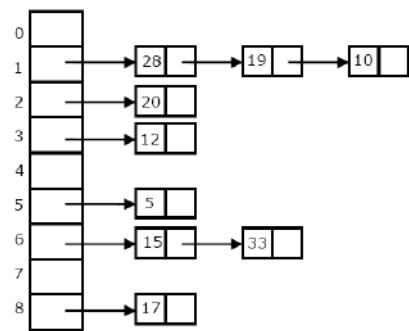
-- Amar Vashishth (30.5k points)

3.7.14 Hashing: GATE2014-1-40 top

<https://gateoverflow.in/1918>



Selected Answer



So, Maximum & minimum chain lengths are **3** & **0** respectively.

Average chain length = $(0 + 3 + 1 + 1 + 0 + 1 + 2 + 0 + 1)/9 = 1$.

So, Answer is A.

26 votes

-- Jay (1.1k points)

3.7.15 Hashing: GATE2014-3-40 top

<https://gateoverflow.in/2074>



Selected Answer

We have **100** slots each of which are picked with equal probability by the hash function (since hashing is uniform). So, to avoid first **3** slots, the hash function has to pick from the remaining **97** slots. And repetition is allowed, since chaining is used- meaning a list of elements are stored in a

slot and not a single element.

$$\text{So, required probability} = \frac{97}{100} \times \frac{97}{100} \times \frac{97}{100}$$

$$= (97 \times 97 \times 97)/100^3$$

47 votes

-- Arjun Suresh (350k points)

3.7.16 Hashing: GATE2015-2-33 top

<https://gateoverflow.in/8152>



Since mod 10 is used, the last digit matters.

If we CUBE all numbers from 0 to 9, we get the following

Number	Cube	Last Digit in Cube
0	0	0
1	1	1
2	8	8
3	27	7
4	64	4
5	125	5
6	216	6
7	343	3
8	512	2
9	729	9

Therefore all numbers from 0 to 2020 are equally divided in 10 buckets. If we make a table for square, we won't get equal distribution. In the following table. 1, 4, 6 and 9 are repeated, so these buckets would have more entries and there are no buckets corresponding to 2, 3, 7 and 8.

Number	Square	Last Digit in Cube
0	0	0
1	1	1
2	4	4
3	9	9
4	16	6
5	25	5
6	36	6
7	49	9
8	64	4
9	81	1

<http://geeksquiz.com/gate-gate-cs-2015-set-2-question-43/>

48 votes

-- Anu (5.9k points)

3.7.17 Hashing: GATE2015-3-17 top

<https://gateoverflow.in/8414>



A critical statistic for a hash table is the **load factor**, that is the number of entries divided by the number of buckets:

$$\text{Load factor} = n/k$$

where:

n = number of entries

k = number of buckets

As the load factor grows larger, the hash table becomes slower, and it may even fail to work

(depending on the method used).

Here, load factor = $2000/25 = 80$

22 votes

-- Akash Kanase (42.5k points)

3.8

Heap(24) top

3.8.1 Heap: GATE1996-2.11 top

<https://gateoverflow.in/2740>

The minimum number of interchanges needed to convert the array into a max-heap is

89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70

- A. 0
- B. 1
- C. 2
- D. 3

gate1996 data-structure heap easy

[Answer](#)

3.8.2 Heap: GATE1999-12 top

<https://gateoverflow.in/1511>

- A. In binary tree, a full node is defined to be a node with 2 children. Use induction on the height of the binary tree to prove that the number of full nodes plus one is equal to the number of leaves.
- B. Draw the min-heap that results from insertion of the following elements in order into an initially empty min-heap: 7, 6, 5, 4, 3, 2, 1. Show the result after the deletion of the root of this heap.

gate1999 data-structure heap normal

[Answer](#)

3.8.3 Heap: GATE2001-1.15 top

<https://gateoverflow.in/708>

Consider any array representation of an n element binary heap where the elements are stored from index 1 to index n of the array. For the element stored at index i of the array ($i \leq n$), the index of the parent is

- A. $i - 1$
- B. $\lfloor \frac{i}{2} \rfloor$
- C. $\lceil \frac{i}{2} \rceil$
- D. $\frac{(i+1)}{2}$

gate2001 data-structure heap easy

[Answer](#)

3.8.4 Heap: GATE2003-23 top

<https://gateoverflow.in/1110>

In a min-heap with n elements with the smallest element at the root, the 7^{th} smallest element can be found in time

- A. $\Theta(n \log n)$
- B. $\Theta(n)$
- C. $\Theta(\log n)$
- D. $\Theta(1)$

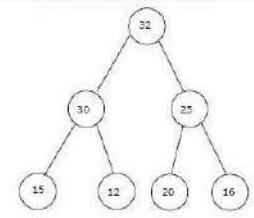
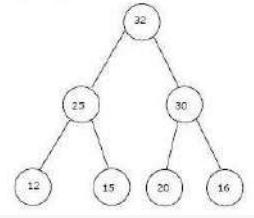
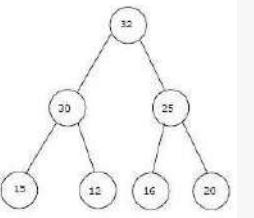
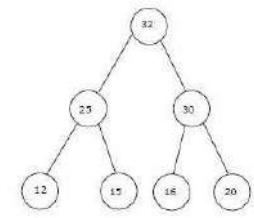
gate2003 data-structure heap

Answer

3.8.5 Heap: GATE2004-37 [top](#)

<https://gateoverflow.in/1034>

The elements 32, 15, 20, 30, 12, 25, 16, are inserted one by one in the given order into a maxHeap. The resultant maxHeap is

- A. 
- B. 
- C. 
- D. 

gate2004 data-structure heap normal

Answer

3.8.6 Heap: GATE2004-IT-53 [top](#)

<https://gateoverflow.in/3696>

An array of integers of size n can be converted into a heap by adjusting the heaps rooted at each internal node of the complete binary tree starting at the node $\lfloor (n-1)/2 \rfloor$, and doing this adjustment up to the root node (root node is at index 0) in the order $\lfloor (n-1)/2 \rfloor, \lfloor (n-3)/2 \rfloor, \dots, 0$. The time required to construct a heap in this manner is

- A. $O(\log n)$
- B. $O(n)$
- C. $O(n \log \log n)$

- D. $O(n \log n)$

gate2004-it data-structure heap normal

[Answer](#)

3.8.7 Heap: GATE2005-34 [top](#)

<https://gateoverflow.in/1370>

A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is: 10, 8, 5, 3, 2. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

- A. 10, 8, 7, 5, 3, 2, 1
- B. 10, 8, 7, 2, 3, 1, 5
- C. 10, 8, 7, 1, 2, 3, 5
- D. 10, 8, 7, 3, 2, 1, 5

gate2005 data-structure heap normal

[Answer](#)

3.8.8 Heap: GATE2006-10 [top](#)

<https://gateoverflow.in/889>

In a binary max heap containing n numbers, the smallest element can be found in time

- A. $O(n)$
- B. $O(\log n)$
- C. $O(\log \log n)$
- D. $O(1)$

gate2006 data-structure heap easy

[Answer](#)

3.8.9 Heap: GATE2006-76 [top](#)

<https://gateoverflow.in/1852>

Statement for Linked Answer Questions 76 & 77:

A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows: The root is stored in the first location, $a[0]$, nodes in the next level, from left to right, is stored from $a[1]$ to $a[3]$. The nodes from the second level of the tree from left to right are stored from $a[4]$ location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location $a[n]$ and pushing it up the tree to satisfy the heap property.

76. Which one of the following is a valid sequence of elements in an array representing 3-ary max heap?

- A. 1, 3, 5, 6, 8, 9
- B. 9, 6, 3, 1, 8, 5
- C. 9, 3, 6, 8, 5, 1
- D. 9, 5, 6, 8, 3, 1

gate2006 data-structure heap normal

[Answer](#)

3.8.10 Heap: GATE2006-77 [top](#)

<https://gateoverflow.in/87191>

Statement for Linked Answer Questions 76 & 77:

A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows: The root is stored in the first location, $a[0]$, nodes in the next level, from left to right, is stored from $a[1]$ to $a[3]$. The nodes from the second level of the tree from left to right are stored from $a[4]$ location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location $a[n]$ and pushing it up the tree to satisfy the heap property.

77. Suppose the elements 7, 2, 10 and 4 are inserted, in that order, into the valid 3-ary max heap found in the previous question, Q.76. Which one of the following is the sequence of items in the array representing the resultant heap?

- A. 10, 7, 9, 8, 3, 1, 5, 2, 6, 4
- B. 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
- C. 10, 9, 4, 5, 7, 6, 8, 2, 1, 3
- D. 10, 8, 6, 9, 7, 2, 3, 4, 1, 5

[gate2006](#) [data-structure](#) [heap](#) [normal](#)

[Answer](#)

3.8.11 Heap: GATE2006-IT-44 [top](#)

<https://gateoverflow.in/3587>

Which of the following sequences of array elements forms a heap?

- A. {23, 17, 14, 6, 13, 10, 1, 12, 7, 5}
- B. {23, 17, 14, 6, 13, 10, 1, 5, 7, 12}
- C. {23, 17, 14, 7, 13, 10, 1, 5, 6, 12}
- D. {23, 17, 14, 7, 13, 10, 1, 12, 5, 7}

[gate2006-it](#) [data-structure](#) [heap](#) [easy](#)

[Answer](#)

3.8.12 Heap: GATE2006-IT-72 [top](#)

<https://gateoverflow.in/3616>

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0. If only the root node does not satisfy the heap property, the algorithm to convert the complete binary tree into a heap has the best asymptotic time complexity of

- A. $O(n)$
- B. $O(\log n)$
- C. $O(n \log n)$
- D. $O(n \log \log n)$

[gate2006-it](#) [data-structure](#) [heap](#) [easy](#)

[Answer](#)

3.8.13 Heap: GATE2007-47 [top](#)

<https://gateoverflow.in/1245>

Consider the process of inserting an element into a *Max Heap*, where the *Max Heap* is represented by an *array*. Suppose we perform a binary search on the path from the new leaf to the

root to find the position for the newly inserted element, the number of *comparisons* performed is:

- A. $\Theta(\log_2 n)$
- B. $\Theta(\log_2 \log_2 n)$
- C. $\Theta(n)$
- D. $\Theta(n \log_2 n)$

gate2007 data-structure heap normal

[Answer](#)

3.8.14 Heap: GATE2009-59 [top](#)

<https://gateoverflow.in/1341>

Consider a binary max-heap implemented using an array.

Which one of the following array represents a binary max-heap?

- A. {25, 12, 16, 13, 10, 8, 14}
- B. {25, 14, 13, 16, 10, 8, 12}
- C. {25, 14, 16, 13, 10, 8, 12}
- D. {25, 14, 12, 13, 10, 8, 16}

gate2009 data-structure heap normal

[Answer](#)

3.8.15 Heap: GATE2009-60 [top](#)

<https://gateoverflow.in/43466>

Consider a binary max-heap implemented using an array.

What is the content of the array after two delete operations on {25, 14, 16, 13, 10, 8, 12}

- A. {14, 13, 12, 10, 8}
- B. {14, 12, 13, 8, 10}
- C. {14, 13, 8, 12, 10}
- D. {14, 13, 12, 8, 10}

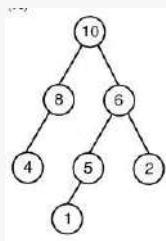
gate2009 data-structure heap normal

[Answer](#)

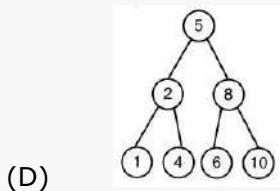
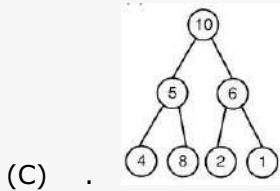
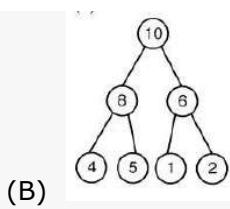
3.8.16 Heap: GATE2011-23 [top](#)

<https://gateoverflow.in/2125>

A max-heap is a heap where the value of each parent is greater than or equal to the value of its children. Which of the following is a max-heap?



(A)



gate2011 data-structure heap easy

[Answer](#)

3.8.17 Heap: GATE2014-2-12 [top](#)

<https://gateoverflow.in/1967>

A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is: 10, 8, 5, 3, 2. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

- A. 10, 8, 7, 3, 2, 1, 5
- B. 10, 8, 7, 2, 3, 1, 5
- C. 10, 8, 7, 1, 2, 3, 5
- D. 10, 8, 7, 5, 3, 2, 1

gate2014-2 data-structure heap normal

[Answer](#)

3.8.18 Heap: GATE2015-1-32 [top](#)

<https://gateoverflow.in/8273>

Consider a max heap, represented by the array: 40, 30, 20, 10, 15, 16, 17, 8, 4 .

Array index	1	2	3	4	5	6	7	8	9
Value	40	30	20	10	15	16	17	8	4

Now consider that a value 35 is inserted into this heap. After insertion, the new heap is

- A. 40, 30, 20, 10, 15, 16, 17, 8, 4, 35
- B. 40, 35, 20, 10, 30, 16, 17, 8, 4, 15
- C. 40, 30, 20, 10, 35, 16, 17, 8, 4, 15
- D. 40, 35, 20, 10, 15, 16, 17, 8, 4, 30

gate2015-1 data-structure heap easy

[Answer](#)

3.8.19 Heap: GATE2015-2-17 [top](#)

<https://gateoverflow.in/8091>

Consider a complete binary tree where the left and right subtrees of the root are max-heaps. The lower bound for the number of operations to convert the tree to a heap is

- A. $\Omega(\log n)$
- B. $\Omega(n)$
- C. $\Omega(n \log n)$
- D. $\Omega(n^2)$

gate2015-2 data-structure heap normal

[Answer](#)

3.8.20 Heap: GATE2015-3-19 [top](#)

<https://gateoverflow.in/8418>

Consider the following array of elements.

$\langle 89, 19, 50, 17, 12, 15, 2, 5, 7, 11, 6, 9, 100 \rangle$

The minimum number of interchanges needed to convert it into a max-heap is

- A. 4
- B. 5
- C. 2
- D. 3

gate2015-3 data-structure heap normal

[Answer](#)

3.8.21 Heap: GATE2016-1-37 [top](#)

<https://gateoverflow.in/39706>

An operator $\text{delete}(i)$ for a binary heap data structure is to be designed to delete the item in the i -th node. Assume that the heap is implemented in an array and i refers to the i -th index of the array. If the heap tree has depth d (number of edges on the path from the root to the farthest leaf), then what is the time complexity to re-fix the heap efficiently after the removal of the element?

- A. $O(1)$
- B. $O(d)$ but not $O(1)$
- C. $O(2^d)$ but not $O(d)$
- D. $O(d 2^d)$ but not $O(2^d)$

gate2016-1 data-structure heap normal

[Answer](#)

3.8.22 Heap: GATE2016-2-34 [top](#)

<https://gateoverflow.in/39585>

A complete binary min-heap is made by including each integer in $[1, 1023]$ exactly once. The depth of a node in the heap is the length of the path from the root of the heap to that node. Thus, the root is at depth 0. The maximum depth at which integer 9 can appear is _____.

gate2016-2 data-structure heap normal numerical-answers

[Answer](#)

3.8.23 Heap: ISI2014-CS-2b [top](#)

<https://gateoverflow.in/47438>

Let H_1 and H_2 be two complete binary trees that are heaps as well. Assume H_1 and H_2 are max-heaps, each of size n . Design and analyze an efficient algorithm to merge H_1 and H_2 to a new

max-heap H of size $2n$.

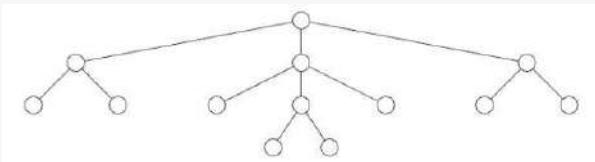
[descriptive](#) [isi2014](#) [algorithms](#) [binary-tree](#) [heap](#)

[Answer](#)

3.8.24 Heap: TIFR2014-B-19 [top](#)

<https://gateoverflow.in/27352>

Consider the following tree with 13 nodes.



Suppose the nodes of the tree are randomly assigned distinct labels from $\{1, 2, \dots, 13\}$, each permutation being equally likely. What is the probability that the labels form a min-heap (i.e., every node receives the minimum label in its subtree)?

- A. $(\frac{1}{6!})(\frac{1}{3!})^2$
- B. $(\frac{1}{3!})^2(\frac{1}{2!})^3$
- C. $(\frac{1}{13})(\frac{1}{6})(\frac{1}{3})^3$
- D. $\frac{2}{13}$
- E. $\frac{1}{2^{13}}$

[tifr2014](#) [heap](#)

[Answer](#)

Answers: Heap

3.8.1 Heap: GATE1996-2.11 [top](#)

<https://gateoverflow.in/2740>



Selected Answer

"The minimum number of interchanges needed to convert the array
89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70
into a heap with the maximum element at the root node is:"

This is the correction.

Answer: C.

Only element 70 violates the rule. Hence, it must be shifted to its proper position.

Step1: $swap(10, 70)$

Step2: $swap(40, 70)$

Hence, only 2 interchanges are required.

19 votes

-- Gate Keeda (19,6k points)

3.8.2 Heap: GATE1999-12 [top](#)

<https://gateoverflow.in/1511>



Selected Answer

a. Note My Convention:-

no. Of full node = F

no. Of leaf node = L

Base Case: $H = 0$.

A binary tree of height 0 is just a single node with no children, and therefore has 1 leaf.

$$F + 1 = L$$

$$0 + 1 = 1$$

so the base case satisfies the induction hypothesis (see below).

Induction Hypothesis (*I.H.*): Suppose that for some $k \geq 0$, all binary trees of height $\leq k$ have $(F + 1) = L$ leaves.

Induction Step: Let T be a binary tree of height $k + 1$. Then T 's left and right subtrees are each binary trees of height $\leq k$, and thus by the *I.H.* both subtree have $(F + 1)$ leaves. The number of leaves in T is equal to the sum of the number of leaves in T 's subtrees,

$$(F + 1)_{\text{left sub tree}} + (F + 1)_{\text{right sub tree}} = L_{\text{left sub tree}} + L_{\text{right sub tree}}$$

$$2F + 2 = 2L$$

$$2(F + 1) = 2(L)$$

$$\therefore F + 1 = L \text{ (proved)}$$

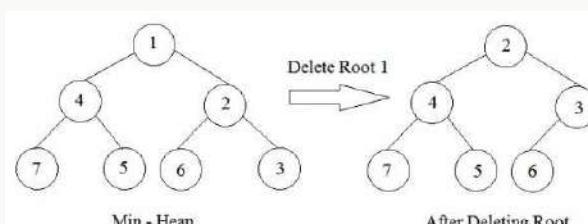
Hence, the hypothesis holds for $k + 1$, and so the theorem is proved.

b. Here in question they mentioned to insert element in given order into an empty Heap.

So here we have to use Insertion Method to create the heap instead of using Heapify Method to build the heap.

Plz refer below img the LHS shows the resultant heap after doing insertion of the keys into initial empty heap.

RHS heap is the result of deletion of root.



13 votes

-- Rajesh Pradhan (22.6k points)

3.8.3 Heap: GATE2001-1.15 top

<https://gateoverflow.in/708>



Selected Answer

for node at index i

left $child(L)$ at $2i$

right $child(R)$ at $2i + 1$

for node at index i

parent will be at floor $i/2$

21 votes

-- Pooja Palod (31.3k points)

3.8.4 Heap: GATE2003-23 [top](#)

<https://gateoverflow.in/1110>



Selected Answer

Time to find the smallest element on a min-heap- one retrieve operation - $\Theta(1)$

Time to find the second smallest element on a min-heap- requires $2^2 - 1 = 3$ check operations to find the second smallest element out of 3 elements - $\Theta(1)$

Time to find the 7th smallest element - requires $O(2^7 - 1) = O(127)$ check operations to find the seventh smallest element out of 127 possible ones - $\Theta(1)$

In short if the number of required operations is independent of the input size n , then it is always $\Theta(1)$.

(Here, we are doing a level order traversal of the heap and checking the elements)

If we are not allowed to traverse the heap and allowed only default heap-operations, we will be restricted with doing Extract-min 7 times which would be $O(\log n)$.

1 53 votes

-- gatecse (18k points)

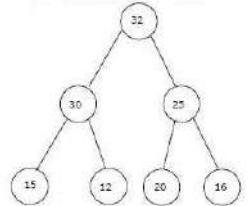
3.8.5 Heap: GATE2004-37 [top](#)

<https://gateoverflow.in/1034>



Selected Answer

Answer is **option A.**



Just keep inserting elements making sure resulting Tree is nearly Complete.(Heap Property) .

While inserting any node, if you find that Value of New Node > Value of its parent, bubble it up to keep Max heap property

1 16 votes

-- Akash Kanase (42.5k points)

3.8.6 Heap: GATE2004-IT-53 [top](#)

<https://gateoverflow.in/3696>



Selected Answer

By using **Build Heap method** we can create heap from complete binary tree which will take $O(n)$.

1 24 votes

-- Sneha Goel (1.1k points)

3.8.7 Heap: GATE2005-34 [top](#)

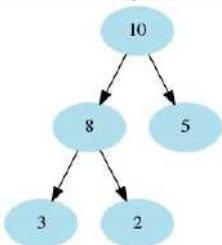
<https://gateoverflow.in/1370>



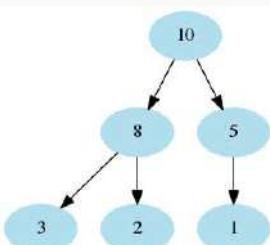
Selected Answer

ans is (D)....whenever we insert an element in heap,it will always inserted in last level from left to right..so here we insert element 1 and 7 as a child of node 5.then we perform heapify algorithm until we get the min/max heap..so here finally in above question we get the heap whose level order traversal is 10,8,7,3,2,1,5

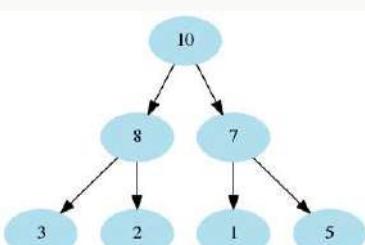
Initial heap:



After insert of 1



After insert of 7



18 votes

-- neha pawar (4.2k points)

3.8.8 Heap: GATE2006-10 top

<https://gateoverflow.in/889>



Selected Answer

A. $O(n)$

In a max heap, the smallest element is always present at a leaf node. Heap being a complete binary tree, there can be up to $\frac{n}{2}$ leaf nodes and to examine all of them we would need $O(n)$ time.

27 votes

-- Keith Kr (6.1k points)

3.8.9 Heap: GATE2006-76 top

<https://gateoverflow.in/1852>

D.

The method to solve is clearly given in the question itself. Just one thing to mention is the heap property. Max Heap = The parent is always greater or equal to the child. if not then swap the respective child with the parent to satisfy the property of the heap.

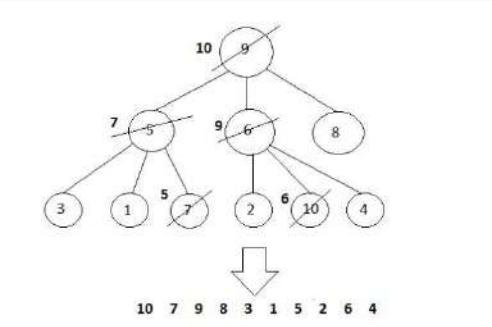
6 votes

-- Gate Keeda (19.6k points)

3.8.10 Heap: GATE2006-77 top

<https://gateoverflow.in/87191>

Selected Answer



Heap will be constructed like this, based on the correct answer of Q.76 (which is
9 5 6 8 3 1)

 12 votes

-- Anurag Pandey (13.6k points)

3.8.11 Heap: GATE2006-IT-44 top

<https://gateoverflow.in/3587>

Selected Answer

For a heap(max heap) parent should be greater than or equal to children. in a heap of $[1..n]$ left child of i th node will be at $2 * i$ th position and right child will be at $2 * i + 1$ position.

So, for given options we can verify it.

Option C seems to be following the property.

 15 votes

-- Sankaranarayanan P.N (11.5k points)

3.8.12 Heap: GATE2006-IT-72 top

<https://gateoverflow.in/3616>

Selected Answer

The question is saying best case which will be when only one swap will be required which will be order of 1.

As no option matches just call heapify at the root - $O(\log n)$.

 17 votes

-- Ravi Singh (15.7k points)

3.8.13 Heap: GATE2007-47 top

<https://gateoverflow.in/1245>

Selected Answer

Number of elements in the path from new leaf to root = $\log n$, and all elements are sorted from leaf to root so we can do a binary search which will result in $O(\log \log n)$ number of comparisons.

Since in heap is a complete binary tree, in an array implementation, from every node index, we can know its depth and this will be the n for binary search.

48 votes

-- Vikrant Singh (13.5k points)

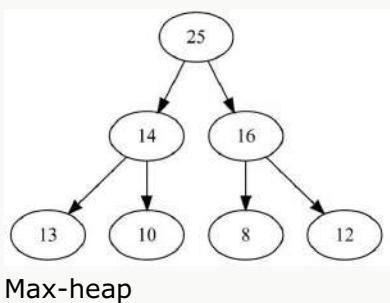
3.8.14 Heap: GATE2009-59 top

<https://gateoverflow.in/1341>



Answer : (C)

The binary max-Heap looks like this :



10 votes

-- Dipak Majhi (979 points)

Taking the given array as level order traversal, we can build binary tree.

- (A) 13 comes as child of 12, which is not allowed in a binary max-heap
- (B) 16 comes as child of 14 violating max-heap property
- (C) is a valid binary max-heap as all children are smaller than their parent
- (D) 16 comes as child of 12, violating max-heap property

17 votes

-- Arjun Suresh (350k points)

3.8.15 Heap: GATE2009-60 top

<https://gateoverflow.in/43466>



During delete, the root element is removed, replaced with the last element and heap property is corrected by pushing the root downwards. So, for first delete,

25 14 16 13 10 8 12 → 12 14 16 13 10 8 → 16 14 12 13 10 8 (the element not satisfying max-heap property is exchanged with the largest of its children) (heap property satisfied)

Second delete:

16 14 12 13 10 8 → 8 14 12 13 10 → 14 8 12 13 10 → 14 13 12 8 10 (heap property satisfied)

<http://homepages.ius.edu/RWISMAN/C455/html/notes/Chapter6/heapify.htm>

23 votes

-- Arjun Suresh (350k points)

3.8.16 Heap: GATE2011-23 top

<https://gateoverflow.in/2125>

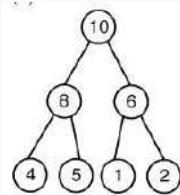
Selected Answer

in option (A) - it is not a max heap because it is not complete binary tree .

in option (C) - it is complete binary tree but not follow the max heap property i.e. the values of parent nodes always greater than child nodes

and there node of value 5 is less than one of its children.

in option (D) - similar to above (C) option explanation here node of value 2 is less than to the value 4 .



correct option is (B) that is satisfy both properties and all of the max heap .

19 votes

-- ASHU2015 (325 points)

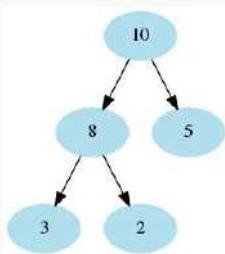
3.8.17 Heap: GATE2014-2-12 top

<https://gateoverflow.in/1967>

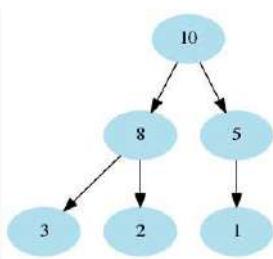
Selected Answer

Answer is (A)....whenever insertion will be done in heap ,it will always inserted in last level from left to right.so we insert 1 and 7 as a child of node 5 now we perform heapify algorithm until heap property will satisfied..and then we get the heap whose level order traversal is 10,8,7,3,2,1,5 .

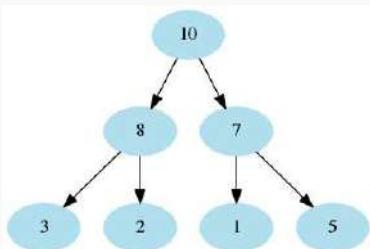
Initial heap



After insert of 1



After insert of 7



17 votes

-- neha pawar (4.2k points)

3.8.18 Heap: GATE2015-1-32 top

<https://gateoverflow.in/8273>



Selected Answer

Heap is complete binary tree. To insert a new element, we put it at the end of the tree and move up towards root till heap property is satisfied. Here, 35 comes as child of 15, with the path 40 – 30 – 15 – 35 . So, we swap 15,35 and then 30,35 to get the new path 40 – 35 – 30 – 15 . So, new heap will be 40 35 20 10 30 16 17 8 4 15.

31 votes

-- Arjun Suresh (350k points)

3.8.19 Heap: GATE2015-2-17 top

<https://gateoverflow.in/8091>



Selected Answer

Answer is (A).

Here, lower bound imply best algorithm which works for all cases and hence we should consider worst-case.

ax-Heapify(root).

38 votes

-- Vikrant Singh (13.5k points)

3.8.20 Heap: GATE2015-3-19 top

<https://gateoverflow.in/8418>



Selected Answer

First interchange 15-100

2nd 50-100

3rd 89-100

So, total interchange 3 so option (D) is correct.

22 votes

-- Anoop Sonkar (5k points)

3.8.21 Heap: GATE2016-1-37 top<https://gateoverflow.in/39706>

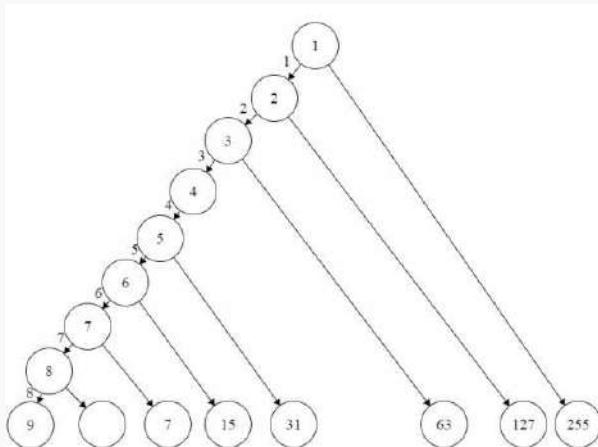
Answer would be (B) $O(d)$ but not $O(1)$.. as we need to apply heapify.. and suppose if we are deleting root, in worst case would take $O(d)$ time..

 42 votes

-- Abhilash Panicker (9.5k points)

3.8.22 Heap: GATE2016-2-34 top<https://gateoverflow.in/39585>

Here answer is 8. With 1024 nodes, we can easily build min heap Check following diagram



Now once we place 1 – 9 then remaining elements can be placed easily to fill up heap (While keeping heap property of course) Total elements we need for this heap is 512, we have given 1024 ! So Yes, 8 is answer !

 49 votes

-- Akash Kanase (42.5k points)

3.8.23 Heap: ISI2014-CS-2b top<https://gateoverflow.in/47438>

First copy the two arrays of H_1 and H_2 into a new array of size $2n$...then apply build heap operation to get H ...Time complexity= $O(2n) = O(n)$

<http://www.geeksforgeeks.org/merge-two-binary-max-heaps/>

 10 votes

-- debanjan sarkar (4k points)

3.8.24 Heap: TIFR2014-B-19 top<https://gateoverflow.in/27352>

(C) is Correct -

$$\text{Probability} = \frac{\text{Number of Favorable Outcomes(minheaps)}}{\text{Total Trees}(13!)}$$

Now, total Min-Heaps :

Firstly we select **minimum element** i.e. **1 for root = 1 way**

We have **3 subtrees of 3, 6, 3 sizes respectively**

First Subtree with 3 elements:

Steps:

1. Choose **3 elements** = ${}^{12}C_3$ **ways**
2. Give minimum to root = **1 way**
3. **2 elements , 2 children(left & right) = 2 ways** { Any of the 2 elements can be given to left or right child}

Second Subtree with 3 elements :

Steps:

1. Choose **3 elements from 9 left elements** = 9C_3 **ways**
2. Similar to above assign these = **2 ways**

Subtree with 6 elements:{ We already have 6 elements left now}

1 Root , 1left_most child , 1 right_most child , 3 in middle **

Steps:

1. Choose root = **1 way** {minimum}
2. Now **5 elements left, choose leftmost child = 5 ways** (choose anyone)
3. Now **4 elements left, choose right_most child = 4 ways** (choose anyone)
4. Now **3 elements left for middle :**
 - a) assign root = **1 way**
 - b) assign left and right anyone = **2 ways**

For, Total Heaps multiply all ways **above.**

$$\begin{aligned}\text{Total probability} &= \frac{\text{Total Heaps}}{\text{Total Trees}(13!)} \\ &= \frac{12! * 2 * 9! * 2 * 5 * 4 * 2}{13! * 3! * 9! * 3! * 6!} \\ &= \left(\frac{1}{13}\right) \left(\frac{1}{6}\right) \left(\frac{1}{3}\right)^3\end{aligned}$$

👍 29 votes

-- Himanshu Agarwal (15.3k points)

3.9

Infix Postfix(2) top

3.9.1 Infix Postfix: GATE1997-1.7 top

<https://gateoverflow.in/2223>

Which of the following is essential for converting an infix expression to the postfix form efficiently?

- A. An operator stack
- B. An operand stack
- C. An operand stack and an operator stack
- D. A parse tree

[gate1997](#) [normal](#) [infix-postfix](#) [stack](#) [data-structure](#)

Answer

3.9.2 Infix Postfix: GATE1998-19b [top](#)

<https://gateoverflow.in/15708>

Compute the post fix equivalent of the following expression $3^* \log(x + 1) - \frac{a}{2}$

[gate1998](#) [stack](#) [infix-postfix](#)

Answer

Answers: Infix Postfix

3.9.1 Infix Postfix: GATE1997-1.7 [top](#)

<https://gateoverflow.in/2223>



Selected Answer

A. An operator stack // Infix to (Postfix or Prefix)

B. An operand stack //Postfix or Prefix Evaluation

C. An operand stack and an operator stack //we never use two stacks

But for Prefix to (Infix or postfix) OR Postfix to (Infix or prefix) We can use a stack where both operator and operand can present simultaneously

D. A parse tree // Not relevant to this question

Hence, **Option A** is Answer.

<http://condor.depaul.edu/ichu/csc415/notes/notes9/Infix.htm> is a good read.

18 votes

-- Rajesh Pradhan (22.6k points)

3.9.2 Infix Postfix: GATE1998-19b [top](#)

<https://gateoverflow.in/15708>



Selected Answer

According to <http://faculty.washington.edu/jstraub/dsa/aexp/>

The priority of the operators follows the usual conventions:

- The highest priority is assigned to unary operators (note that, in this context, a function such as \sin is considered a unary operator). All unary operators have the same priority.
- Exponentiation has the second highest priority.
- The third highest priority is assigned to the multiplication and division operators.
- The lowest priority is given to the addition and subtraction operators.

Example:->>

Infix expression: $3 * \log(10)$

Postfix expression:

$= 3 * (10 \log) \quad //(\text{Priority of unary operator } \log \text{ forces } \log(10) \text{ to evaluate first.})$

$= 3 10 \log *$

Now For our case

$3 * \log(x + 1) - a/2$

first content inside parenthesis will evaluate

so $x + 1$ will become $x1+$

now **among (*, /, log, +, -)operators ..log has highest priority due to unary opeartor**

So, $\log(x1+)$ will become $x1 + \log$

now **suppose z = x1 + log**

so convert this $3 * z - a/2$ to postfix

$3z * a2/-$

now substitue $z = x1 + \log$

$3x1 + \log * a2/-$ is Ans.

37 votes

-- Rajesh Pradhan (22.6k points)

3.10

Linked Lists(20) top

3.10.1 Linked Lists: CMI2012-B-07 top

<https://gateoverflow.in/46590>

We use the notation $[x_1, x_2, \dots, x_n]$ to denote a list of integers. $[]$ denotes the empty list, and $[n]$ is the list consisting of one integer n . For a nonempty list l , $head(l)$ returns the first element of l , and $tail(l)$ returns the list obtained by removing the first element from l . The function $length(l)$ returns the length of a list. For example,

- $head([11, -1, 5]) = 11, tail([11, -1, 5]) = [-1, 5] \ .$
- $head([7]) = 7, tail([7]) = [] \ .$
- $length([]) = 0, length([7]) = 1, length([11, -1, 5]) = 3 \ .$

We use or, and and not to denote the usual operations on boolean values *true* and *false*.

Consider the following functions, each of which takes a list of integers as input and returns a boolean value.

```
f(l)
  if (length(l) < 2) then return(true)
  else return(g(l) or h(l))

g(l)
  if (length(l) < 2) then return(true)
  else
    if (head(l) < head(tail(l))) then return h(tail(l))
    else return(false)

h(l)
  if (length(l) < 2) then return(true)
```

```

    else
        if (head(l) > head(tail(l))) then return g(tail(l))
        else return (false)

```

When does $f(l)$ return the value true for an input l ? Explain your answer.

[cmi2012](#) [descriptive](#) [data-structure](#) [linked-lists](#)

Answer

3.10.2 Linked Lists: GATE1987-1-xv [top](#)

<https://gateoverflow.in/80298>

In a circular linked list organisation, insertion of a record involves modification of

- A. One pointer.
- B. Two pointers.
- C. Multiple pointers.
- D. No pointer.

[gate1987](#) [data-structure](#) [linked-lists](#)

Answer

3.10.3 Linked Lists: GATE1993-13 [top](#)

<https://gateoverflow.in/2310>

Consider a singly linked list having n nodes. The data items d_1, d_2, \dots, d_n are stored in these n nodes. Let X be a pointer to the j^{th} node ($1 \leq j \leq n$) in which d_j is stored. A new data item d stored in node with address Y is to be inserted. Give an algorithm to insert d into the list to obtain a list having items $d_1, d_2, \dots, d_j, d, \dots, d_n$ in order without using the header.

[gate1993](#) [data-structure](#) [linked-lists](#) [normal](#)

Answer

3.10.4 Linked Lists: GATE1994-1.17, UGCNET-Sep2013-II-32

<https://gateoverflow.in/2460>

Linked lists are not suitable data structures for which one of the following problems?

- A. Insertion sort
- B. Binary search
- C. Radix sort
- D. Polynomial manipulation

[gate1994](#) [data-structure](#) [linked-lists](#) [normal](#) [ugcnetsep2013ii](#)

Answer

3.10.5 Linked Lists: GATE1995-2.22 [top](#)

<https://gateoverflow.in/2634>

Which of the following statements is true?

- I. As the number of entries in a hash table increases, the number of collisions increases.
- II. Recursive programs are efficient
- III. The worst case complexity for Quicksort is $O(n^2)$

IV. Binary search using a linear linked list is efficient

- A. I and II
- B. II and III
- C. I and IV
- D. I and III

[gate1995](#) [data-structure](#) [linked-lists](#) [hashing](#)

Answer

3.10.6 Linked Lists: GATE1997-1.4 [top](#)

<https://gateoverflow.in/2220>

The concatenation of two lists is to be performed on $O(1)$ time. Which of the following implementations of a list should be used?

- A. Singly linked list
- B. Doubly linked list
- C. Circular doubly linked list
- D. Array implementation of list

[gate1997](#) [data-structure](#) [linked-lists](#) [easy](#)

Answer

3.10.7 Linked Lists: GATE1997-18 [top](#)

<https://gateoverflow.in/2278>

Consider the following piece of 'C' code fragment that removes duplicates from an ordered list of integers.

```
Node *remove-duplicates (Node* head, int *j) { Node *t1, *t2; *j=0; t1 = head; if (t1!=NULL) t2 = t1->next; else return head; *j = 1; if(t2 == NULL) return head; while (t2 != NULL) { if (t1.val != t2.val) -----> (S1) { (*j)++; t1-> next = t2; t1 = t2; -----> (S2) } t2 = t2->next; } t1-> next = NULL; return head; }
```

Assume the list contains n elements ($n \geq 2$) in the following questions.

How many times is the comparison in statement $S1$ made?

What is the minimum and the maximum number of times statements marked $S2$ get executed?

What is the significance of the value in the integer pointed to by j when the function completes?

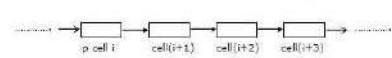
[gate1997](#) [data-structure](#) [linked-lists](#) [normal](#)

Answer

3.10.8 Linked Lists: GATE1998-19a [top](#)

<https://gateoverflow.in/1733>

- a. Let p be a pointer as shown in the figure in a single linked list.



What do the following assignment statements achieve?

```
q: = p -> next
p -> next:= q -> next
```

```

q -> next := (q -> next) -> next
(p -> next) -> next := q

```

[gate1998](#) [data-structure](#) [linked-lists](#) [normal](#)

Answer

3.10.9 Linked Lists: GATE1999-11b [top](#)

<https://gateoverflow.in/93575>

Write a constant time algorithm to insert a node with data D just before the node with address p of a singly linked list.

[gate1999](#) [data-structure](#) [linked-lists](#)

Answer

3.10.10 Linked Lists: GATE2002-1.5 [top](#)

<https://gateoverflow.in/809>

In the worst case, the number of comparisons needed to search a single linked list of length n for a given element is

- A. $\log n$
- B. $\frac{n}{2}$
- C. $\log_2 n - 1$
- D. n

[gate2002](#) [easy](#) [data-structure](#) [linked-lists](#)

Answer

3.10.11 Linked Lists: GATE2003-90 [top](#)

<https://gateoverflow.in/973>

Consider the function f defined below.

```

struct item {
    int data;
    struct item * next;
};

int f(struct item *p) {
    return ((p == NULL) || (p->next == NULL) ||
            ((p->data <= p->next->data) &&
             f(p->next)));
}

```

For a given linked list p , the function f returns 1 if and only if

- A. the list is empty or has exactly one element
- B. the elements in the list are sorted in non-decreasing order of data value
- C. the elements in the list are sorted in non-increasing order of data value
- D. not all elements in the list have the same data value

[gate2003](#) [data-structure](#) [linked-lists](#) [normal](#)

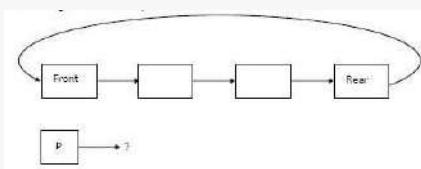
Answer

3.10.12 Linked Lists: GATE2004-36 [top](#)

<https://gateoverflow.in/1033>

A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue.

To which node should p point such that both the operations `enQueue` and `deQueue` can be performed in constant time?



- A. rear node
- B. front node
- C. not possible with a single pointer
- D. node next to front

[gate2004](#) [data-structure](#) [linked-lists](#) [normal](#)

[Answer](#)

3.10.13 Linked Lists: GATE2004-40 [top](#)

<https://gateoverflow.in/1037>

Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among `union`, `intersection`, `membership`, `cardinality` will be the slowest?

- A. `union` only
- B. `intersection`, `membership`
- C. `membership`, `cardinality`
- D. `union`, `intersection`

[gate2004](#) [data-structure](#) [linked-lists](#) [normal](#)

[Answer](#)

3.10.14 Linked Lists: GATE2004-IT-13 [top](#)

<https://gateoverflow.in/3654>

Let P be a singly linked list. Let Q be the pointer to an intermediate node x in the list. What is the worst-case time complexity of the best-known algorithm to delete the node x from the list?

- A. $O(n)$
- B. $O(\log^2 n)$
- C. $O(\log n)$
- D. $O(1)$

[gate2004-it](#) [data-structure](#) [linked-lists](#) [normal](#) [ambiguous](#)

[Answer](#)

3.10.15 Linked Lists: GATE2005-IT-54 [top](#)

<https://gateoverflow.in/3815>

The following C function takes a singly-linked list of integers as a parameter and rearranges the elements of the list. The list is represented as pointer to a structure. The function is called with the list containing the integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes execution?

```

struct node {int value; struct node *next;};
void rearrange (struct node *list) {
    struct node *p, *q;
  
```

```

int temp;
if (!list || !list -> next) return;
p = list; q = list -> next;
while (q) {
    temp = p -> value;
    p -> value = q -> value;
    q -> value = temp;
    p = q -> next;
    q = p ? p -> next : 0;
}
}

```

- A. 1,2,3,4,5,6,7
- B. 2,1,4,3,6,5,7
- C. 1,3,2,5,4,7,6
- D. 2,3,4,5,6,7,1

gate2005-it data-structure linked-lists normal

Answer

3.10.16 Linked Lists: GATE2008-62 [top](#)

<https://gateoverflow.in/485>

The following C function takes a single-linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers 1,2,3,4,5,6,7 in the given order. What will be the contents of the list after function completes execution?

```

struct node {
    int value;
    struct node *next;
};

void rearrange(struct node *list) {
    struct node *p, *q;
    int temp;
    if (!list || !list -> next) return;
    p = list; q = list -> next;
    while(q) {
        temp = p -> value; p->value = q -> value;
        q->value = temp; p = q ->next;
        q = p ? p ->next : 0;
    }
}

```

- A. 1,2,3,4,5,6,7
- B. 2,1,4,3,6,5,7
- C. 1,3,2,5,4,7,6
- D. 2,3,4,5,6,7,1

gate2008 data-structure linked-lists normal

Answer

3.10.17 Linked Lists: GATE2010-36 [top](#)

<https://gateoverflow.in/2337>

The following C function takes a singly-linked list as input argument. It modifies the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank.

```

typedef struct node
{

```

```

    int value;
    struct node *next;
} node;
Node *move_to_front(Node *head)
{
    Node *p, *q;
    if ((head == NULL) || (head -> next == NULL))
        return head;
    q = NULL;
    p = head;
    while (p->next != NULL)
    {
        q=p;
        p=p->next;
    }
    _____
    return head;
}

```

Choose the correct alternative to replace the blank line.

- A. $q = NULL; p \rightarrow next = head; head = p;$
- B. $q \rightarrow next = NULL; head = p; p \rightarrow next = head;$
- C. $head = p; p \rightarrow next = q; q \rightarrow next = NULL;$
- D. $q \rightarrow next = NULL; p \rightarrow next = head; head = p;$

gate2010 data-structure linked-lists normal

Answer

3.10.18 Linked Lists: GATE2016-2-15 [top](#)

<https://gateoverflow.in/39557>

N items are stored in a sorted doubly linked list. For a *delete* operation, a pointer is provided to the record to be deleted. For a *decrease-key* operation, a pointer is provided to the record on which the operation is to be performed.

An algorithm performs the following operations on the list in this order: $\Theta(N)$ *delete*, $O(\log N)$ *insert*, $O(\log N)$ *find*, and $\Theta(N)$ *decrease-key*. What is the time complexity of all these operations put together?

- A. $O(\log^2 N)$
- B. $O(N)$
- C. $O(N^2)$
- D. $\Theta(N^2 \log N)$

gate2016-2 data-structure linked-lists time-complexity normal

Answer

3.10.19 Linked Lists: GATE2017-1-08 [top](#)

<https://gateoverflow.in/118711>

Consider the C code fragment given below.

```

typedef struct node {
    int data;
    node* next;
} node;

```

```
void join(node* m, node* n) {
    node* p = n;
    while(p->next != NULL) {
        p = p->next;
    }
    p->next = m;
}
```

Assuming that m and n point to valid NULL-terminated linked lists, invocation of `join` will

- A. append list m to the end of list n for all inputs.
- B. either cause a null pointer dereference or append list m to the end of list n .
- C. cause a null pointer dereference for all inputs.
- D. append list n to the end of list m for all inputs.

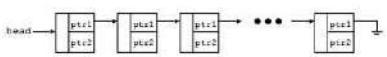
gate2017-1 data-structure linked-lists normal

[Answer](#)

3.10.20 Linked Lists: ISI2015-CS-3a [top](#)

<https://gateoverflow.in/47273>

Consider a linked list containing n nodes, where each node contains two pointers $ptr1$ and $ptr2$. For each node, $ptr1$ points to the next node of the list. Describe how pointer $ptr2$ should be set up for each node so that you will be able to locate the i -th node from the start node in the list traversing no more than $\lceil \log i \rceil + [i/2]$ nodes.



descriptive isi2015 data-structure linked-lists

[Answer](#)

Answers: Linked Lists

3.10.1 Linked Lists: CMI2012-B-07 [top](#)

<https://gateoverflow.in/46590>



Selected Answer

For any list L , if there exists 3 consecutive numbers which are sorted or reverse sorted, then the function will return false .

Here, we assume that any 2 consecutive elements will always be different. In the question they have not specified what happens when we compare 2 consecutive similar numbers.

Let me consider a list $L : [X_1, X_2, X_3, \dots, X_n], n > 2$

The 2 possible input sequences for which the function returns true will be,

$X_1 > X_2 < X_3 > X_4 < X_5 \dots$
 $X_1 < X_2 > X_3 < X_4 > X_5 \dots$

4 votes

-- balainstein (1.3k points)

3.10.2 Linked Lists: GATE1987-1-xv [top](#)

<https://gateoverflow.in/80298>



Selected Answer

Suppose we have to insert node p after node q then

$$p \rightarrow next = q \rightarrow next$$

$$q \rightarrow next = p$$

So, two pointers,

Answer is (B).

13 votes

-- Sanket_ (4.3k points)

3.10.3 Linked Lists: GATE1993-13 top

<https://gateoverflow.in/2310>



Selected Answer

Following 2 lines are enough to realise above constraint :->>

1. $Y \rightarrow next = X \rightarrow next$

2. $X \rightarrow next = Y$

14 votes

-- Rajesh Pradhan (22.6k points)

3.10.4 Linked Lists: GATE1994-1.17, UGCNET-Sep2013-II-32

<https://gateoverflow.in/2460>



Selected Answer

Linked lists are suitable for:

Insertion sort: No need to swap here just find appropriate place and join the link

Polynomial manipulation: Linked List is a natural solution for polynomial manipulation

Radix sort: Here we are putting digits according to same position(unit,tens) into buckets; which can be effectively handled by linked lists.

Not Suitable for:

Binary search: Because finding mid element itself takes $O(n)$ time.

So, Option B is answer.

20 votes

-- Rajesh Pradhan (22.6k points)

3.10.5 Linked Lists: GATE1995-2.22 top

<https://gateoverflow.in/2634>



Selected Answer

Its (D) according to me.

Binary search using linked list is not efficient as it will not give $O(\log n)$, because we will not be able to find the mid in constant time. Finding mid in linked list takes $O(n)$ time.

Recursive programs are not efficient because they take a lot of space, Recursive methods will often throw *StackOverflowException* while processing big sets. moreover it has its own advantages too.

22 votes

-- Gate Keeda (19.6k points)

3.10.6 Linked Lists: GATE1997-1.4 top

<https://gateoverflow.in/2220>



Selected Answer

Option C (Circular Doubly linked List)

Analyze below Code which is $O(1)$

Suppose List1's first element is pointed by pointer $p1$

And List2's first element is pointed by $p2$

And tmp is a temporary pointer of node type.

```
p1->prev->next = p2 ;
tmp= p2-> prev ;
p2-> prev= p1-> prev ;
p1-> prev = tmp-> next ;
tmp -> next = p1
```

Option A&B of linked list are not possible in $O(1)$. Bcz they cant find out rear element without doing linear traversal.

For **Option D**. Array implementation it requires $O(n1 + n2)$ Copy operation where $n1$ represents size of List1.

13 votes

-- Rajesh Pradhan (22.6k points)

3.10.7 Linked Lists: GATE1997-18 top

<https://gateoverflow.in/2278>



Selected Answer

- a. As we are comparing here pair wise so for n elements we require compulsory $n - 1$ comparison
- b. S^2 is executed only for distinct elements so **max** $n - 1$ times and **min** 0 when all r duplicates or list contain no or 1 element.
- c. j holds the count on number of distinct elements in the ordered list.

11 votes

-- Rajesh Pradhan (22.6k points)

3.10.8 Linked Lists: GATE1998-19a top

<https://gateoverflow.in/1733>



Selected Answer
Swaps the two nodes next to p in the linked list.

1 21 votes

-- Arjun Suresh (350k points)

3.10.9 Linked Lists: GATE1999-11b top

<https://gateoverflow.in/93575>



Let A, B, C, D, E, F be the data.
 $A \rightarrow B \rightarrow C \rightarrow E \rightarrow F$

Let pointer to E be p .

A node with data D has to be inserted before E .

I'll do one thing - add D just after node E and it'll take constant time.

$p.next \rightarrow Address_of_D$

$Address_of_D.next \rightarrow Address_of_F$

$A \rightarrow B \rightarrow C \rightarrow E \rightarrow D \rightarrow F$

Take a temporary variable and swap the values E and D ..

$temp = p.data$

$p.data = p.next.data$

$p.next.data = temp$

Now linked list wil look like

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$

still one more work left - now pointer p pointing to D so increment pointer p to point data E .

$p = p \rightarrow next$

1 19 votes

-- Digvijay (54.9k points)

3.10.10 Linked Lists: GATE2002-1.5 top

<https://gateoverflow.in/809>



A & C are not correct as we can not do binary search in Linked list.

B seems like average case, be we are asked for worst case.

Worst case is we do not find the element in list. We might end up searching entire list & comparing with each element. So, answer -> (**D**). n

1 19 votes

-- Akash Kanase (42.5k points)

3.10.11 Linked Lists: GATE2003-90 top

<https://gateoverflow.in/973>



It returns 1 if and only if the linked list is sorted in non-decreasing order- (**B**) option.

It returns 1 if the list is empty or has just 1 element also, but if and only if makes (A) option false.

25 votes

-- Bhagirathi Nayak (14.1k points)

3.10.12 Linked Lists: GATE2004-36 top

<https://gateoverflow.in/1033>



Selected Answer

The pointer points to the Rear node.

EnQueue: Insert newNode after Rear, and make Rear point to the newly inserted node:

```
//struct node *newNode;
newNode->next = rear->next;
rear->next = newNode;
rear=newNode;
```

DeQueue: Delete the Front node, and make the second node the front node.

```
//rear->next points to the front node.
//front->next points to the second node.
struct node* front;
front = rear->next;
rear->next = front->next;
free(front);
```

55 votes

-- Pragy Agarwal (20.6k points)

3.10.13 Linked Lists: GATE2004-40 top

<https://gateoverflow.in/1037>



Selected Answer

Answer is (D).

membership is linear search - $O(n_1 + n_2)$

Cardinality is linear - $O(n_1 + n_2)$

For union we need to ensure no duplicate elements should be present - $O(n_1 \times n_2)$ for each element we need to check if that element exists in other set

For intersection also for every element in set1 we need to scan set2 - $O(n_1 \times n_2)$

40 votes

-- Ankit Rokde (9k points)

3.10.14 Linked Lists: GATE2004-IT-13 top

<https://gateoverflow.in/3654>

Since Q is pointing to node X , it can be done in $O(1)$ time..

Algo:

$Q \rightarrow data = Q \rightarrow next \rightarrow data ; //$ Copy the value of next node into Q .

```
del = Q -> next; // take another pointer variable pointing to next node of Q.
Q -> next = Q -> next -> next;
free(del);
```

Upvote 43 votes

-- gate_asp (749 points)

3.10.15 Linked Lists: GATE2005-IT-54 top

<https://gateoverflow.in/3815>



I think it's (B) 2,1,4,3,6,5,7 :

As,

p and q are swapping each other. where q is $p \rightarrow \text{next}$ all the time.

Upvote 17 votes

-- sumit kumar singh dixit (2.1k points)

3.10.16 Linked Lists: GATE2008-62 top

<https://gateoverflow.in/485>



The loop is interchanging the adjacent elements of the list. But after each interchange, next interchange starts from the unchanged elements only (due to $p = q \rightarrow \text{next};$).

1st iteration 1,2,3,4,5,6,7
 $\Rightarrow 2,1,3,4,5,6,7$

2nd iteration 2,1,4,3,5,6,7

3rd iteration 2,1,4,3,6,5,7

p pointing to null q pointing to 7, as p is false hence $q = p?$ $p \rightarrow \text{next}:0;$ will return $q = 0$ ending the loop

Answer is **option B.**

Upvote 17 votes

-- Manali (2.6k points)

3.10.17 Linked Lists: GATE2010-36 top

<https://gateoverflow.in/2337>



As per given code p points to last node which should be head in modified.

q is the previous node of tail which should be tail for modified

Answer is (D).

Upvote 26 votes

-- Sankaranarayanan P.N (11.5k points)

3.10.18 Linked Lists: GATE2016-2-15 top

<https://gateoverflow.in/39557>



Selected Answer

Answer is (C).

- Delete $O(1)$
- Insert $O(N)$
- Find $O(N)$

Decrease Key $\Rightarrow O(N)$ //Because we need to search position in Linked list. (It is similar to a Delete followed by an Insert with the decreased value)

- $O(n)$ delete $\Rightarrow O(N \times 1) = O(N)$
- $O(\log N)$ find $\Rightarrow O(\log N \times N) = O(N \log N)$
- $O(\log N)$ insert $\Rightarrow O(N \log N)$
- $O(N)$ decrease key $\Rightarrow O(N \times N) = O(N^2)$

Even though it says at start we got N elements, then we are deleting $Q(N)$ elements, here $Q(N)$ can be anything like $N/2, N/4, N/3$ and list need not be empty, then above explanation holds !

In case it actually deleted all elements at start analysis will be something like below \Rightarrow

All N elements are deleted, Time complexity $O(1)$ each delete, total delete $O(N)$

Now $\log N$ insert, it'll take $1 + 2 + \log N$ time, then $(\log N * \log N - 1)/2 \Rightarrow O((\log N)^2)$

Now $\log N$ finds \Rightarrow it'll take $\log N$ time per find (because find take $O(N)$ but here $N = \log N$)
 $\Rightarrow O((\log N)^2)$

Now N decrease key, Size of list is $\log N$, each decrease key can take $O(\text{size of list})$

So, size of list * no. of decrease key $\Rightarrow N \times \log N = O(N \log N)$

There is no option like $O(N \log N)$

So, correct answer is $O(N^2)$.

47 votes

-- Akash Kanase (42.5k points)

3.10.19 Linked Lists: GATE2017-1-08 top

<https://gateoverflow.in/118711>



Selected Answer

Here is the implemented code in c (-std=c99).

```
#include <stdio.h>
#include <stdlib.h>

#define M 5
#define N 4

int M_data[M] = {1,2,3,4,5};
int N_data[N] = {6,7,8,9};

typedef struct node {
```

```

int data;
struct node * next;
} node;

void join(node *m, node *n) {
    node * p = n;
    while(p->next != NULL) p = p->next;
    p->next = m;
}

node * bulk_insert(int list_data[], int size) {
    node * head = NULL;
    for(int i=0;i<size;i++) {
        node * newnode = malloc(sizeof(node));
        newnode->data = list_data[i];
        newnode->next = NULL;

        if(head == NULL) {
            head = newnode;
        } else {
            node * temp = head;
            while(temp->next != NULL) temp = temp->next;
            temp->next = newnode;
        }
    }
    return head;
}
void display(node *);
void list_dealloc(node */*deallocation prototype*/);

int main() {
    node * m = NULL;
    node * n = NULL;
    // insert m_list data
    m = bulk_insert(M_data, M);
    n = bulk_insert(N_data, N); // commenting this causes runtime error
                                // is list n is empty
    printf("\n before join operation :\n");
    display(m);
    display(n);

    join(m, n);

    printf("\n after join operation :\n");
    display(m);
    display(n);

    //list_dealloc(m); no need now
    list_dealloc(n); // OK
    return 0;
}

void display(node *head) {
    while(head != NULL) {
        printf("%d->", head->data);
        head = head->next;
    }
    printf("null\n");
}
void list_dealloc(node * head) {
    while(head != NULL) {
        node * temp = head;
        head = head->next;
        free(temp);
    }
}

```

```
    free (temp);  
}
```

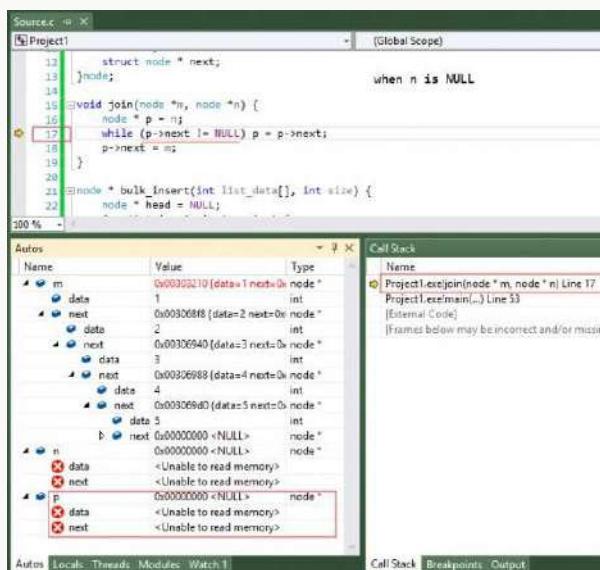
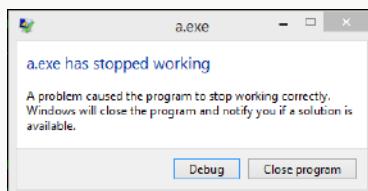
With both **n** and **m** and **n** being **non-empty** linked list, then,

O/P:

```
before join operation :  
1->2->3->4->5->null  
6->7->8->9->null  
  
after join operation :  
1->2->3->4->5->null  
6->7->8->9->1->2->3->4->5->null
```

With n being **empty** linked list, then,

O/P:



 32 votes

-- Debashish Deka (56.7k points)

3.10.20 Linked Lists: ISI2015-CS-3a top

<https://gateoverflow.in/47273>



Link 1st node's ptr2 pointer to 2nd node. then 2nd node to 4th node. 4th node to 8th node, 8th node to 16th so on.

For example I need 100th node then do like this- go to $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128$. Then it takes $\log i$ time.

Then to go to 100 start at 64 and linear traversal has to be done for 36 times.

This linear traversal should take $i/2$ node traversals in the worst case.

Overall it would take traversing no more than $\log i + i/2$ nodes.

15 votes

-- Gaurab Ghosh (2.2k points)

3.11 Priority Queue(1) top

3.11.1 Priority Queue: GATE1997-4.7 top

<https://gateoverflow.in/2248>

A priority queue Q is used to implement a stack that stores characters. PUSH (C) is implemented as INSERT (Q, C, K) where K is an appropriate integer key chosen by the implementation. POP is implemented as DELETEMIN(Q). For a sequence of operations, the keys chosen are in

- A. non-increasing order
- B. non-decreasing order
- C. strictly increasing order
- D. strictly decreasing order

gate1997 data-structure stack normal priority-queue

Answer

Answers: Priority Queue

3.11.1 Priority Queue: GATE1997-4.7 top

<https://gateoverflow.in/2248>



Selected Answer

Implementing stack using priority queue require first element inserted in stack will be deleted at last, and to implement it using deletemin() operation of queue will require first element inserted in queue must have highest priority.

So the keys must be in strictly decreasing order.

19 votes

-- Suraj Kaushal (351 points)

3.12 Queues(12) top

3.12.1 Queues: GATE1992-09 top

<https://gateoverflow.in/588>

Suggest a data structure for representing a subset S of integers from 1 to n . Following operations on the set S are to be performed in constant time (independent of cardinality of S).

- i. MEMBER Check whether X is in (X) : the set S or not

- I f S is not empty,
ii. FIND-ONE return one element of
 (S) : the set S (any
arbitrary element will
do)
- iii. ADD (X): Add integer X to set
 S
- iv. DELETE Delete integer X from
 (X) : S .

Give pictorial examples of your data structure. Give routines for these operations in an English like language. You may assume that the data structure has been suitable initialized. Clearly state your assumptions regarding initialization.

[gate1992](#) [data-structure](#) [normal](#) [descriptive](#) [queues](#)

Answer

3.12.2 Queues: GATE1994-26 [top](#)

<https://gateoverflow.in/2522>

A queue Q containing n items and an empty stack S are given. It is required to transfer all the items from the queue to the stack, so that the item at the front of queue is on the TOP of the stack, and the order of all other items are preserved. Show how this can be done in $O(n)$ time using only a constant amount of additional storage. Note that the only operations which can be performed on the queue and stack are Delete, Insert, Push and Pop. Do not assume any implementation of the queue or stack.

[gate1994](#) [data-structure](#) [queues](#) [stack](#) [normal](#)

Answer

3.12.3 Queues: GATE1996-1.12 [top](#)

<https://gateoverflow.in/2716>

Consider the following statements:

- i. First-in-first out types of computations are efficiently supported by STACKS.
 - ii. Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.
 - iii. Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.
 - iv. Last-in-first-out type of computations are efficiently supported by QUEUES.
- A. (ii) and (iii) are true
B. (i) and (ii) are true
C. (iii) and (iv) are true
D. (ii) and (iv) are true

[gate1996](#) [data-structure](#) [easy](#) [queues](#) [stack](#) [linked-lists](#)

Answer

3.12.4 Queues: GATE2001-2.16 [top](#)

<https://gateoverflow.in/734>

What is the minimum number of stacks of size n required to implement a queue of size n ?

- A. One
- B. Two
- C. Three
- D. Four

gate2001 | data-structure | easy | stack | queues

[Answer](#)

3.12.5 Queues: GATE2006-49 [top](#)

<https://gateoverflow.in/1826>

An implementation of a queue Q , using two stacks $S1$ and $S2$, is given below:

```
void insert (Q, x) {
    push (S1, x);
}
void delete (Q) {
    if (stack-empty(S2)) then
        if (stack-empty(S1)) then {
            print("Q is empty");
            return;
        }
        else while (! (stack-empty(S1))) {
            x=pop (S1);
            push (S2,x);
        }
    x=pop (S2);
}
```

Let n insert and m ($\leq n$) delete operations be performed in an arbitrary order on an empty queue Q . Let x and y be the number of push and pop operations performed respectively in the process. Which one of the following is true for all m and n ?

- A. $n + m \leq x < 2n$ and $2m \leq y \leq n + m$
- B. $n + m \leq x < 2n$ and $2m \leq y \leq 2n$
- C. $2m \leq x < 2n$ and $2m \leq y \leq n + m$
- D. $2m \leq x < 2n$ and $2m \leq y \leq 2n$

gate2006 | data-structure | queues | stack | normal

[Answer](#)

3.12.6 Queues: GATE2007-IT-30 [top](#)

<https://gateoverflow.in/3463>

Suppose you are given an implementation of a queue of integers. The operations that can be performed on the queue are:

- i. $isEmpty(Q)$ — returns true if the queue is empty, false otherwise.
- ii. $delete(Q)$ — deletes the element at the front of the queue and returns its value.
- iii. $insert(Q, i)$ — inserts the integer i at the rear of the queue.

Consider the following function:

```
void f (queue Q) {
int i ;
if (!isEmpty(Q)) {
    i = delete(Q);
    f(Q);
    insert(Q, i);
}
```

}

What operation is performed by the above function f ?

- A. Leaves the queue Q unchanged
- B. Reverses the order of the elements in the queue Q
- C. Deletes the element at the front of the queue Q and inserts it at the rear keeping the other elements in the same order
- D. Empties the queue Q

gate2007-it data-structure queues normal

Answer

3.12.7 Queues: GATE2012-35 top

<https://gateoverflow.in/1756>

Suppose a circular queue of capacity $(n - 1)$ elements is implemented with an array of n elements. Assume that the insertion and deletion operations are carried out using REAR and FRONT as array index variables, respectively. Initially, $REAR = FRONT = 0$. The conditions to detect queue full and queue empty are

- A. full: $(REAR + 1) \bmod n == FRONT$
empty: $REAR == FRONT$
- B. full: $(REAR + 1) \bmod n == FRONT$
empty: $(FRONT + 1) \bmod n == REAR$
- C. full: $REAR == FRONT$
empty: $(REAR + 1) \bmod n == FRONT$
- D. full: $(FRONT + 1) \bmod n == REAR$
empty: $REAR == FRONT$

gate2012 data-structure queues normal

Answer

3.12.8 Queues: GATE2013-44 top

<https://gateoverflow.in/61>

Consider the following operation along with Enqueue and Dequeue operations on queues, where k is a global parameter.

```
ultiDequeue(Q) {
    m = k
    while (Q is not empty) and (m > 0) {
        Dequeue(Q)
        m = m - 1
    }
}
```

What is the worst case time complexity of a sequence of n queue operations on an initially empty queue?

- A. $\Theta(n)$
- B. $\Theta(n + k)$
- C. $\Theta(nk)$
- D. $\Theta(n^2)$

[gate2013](#) [data-structure](#) [algorithms](#) [normal](#) [queues](#)
Answer

3.12.9 Queues: GATE2016-1-10 [top](#)

<https://gateoverflow.in/39667>

A queue is implemented using an array such that ENQUEUE and DEQUEUE operations are performed efficiently. Which one of the following statements is **CORRECT** (n refers to the number of items in the queue) ?

- Both operations can be performed in $O(1)$ time.
- At most one operation can be performed in $O(1)$ time but the worst case time for the operation will be $\Omega(n)$.
- The worst case time complexity for both operations will be $\Omega(n)$.
- Worst case time complexity for both operations will be $\Omega(\log n)$

[gate2016-1](#) [data-structure](#) [queues](#) [normal](#)
Answer

3.12.10 Queues: GATE2016-1-41 [top](#)

<https://gateoverflow.in/39684>

Let Q denote a queue containing sixteen numbers and S be an empty stack. $Head(Q)$ returns the element at the head of the queue Q without removing it from Q . Similarly $Top(S)$ returns the element at the top of S without removing it from S . Consider the algorithm given below.

```

while Q is not Empty do
    if S is Empty OR Top(S) ≤ Head (Q) then
        x:= Dequeue (Q);
        Push (S, x);
    else
        x:= Pop (S);
        Enqueue (Q, x);
    end
end

```

The maximum possible number of iterations of the **while** loop in the algorithm is _____.

[gate2016-1](#) [data-structure](#) [queues](#) [difficult](#) [numerical-answers](#)
Answer

3.12.11 Queues: GATE2017-2-13 [top](#)

<https://gateoverflow.in/118253>

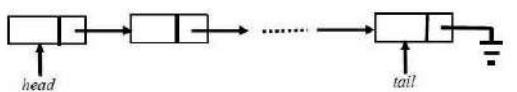
A circular queue has been implemented using a singly linked list where each node consists of a value and a single pointer pointing to the next node. We maintain exactly two external pointers FRONT and REAR pointing to the front node and the rear node of the queue, respectively. Which of the following statements is/are CORRECT for such a circular queue, so that insertion and deletion operations can be performed in $O(1)$ time?

- Next pointer of front node points to the rear node.
 - Next pointer of rear node points to the front node.
- (I) only.
 - (II) only.
 - Both (I) and (II).
 - Neither (I) nor (II).

[gate2017-2](#) [data-structure](#) [queues](#)

Answer**3.12.12 Queues: GATE2018-3** top<https://gateoverflow.in/204077>

A queue is implemented using a non-circular singly linked list. The queue has a head pointer and a tail pointer, as shown in the figure. Let n denote the number of nodes in the queue. Let 'enqueue' be implemented by inserting a new node at the head, and 'dequeue' be implemented by deletion of a node from the tail.



Which one of the following is the time complexity of the most time-efficient implementation of 'enqueue' and 'dequeue', respectively, for this data structure?

- A. $\theta(1), \theta(1)$
- B. $\theta(1), \theta(n)$
- C. $\theta(n), \theta(1)$
- D. $\theta(n), \theta(n)$

[gate2018](#) [algorithms](#) [data-structure](#) [queues](#) [normal](#) [linked-lists](#)
Answer**Answers: Queues****3.12.1 Queues: GATE1992-09** top<https://gateoverflow.in/588>

A queue with a hashtable.

Initialize hashtable with 0.

When inserting X into the queue update $\text{hashtable}[X] = 0$ to $\text{hashtable}[X] = 1$.

- i. If $\text{hashtable}[X] = 1$ then return true.
- ii. Return the element at the front or rear of the queue.
- iii. Add the element X to the queue at the rear end and update $\text{hashtable}[X] = 0$ to $\text{hashtable}[X] = 1$.
- iv. Delete the element X from the front end of the queue and update $\text{hashtable}[X] = 1$ to $\text{hashtable}[X] = 0$.

10 votes

-- Rajarshi Sarkar (34.1k points)

3.12.2 Queues: GATE1994-26 top<https://gateoverflow.in/2522>

Selected Answer

We can do this by first extracting items one by one from Q , and inserting them to S . After all items are done, S will contain the items in reverse order. Now, pop the elements from S and insert to Q . After this operation, items in Q will be in reverse order from the starting. Now, extract items from Q and push on to stack and we are done.

```

Do
Delete an item from Q
Push the item to S
While (! empty Q);
Do
Pop an item from S
Insert the item to Q
While (! empty S);
Do
Delete an item from Q
Push the item to S
While (! empty Q);

```

32 votes

-- Arjun Suresh (350k points)

3.12.3 Queues: GATE1996-1.12 top

<https://gateoverflow.in/2716>



(A). (i) and (iv) are false.

[http://en.wikipedia.org/wiki/List_\(abstract_data_type\)#Operations](http://en.wikipedia.org/wiki/List_(abstract_data_type)#Operations)

22 votes

-- Gate Keeda (19.6k points)

3.12.4 Queues: GATE2001-2.16 top

<https://gateoverflow.in/734>



A queue can be implemented using two stacks.

Let queue be represented as " q " and stacks used to implement q be "stack1" and "stack2".

q can be implemented in two ways:

Method 1 (By making EnQueue operation costly)

This method makes sure that newly entered element is always at the bottom of stack 1, so that deQueue operation just pops from stack1. To put the element at top of stack1, stack2 is used.

$enQueue(q, x)$

1. While stack1 is not empty, push everything from stack1 to stack2.
2. Push x to stack1 (assuming size of stacks is unlimited).
3. Push everything back to stack1.

$deQueue(q)$

1. If stack1 is empty then error
2. Pop an item from stack1 and return it

Method 2 (By making deQueue operation costly)

In this method, in en-queue operation, the new element is entered at the top of stack1. In de-queue operation, if stack2 is empty then all the elements are moved to stack2 and finally top of

stack2 is returned.
 $\text{enQueue}(q, x)$

1. Push x to stack1 (assuming size of stacks is unlimited).

$\text{deQueue}(q)$

1. If both stacks are empty then error.
2. If stack2 is empty
 While stack1 is not empty, push everything from stack1 to stack2.
3. Pop the element from stack2 and return it.

17 votes

-- Dipak Majhi (979 points)

3.12.5 Queues: GATE2006-49 top

<https://gateoverflow.in/1826>



Selected Answer

Answer is (A).

The order in which insert and delete operations are performed matters here.

The best case: Insert and delete operations are performed alternatively. In every delete operation, 2 pop and 1 push operations are performed. So, total $m + n$ push (n push for $\text{insert}()$ and m push for $\text{delete}()$) operations and $2m$ pop operations are performed.

The worst case: First n elements are inserted and then m elements are deleted. In first delete operation, $n + 1$ pop operations and n push operation are performed. Other than first, in all delete operations, 1 pop operation is performed. So, total $m + n$ pop operations and $2n$ push operations are performed (n push for $\text{insert}()$ and m push for $\text{delete}()$)

38 votes

-- Kalpana Bhargav (3.3k points)

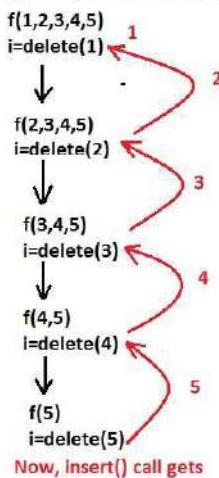
3.12.6 Queues: GATE2007-IT-30 top

<https://gateoverflow.in/3463>



Selected Answer

Suppose Q contains 1,2,3,4,5



$\text{insert}()$ will inserts the value in just reverse order.

23 votes

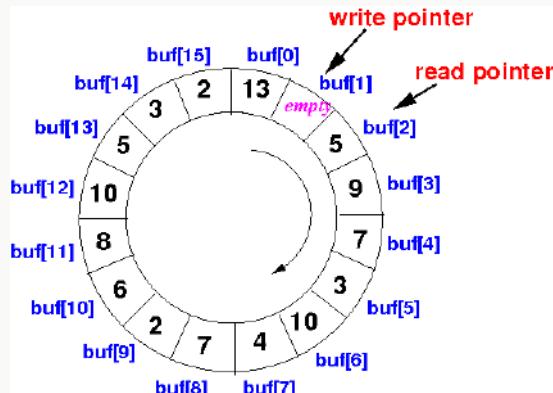
-- srestha (87k points)

3.12.7 Queues: GATE2012-35 top

<https://gateoverflow.in/1756>



Selected Answer



rear = Write

front = Read

full: $(REAR + 1) \bmod n == FRONT$

empty: $REAR == FRONT$

Only option (A) matches.

22 votes

-- Prashant Singh (59.8k points)

3.12.8 Queues: GATE2013-44 top

<https://gateoverflow.in/61>



Selected Answer

There are three possible operations on queue- Enqueue, Dequeue and MultiDequeue. MultiDequeue is calling Dequeue multiple times based on a global variable k . Since, the queue is initially empty, whatever be the order of these operations, there cannot be more no. of Dequeue operations than Enqueue operations. Hence, the total no. operations will be n only.

54 votes

-- Arjun Suresh (350k points)

3.12.9 Queues: GATE2016-1-10 top

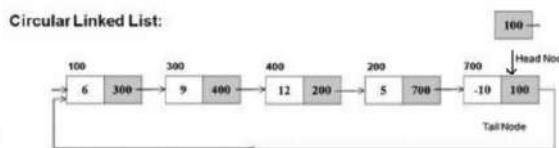
<https://gateoverflow.in/39667>



Selected Answer

Answer (A) - Circular Queue Implementation

- Both operations can be performed in $O(1)$ time in Circular Queue implementation where Enqueue and Dequeue operation are done at last node. Single pointer needed at last node.



Refer : <http://www.mathcs.emory.edu/~cheung/Courses/171/Syllabus/8-List/array-queue2.html>

34 votes

-- Abhilash Panicker (9.5k points)

3.12.10 Queues: GATE2016-1-41 top

<https://gateoverflow.in/39684>



While loop will run for the maximum number of times when the Queue elements are sorted in descending order.

Let's suppose that initially, Queue elements are $16, 15, 14, 13, \dots, 2, 1$

Now, 16 will be first pushed into the stack, So, now stack top is 16 and $Head(Q)$ is 15, So 16 will be popped out of the stack (since, "if S is Empty OR $Top(S) \leq Head(Q)$ " returns false, hence else part will be executed) and enqueue into the Queue.

So, after two iterations Queue elements will be $\rightarrow 15, 14, 13, \dots, 2, 1, 16$ and stack will be empty.

Similarly, each of the elements $15, 14, 13, \dots, 2$ will be pushed into the stack and immediately popped out of the stack(when popped out of the stack then also enqueue into the queue).So after 30 iterations stack will be empty and Queue contains will be like $\Rightarrow 1, 16, 15, 14, \dots, 2$.

Now 1 will be Dequeued and pushed into the stack. Once 1 is pushed into the stack, it will never be popped(or we can say never be enqueue into the Queue again) because in order to Pop 1, there should be an element into the Queue which is less than 1 and that element comes at the front of the queue, since there is no element currently present in the Queue which is less than 1, there is no way to pop 1.

So, after 31 iterations Queue is $\Rightarrow 16, 15, 14, \dots, 2$ and stack contains 1.

Now, the problem boils down to Queue with 15 elements.

Using the similar logic we can say after another 29 iterations (Total = $31 + 29$)Queue will be like $\Rightarrow 16, 15, 14, \dots, 3$ and stack contains 1,2 (stack top is 2) and then 2 will remain there in the stack forever.

Similar way if we go on then, after $31 + 29 + 27 + 25 + \dots + 1$ iterations Queue will be empty.

This is in A.P. series with $d = 2$. Sum = $(16 * (1 + 31)) / 2 = 16 * 32 / 2 = 256$

65 votes

-- Sourav Basu (1.5k points)

3.12.11 Queues: GATE2017-2-13 top

<https://gateoverflow.in/118253>

Answer is Next pointer to Rear node has Pointer to Front node.

Hence, only (II) is correct.

26 votes

-- Prashant Singh (59.8k points)

3.12.12 Queues: GATE2018-3 top

<https://gateoverflow.in/204077>

Selected Answer

New node to be inserted is P .

```
Enqueue () {
    P->Data=Data
    P->Next=Head
    Head=P
}
```

Time Complexity = $O(1)$ Because only pointer manipulation is involved which takes constant time.

Delete Tail

```
Dequeue () {
    temp=head
    While (temp->Next->Next!=NULL)
        temp=temp->next
    temp->next=NULL
    tail=temp
}
```

Time Complexity = Time for Traversing list, free the last node and keep track of Tail pointer
 $= O(n)$

17 votes

-- Digvijay (54.9k points)

3.13

Stack(16) top

3.13.1 Stack: GATE1991-03,vii top

<https://gateoverflow.in/522>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

The following sequence of operations is performed on a stack:

PUSH (10), PUSH (20), POP, PUSH (10), PUSH (20), POP, POP, POP, PUSH (20), POP

The sequence of values popped out is

- A. 20, 10, 20, 10, 20
- B. 20, 20, 10, 10, 20
- C. 10, 20, 20, 10, 20
- D. 20, 20, 10, 20, 10

gate1991 data-structure stack easy

Answer

3.13.2 Stack: GATE1994-1.14 top

<https://gateoverflow.in/2457>

Which of the following permutations can be obtained in the output (in the same order) using a stack assuming that the input is the sequence 1, 2, 3, 4, 5 in that order?

- A. 3, 4, 5, 1, 2
- B. 3, 4, 5, 2, 1
- C. 1, 5, 2, 3, 4
- D. 5, 4, 3, 1, 2

gate1994 data-structure stack normal

[Answer](#)

3.13.3 Stack: GATE1995-2.21 [top](#)

<https://gateoverflow.in/2633>

The postfix expression for the infix expression $A + B * (C + D)/F + D * E$ is:

- A. $AB + CD + * F / D + E *$
- B. $ABCD + * F / DE * ++$
- C. $A * B + CD / F * DE ++$
- D. $A + * BCD / F * DE ++$

gate1995 data-structure stack easy

[Answer](#)

3.13.4 Stack: GATE2000-13 [top](#)

<https://gateoverflow.in/684>

Suppose a stack implementation supports, in addition to PUSH and POP, an operation REVERSE, which reverses the order of the elements on the stack.

- A. To implement a queue using the above stack implementation, show how to implement ENQUEUE using a single operation and DEQUEUE using a sequence of 3 operations.
- B. The following post fix expression, containing single digit operands and arithmetic operators + and *, is evaluated using a stack.

$5\ 2\ *\ 3\ 4\ +\ 5\ 2\ *\ *+$

Show the contents of the stack

- i. After evaluating $5\ 2\ *\ 3\ 4\ +$
- ii. After evaluating $5\ 2\ *\ 3\ 4\ +\ 5\ 2$
- iii. At the end of evaluation

gate2000 data-structure stack normal descriptive

[Answer](#)

3.13.5 Stack: GATE2003-64 [top](#)

<https://gateoverflow.in/951>

Let \mathbf{S} be a stack of size $n \geq 1$. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and Pop operations take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation. For $m \geq 1$, define the stack-life of m as the time elapsed from the end of $Push(m)$ to the start of the pop operation that removes m from \mathbf{S} . The average stack-life of an element of this stack is

- A. $n(X + Y)$
- B. $3Y + 2X$
- C. $n(X + Y) - X$

D. $Y + 2X$

gate2003 data-structure stack normal

[Answer](#)

3.13.6 Stack: GATE2004-3 [top](#)

<https://gateoverflow.in/1000>

A single array $A[1..MAXSIZE]$ is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables $top1$ and $top2$ ($top1 < top2$) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for "stack full" is

- A. $(top1 = MAXSIZE/2)$ and $(top2 = MAXSIZE/2 + 1)$
- B. $top1 + top2 = MAXSIZE$
- C. $(top1 = MAXSIZE/2)$ or $(top2 = MAXSIZE)$
- D. $top1 = top2 - 1$

gate2004 data-structure stack easy

[Answer](#)

3.13.7 Stack: GATE2004-38, ISRO2009-27 [top](#)

<https://gateoverflow.in/1035>

Assume that the operators $+, -, \times$ are left associative and $^$ is right associative. The order of precedence (from highest to lowest) is $^, \times, +, -$. The postfix expression corresponding to the infix expression $a + b \times c - d^e^f$ is

- A. $abc \times +def^{^{\wedge}}-$
- B. $abc \times +de^f^{\wedge}-$
- C. $ab + c \times d - e^f^{\wedge}$
- D. $- + a \times bc^{^{\wedge}}def$

gate2004 stack isro2009

[Answer](#)

3.13.8 Stack: GATE2004-5 [top](#)

<https://gateoverflow.in/1002>

The best data structure to check whether an arithmetic expression has balanced parentheses is a

- A. queue
- B. stack
- C. tree
- D. list

gate2004 data-structure easy stack

[Answer](#)

3.13.9 Stack: GATE2004-IT-52 [top](#)

<https://gateoverflow.in/3695>

A program attempts to generate as many permutations as possible of the string, ' $abcd$ ' by pushing

the characters a, b, c, d in the same order onto a stack, but it may pop off the top character at any time. Which one of the following strings CANNOT be generated using this program?

- A. $abcd$
- B. $dcba$
- C. $cbad$
- D. $cabd$

gate2004-it data-structure normal stack

[Answer](#)

3.13.10 Stack: GATE2005-IT-13 [top](#)

<https://gateoverflow.in/3758>

A function f defined on stacks of integers satisfies the following properties. $f(\emptyset) = 0$ and $f(push(S, i)) = \max(f(S), 0) + i$ for all stacks S and integers i .

If a stack S contains the integers $2, -3, 2, -1, 2$ in order from bottom to top, what is $f(S)$?

- A. 6
- B. 4
- C. 3
- D. 2

gate2005-it data-structure stack normal

[Answer](#)

3.13.11 Stack: GATE2007-38, ISRO2016-27 [top](#)

<https://gateoverflow.in/1236>

The following postfix expression with single digit operands is evaluated using a stack:

8 2 3 ^ /

Note that $^$ is the exponentiation operator. The top two elements of the stack after the first $*$ is evaluated are

- A. 6,1
- B. 5,7
- C. 3,2
- D. 1,5

gate2007 data-structure stack normal isro2016

[Answer](#)

3.13.12 Stack: GATE2007-IT-32 [top](#)

<https://gateoverflow.in/3465>

Consider the following C program:

```
#include <stdio.h>
#define EOF -1
void push (int); /* push the argument on the stack */
int pop (void); /* pop the top of the stack */
void flagError ();
int main ()
{
    int c, m, n, r;
    while ((c = getchar ()) != EOF)
        if (isdigit (c))
            push (c);
```

```

        else if ( (c == '+') || (c == '*') )
        {
            m = pop ();
            n = pop ();
            r = (c == '+') ? n + m : n*m;
            push (r);
        }
        else if (c != ' ')
            flagError ();
    }
    printf("% c", pop ());
}

```

What is the output of the program for the following input?

5 2 * 3 3 2 + * +

- A. 15
- B. 25
- C. 30
- D. 150

[gate2007-it](#) [stack](#) [normal](#)

[Answer](#)

3.13.13 Stack: GATE2014-2-41 [top](#)

<https://gateoverflow.in/2007>

Suppose a stack implementation supports an instruction *REVERSE*, which reverses the order of elements on the stack, in addition to the *PUSH* and *POP* instructions. Which one of the following statements is TRUE (*with respect to this modified stack*)?

- A. A queue cannot be implemented using this stack.
- B. A queue can be implemented where *ENQUEUE* takes a single instruction and *DEQUEUE* takes a sequence of two instructions.
- C. A queue can be implemented where *ENQUEUE* takes a sequence of three instructions and *DEQUEUE* takes a single instruction.
- D. A queue can be implemented where both *ENQUEUE* and *DEQUEUE* take a single instruction each.

[gate2014-2](#) [data-structure](#) [stack](#) [easy](#)

[Answer](#)

3.13.14 Stack: GATE2015-2-38 [top](#)

<https://gateoverflow.in/8164>

Consider the C program below

```

#include <stdio.h>
int *A, stkTop;
int stkFunc (int opcode, int val)
{
    static int size=0, stkTop=0;
    switch (opcode) {
        case -1: size = val; break;
        case 0: if (stkTop < size) A[stkTop++]=val; break;
        default: if (stkTop) return A[--stkTop];
    }
    return -1;
}
int main()
{

```

```

int B[20]; A=B; stkTop = -1;
stkFunc (-1, 10);
stkFunc (0, 5);
stkFunc (0, 10);
printf ("%d\n", stkFunc(1, 0)+ stkFunc(1, 0));
}

```

The value printed by the above program is _____.

gate2015-2 data-structure stack easy numerical-answers

Answer

3.13.15 Stack: GATE2015-3-12 top

<https://gateoverflow.in/8408>

The result evaluating the postfix expression $10\ 5 + 60\ 6 / * 8 -$ is

- A. 284
- B. 213
- C. 142
- D. 71

gate2015-3 data-structure stack normal

Answer

3.13.16 Stack: TIFR2017-B-3 top

<https://gateoverflow.in/95679>

We have an implementation that supports the following operations on a stack (in the instructions below, s is the name of the stack).

- **isempty(s)** : returns **True** if s is empty, and **False** otherwise.
- **top(s)** : returns the top element of the stack, but does not pop the stack; returns **null** if the stack is empty.
- **push(s, x)** : places x on top of the stack.
- **pop(s)** : pops the stack; does nothing if s is empty.

Consider the following code:

```

pop_ray_pop(x) :
    s=empty
    for i=1 to length(x):
        if (x[i] == '('):
            push(s, x[i])
        else:
            while (top(s)==('')):
                pop(s)
            end while
            push(s, ')')
        end if
    end for
    while not isempty(s):
        print top(s)
        pop(s)
    end while

```

What is the output of this program when

`pop_ray_pop ("(((((((((("`

is executed?

- A. (((
- B.)))) (((
- C.))))
- D. (((())))
- E. ()()

[tifr2017](#) [data-structure](#) [stack](#)

[Answer](#)

Answers: Stack

3.13.1 Stack: GATE1991-03,vii [top](#)

<https://gateoverflow.in/522>



Selected Answer

Let us try something different when you read the word pop then delete the last pushed element and print it. Now ,delete the push word which we have already executed. Now, go on from left to right and do the same.

So, output will be
20,20,10,10,20 .

1 21 votes

-- Bhagirathi Nayak (14.1k points)

3.13.2 Stack: GATE1994-1.14 [top](#)

<https://gateoverflow.in/2457>



Selected Answer

Push 1 push 2 push 3 pop 3 push 4 pop 4 push 5 pop 5 pop 2 pop 1 then o/p is 3,4,5,2,1

So, option is **B**.

1 23 votes

-- Sankaranarayanan P.N (11.5k points)

3.13.3 Stack: GATE1995-2.21 [top](#)

<https://gateoverflow.in/2633>



Selected Answer

$$A + B * (C + D) / F + D * E$$

answer = A B C D + * F / + D E * +

1 29 votes

-- Amar Vashishth (30.5k points)

3.13.4 Stack: GATE2000-13 [top](#)

<https://gateoverflow.in/684>



- Selected Answer
- For enqueue push operation is sufficient
For dequeue operation do the following
-reverse, pop, reverse
 - Contents of stack from top to bottom:
 - i) 7 10
 - ii) 2 5 7 10
 - iii) 80

23 votes

-- Pooja Palod (31.3k points)

3.13.5 Stack: GATE2003-64 top

<https://gateoverflow.in/951>

Selected Answer

Let us represent stack-life of i^{th} element as $S(i)$. The i^{th} element will be in stack till $(n - i)$ elements are pushed and popped. Plus one more Y for the time interval between the push of i_{th} element and the $i + 1^{th}$ element. So,

$$S(i) = Y + 2 \cdot (n - i)(Y + X) = Y + 2 \cdot (n - i)Z = Y + 2nZ - 2iZ$$

where $Z = Y + X$

$$\text{average stack-life will, } A = \Sigma \frac{S(i)}{n}$$

$$nA = nY + 2 \cdot n \cdot n \cdot Z - 2 \cdot Z \cdot \Sigma i$$

$$nA = nY + 2 \cdot n \cdot n \cdot Z - 2 \cdot Z \frac{(n(n+1))}{2}$$

$$nA = nY + 2 \cdot n \cdot n \cdot Z - Z(n \cdot n) - n \cdot Z$$

$$A = Y + 2 \cdot n \cdot Z - (n + 1) \cdot Z$$

$$A = Y + (n - 1) \cdot Z = Y + (n - 1)(X + Y) = n(X + Y) - X$$

31 votes

-- Vikrant Singh (13.5k points)

3.13.6 Stack: GATE2004-3 top

<https://gateoverflow.in/1000>

Selected Answer

Answer is (D).

Since the stacks are growing from opposite ends, initially $top1 = 1$ and $top2 = MAXSIZE$. Now, to make the space usage most efficient we should allow one stack to use the maximum possible space as long as other stack doesn't need it. So, either of the stack can grow as long as there is space on the array and hence the condition must be $top1 = top2 - 1$.

30 votes

-- Aditi Dan (5.2k points)

3.13.7 Stack: GATE2004-38, ISRO2009-27 [top](#)

<https://gateoverflow.in/1035>



Answer is (A).

Here is the procedure first :

Scan Infix Expression from left to right whenever you see operand just print it.

But, in case of operator

if(stack is empty) then push it.

if(precedence(tos) < precedence(current operator)) push it.

else if (precedence(tos) > precedence(current operator)) pop and print.

else if (precedence(tos) == precedence(current operator)) then check for associativity.In case Operators are Left to right then pop and print it otherwise push the current operator (Right to Left Associativity)

And once you have scanned infix expression completely. Make sure pop all the element and print it in same order.

Here, the infix expression is $a + b \times c - d^e^f$

a : print it

$+$: push into the Operator Stack

b : print it

$*$: its having higher precedence than $+$ then push into Operator Stack

c : print it

$'-$: ' $-$ ' is having less precedence than ' $*$ ' so pop from operator stack and print ' $*$ '.after this stack will be having ' $+$ ' on top.which is having same precedence as ' $-$ ' but both are left to right associative then just pop $+$ and print it. Now stack is empty. Push ' $-$ ' to it.

d : print it

$^{^}$: top of the stack is having ' $-$ ' which has lower precedence than ' $^{^}$ ' so simply push ' $^{^}$ ' into stack

e : print it.

$^{^}$: Now top of the stack is ' $^{^}$ ' and has same precedence so associativity will come to picture. Since ' $^{^}$ ' is right associative as given in question. So ' $^{^}$ ' will be pushed.

f : print it.

Now, we have scanned entire infix expression.Now pop the stack untill it becomes empty.This way you will get $abc * +def^{^} -$.

30 votes

-- IgnitorSandeep (495 points)

3.13.8 Stack: GATE2004-5 [top](#)

<https://gateoverflow.in/1002>



STACK scan the expression from left to right whenever a left parenthesis is encountered just PUSH it into stack and whenever a right parenthesis is encountered just POP it from stack. If at the end of expression we are left with an empty stack then it is a correctly parenthesized expression.

21 votes

-- Bhagirathi Nayak (14.1k points)

3.13.9 Stack: GATE2004-IT-52 [top](#)

<https://gateoverflow.in/3695>



- A. push a & pop a , push b & pop b , push c & pop c , and finally push d and pop d

- sequence of popped elements will come to *abcd*
- first push *abcd*, and after that pop one by one, sequence of popped elements will come to *dcb*
 - push *abc*, and after that pop one by one sequence of popped elements will come to *cba*, now push *d* and pop *d*, final sequence comes to *cbad*
 - this sequence is not possible because '*a*' can not be popped before '*b*' any how

19 votes

-- Manu Thakur (39,6k points)

3.13.10 Stack: GATE2005-IT-13 top

<https://gateoverflow.in/3758>



Selected Answer

- i* Element to be pushed
- Initial State $f(\phi) = 0$ For Empty Stack $F(S)$ is 0
- Then we push each element (*i*)one by one and calculate $f(s)$ for each insertion as given

$$F_{new}(S) = \max(F_{previous}(S), 0) + i$$

Is the function to compute $F(s)$ for each insertions

- INSERT 2 into Stack
 $F_{previous}(S) = 0$ [Stack was empty]
 $i = 2$ (inserting element is *i*)
 $F_{new}(S) = \max(F_{previous}(S), 0) + i$
 $F_{new}(S) = \max(0, 0) + 2 = 2$
- INSERT -3 into Stack
 $F_{previous}(S) = 2$ [Stack was empty]
 $i = -3$ (inserting element is *i*)
 $F_{new}(S) = \max(F_{previous}(S), 0) + i$
 $F_{new}(S) = \max(2, 0) + -3 = -1$

Similarly,

i : The element to be pushed
S: Stack

Initially $f(S)=0$.

$f(S)$	$\max(f(S), 0)$	<i>i</i>	Updated $f(S)=\max(f(S), 0)+i$
0	0	2	2
2	2	-3	-1
-1	0	2	2
2	2	-1	1
1	1	2	3

Thus, the answer is Option C.

The value of $F(S)$ after inserting all elements into stack is **3**.

53 votes

-- Shridhar (393 points)

3.13.11 Stack: GATE2007-38, ISRO2016-27 top

<https://gateoverflow.in/1236>



Selected Answer

push 8 so stack is 8
 push 2 so stack is 8 2
 push 8 2 3
 ^ pop 3 and 2 perform opn 2^3 and push to stack. stack is 8 8
 / pop 8 and 8 perform $8/8$ and push result to stack . stack is 1
 push 2 stack is 1 2
 push 3 stack is 1 2 3
 * pop 3 and 2 perform by $2 * 3$ and push . stack is 1 6
 Hence, answer is **A.**

26 votes

-- Sankaranarayanan P.N (11.5k points)

3.13.12 Stack: GATE2007-IT-32 top

<https://gateoverflow.in/3465>

Selected Answer

B) 25
 let first part
 5 ----push
 2-----push
 push----- $5 * 2 = 10$. (pops 5 and 2)
 push 3
 push 3
 push 2
 push $3 + 2 = 5$ (pops 2 and 3)
 push $5 * 3 = 15$ (pops (5 and 3)
 push $15 + 10 = 25$ (pops (15 and 10)

21 votes

-- Arpit Dhuriya (3.5k points)

3.13.13 Stack: GATE2014-2-41 top

<https://gateoverflow.in/2007>

Selected Answer

(C) is the answer. While *ENQUEUE* we *REVERSE* the stack, *PUSH* the element and then again *REVERSE* the stack. For *DEQUEUE* we simply *POP* the element.

(Option (B) can be used to get the first element from the stack by doing a *POP* after *REVERSE* for *DEQUEUE* and *PUSH* for *ENQUEUE*. But we have to restore the stack using *REVERSE* (otherwise next *POP* won't work) which means *DEQUEUE* actually needs 3 instructions and not 2)

33 votes

-- Arjun Suresh (350k points)

3.13.14 Stack: GATE2015-2-38 top

<https://gateoverflow.in/8164>



Answer: 15

The code is pushing 5 and 10 on stack and then popping the top two elements and printing their sum.

<http://ideone.com/kIUDqT>

24 votes

-- Rajarshi Sarkar (34.1k points)

3.13.15 Stack: GATE2015-3-12 top

<https://gateoverflow.in/8408>



We have to keep symbol into stack and when we get two operands followed by operator. We will apply operator on last two operands.

symbol	stack
10	10 (keep in stack)
5	10 5(keep in stack)
+	10 (apply operator on last 2 operands) \Rightarrow 5 + $\qquad\qquad\qquad$ $10 + 5$
60	$= 15$
6	15 60(keep in stack)
/	15 60(keep in stack)
*	6
150	15 60(apply operator on last 2 operands) \Rightarrow 6 / $\qquad\qquad\qquad$ $60 / 6$
8	$= 10$
150	15 10(apply operator on last 2 operands) \Rightarrow * $\qquad\qquad\qquad$ 10
-	* 15 $\qquad\qquad\qquad$ $* 15$
150	$= 150$
8	150 8(Keep in stack)
150	150 8 (apply operator on last 2 operands) \Rightarrow - $\qquad\qquad\qquad$ 150
-	$- 8$
150	$= 142$

So, answer is 142.

32 votes

-- Praveen Saini (54.6k points)

3.13.16 Stack: TIFR2017-B-3 top

<https://gateoverflow.in/95679>



(D) is option.

First push (((on stack. Now when) comes pop all (((and push) on stack. Now push ((and stack become)((. Now when) come it pop all ((from stack and new stack become)). Again) comes and stack become))). Now push (((on stack and the stack becomes ((())). Now pop one by one and get option D as the answer.

11 votes

-- Mehak Sharma (1.6k points)

3.14

Trees(15) top

3.14.1 Trees: GATE1990-13a top

<https://gateoverflow.in/86224>

Consider the height-balanced tree T_t with values stored at only the leaf nodes, shown in Fig.4.

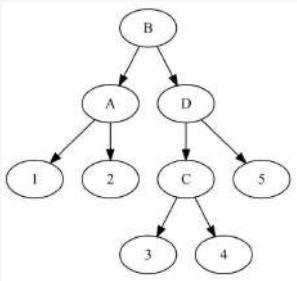


Fig. 4

(i) Show how to merge to the tree, T_1 elements from tree T_2 shown in Fig.5 using node D of tree T_1 .

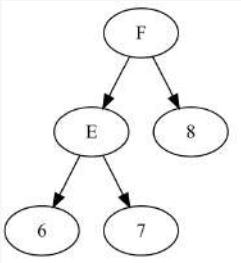


Fig. 5

(ii) What is the time complexity of a merge operation of balanced trees T_1 and T_2 where T_1 and T_2 are of height h_1 and h_2 respectively, assuming that rotation schemes are given. Give reasons.

gate1990 descriptive data-structure trees

Answer

3.14.2 Trees: GATE1992-02,vii top

<https://gateoverflow.in/562>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

A **2 – 3** tree is such that

- a. All internal nodes have either 2 or 3 children
- b. All paths from root to the leaves have the same length.

The number of internal nodes of a **2 – 3** tree having 9 leaves could be

- A. 4
- B. 5
- C. 6
- D. 7

[gate1992](#) [trees](#) [data-structure](#) [normal](#)

[Answer](#)

3.14.3 Trees: GATE1994-5 [top](#)

<https://gateoverflow.in/2501>

A **3 – ary** tree is a tree in which every internal node has exactly three children. Use induction to prove that the number of leaves in a **3 – ary** tree with n internal nodes is $2(n - 1)$.

[gate1994](#) [data-structure](#) [trees](#) [proof](#)

[Answer](#)

3.14.4 Trees: GATE1998-1.24 [top](#)

<https://gateoverflow.in/1661>

Which of the following statements is false?

- A. A tree with n nodes has $(n-1)$ edges
- B. A labeled rooted binary tree can be uniquely constructed given its postorder and preorder traversal results.
- C. A complete binary tree with n internal nodes has $(n+1)$ leaves.
- D. The maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$

[gate1998](#) [data-structure](#) [trees](#) [normal](#)

[Answer](#)

3.14.5 Trees: GATE1998-2.11 [top](#)

<https://gateoverflow.in/1683>

A complete n -ary tree is one in which every node has 0 or n sons. If x is the number of internal nodes of a complete n -ary tree, the number of leaves in it is given by

- A. $x(n - 1) + 1$
- B. $xn - 1$
- C. $xn + 1$
- D. $x(n + 1)$

[gate1998](#) [data-structure](#) [trees](#) [normal](#)

[Answer](#)

3.14.6 Trees: GATE1998-21 [top](#)

<https://gateoverflow.in/1735>

- A. Derive a recurrence relation for the size of the smallest AVL tree with height h .
- B. What is the size of the smallest AVL tree with height 8?

[gate1998](#) [data-structure](#) [trees](#) [descriptive](#) [numerical-answers](#)

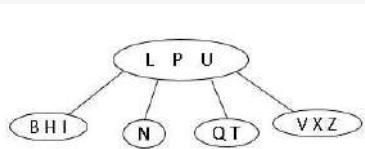
Answer**3.14.7 Trees: GATE2002-2.9** top<https://gateoverflow.in/839>

The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is:

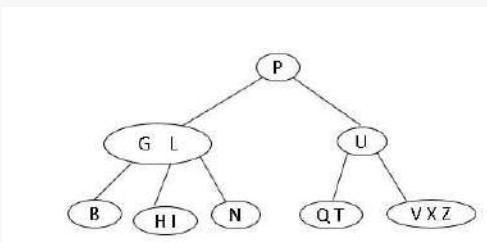
- A. $\frac{n}{2}$
- B. $\frac{(n-1)}{3}$
- C. $\frac{(n-1)}{2}$
- D. $\frac{(2n+1)}{3}$

[gate2002](#) [data-structure](#) [trees](#) [normal](#)
Answer**3.14.8 Trees: GATE2003-65** top<https://gateoverflow.in/952>

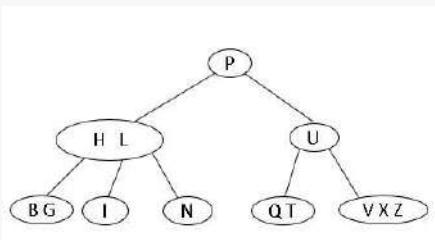
Consider the following **$2 - 3 - 4$** tree (i.e., B-tree with a minimum degree of two) in which each data item is a letter. The usual alphabetical ordering of letters is used in constructing the tree.



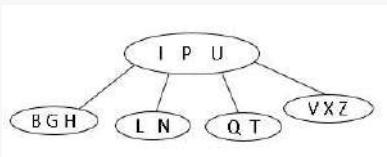
What is the result of inserting **G** in the above tree?



A



B



C

- D. None of the above

[gate2003](#) [algorithms](#) [trees](#) [normal](#)

[Answer](#)

3.14.9 Trees: GATE2004-6 [top](#)

<https://gateoverflow.in/1003>

Level order traversal of a rooted tree can be done by starting from the root and performing

- A. preorder traversal
- B. in-order traversal
- C. depth first search
- D. breadth first search

[gate2004](#) [data-structure](#) [trees](#) [easy](#)

[Answer](#)

3.14.10 Trees: GATE2005-36 [top](#)

<https://gateoverflow.in/1372>

In a complete k -ary tree, every internal node has exactly k children. The number of leaves in such a tree with n internal node is:

- A. nk
- B. $(n - 1)k + 1$
- C. $n(k - 1) + 1$
- D. $n(k - 1)$

[gate2005](#) [data-structure](#) [trees](#) [normal](#)

[Answer](#)

3.14.11 Trees: GATE2007-43 [top](#)

<https://gateoverflow.in/1241>

A complete $n - ary$ tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete $n - ary$ tree. If $L = 41$ and $I = 10$, what is the value of n ?

- A. 3
- B. 4
- C. 5
- D. 6

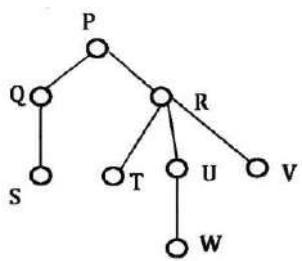
[gate2007](#) [data-structure](#) [trees](#) [normal](#)

[Answer](#)

3.14.12 Trees: GATE2014-3-12 [top](#)

<https://gateoverflow.in/2046>

Consider the following rooted tree with the vertex labeled P as the root:



The order in which the nodes are visited during an in-order traversal of the tree is

- A. $SQPTRWUV$
- B. $SQPTUWRV$
- C. $SQPTWUVR$
- D. $SQPTRUWV$

gate2014-3 data-structure trees easy

Answer

3.14.13 Trees: GATE2014-3-41 [top](#)

<https://gateoverflow.in/2075>

Consider the pseudocode given below. The function $DoSomething()$ takes as argument a pointer to the root of an arbitrary tree represented by the $leftMostChild - rightSibling$ representation. Each node of the tree is of type *treeNode*.

```

typedef struct treeNode* treeptr;

struct treeNode
{
    treeptr leftMostChild, rightSibling;
};

int DoSomething (treeptr tree)
{
    int value=0;
    if (tree != NULL) {
        if (tree->leftMostChild == NULL)
            value = 1;
        else
            value = DoSomething(tree->leftMostChild);
        value = value + DoSomething(tree->rightSibling);
    }
    return (value);
}
  
```

When the pointer to the root of a tree is passed as the argument to $DoSomething$, the value returned by the function corresponds to the

- A. number of internal nodes in the tree.
- B. height of the tree.
- C. number of nodes without a right sibling in the tree.
- D. number of leaf nodes in the tree

gate2014-3 data-structure trees normal

Answer

3.14.14 Trees: GATE2017-1-20 [top](#)

<https://gateoverflow.in/118300>

Let T be a tree with 10 vertices. The sum of the degrees of all the vertices in T is _____

[gate2017-1](#) [data-structure](#) [trees](#) [numerical-answers](#)
Answer**3.14.15 Trees: TIFR2012-B-15** [top](#)<https://gateoverflow.in/25212>

Let T be a tree of n nodes. Consider the following algorithm, that constructs a sequence of leaves u_1, u_2, \dots . Let u_1 be some leaf of tree. Let u_2 be a leaf that is farthest from u_1 . Let u_3 be the leaf that is farthest from u_2 , and, in general, let u_{i+1} be a leaf of T that is farthest from u_i (if there are many choices for u_{i+1} , pick one arbitrarily). The algorithm stops when some u_i is visited again. What can you say about the distance between u_i and u_{i+1} , as $i = 1, 2, \dots$?

- A. For some trees, the distance strictly reduces in each step.
- B. For some trees, the distance increases initially and then decreases.
- C. For all trees, the path connecting u_2 and u_3 is a longest path in the tree.
- D. For some trees, the distance reduces initially, but then stays constant.
- E. For the same tree, the distance between the last two vertices visited can be different, based on the choice of the first leaf u_1 .

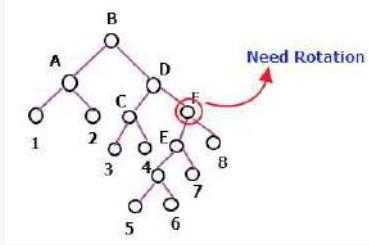
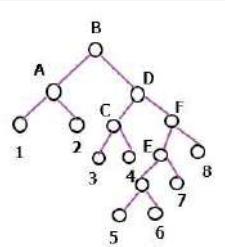
[tifr2012](#) [data-structure](#) [trees](#)
Answer**Answers: Trees****3.14.1 Trees: GATE1990-13a** [top](#)<https://gateoverflow.in/86224>

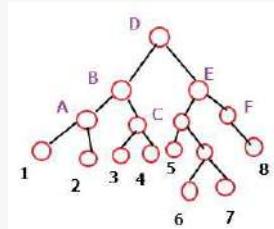
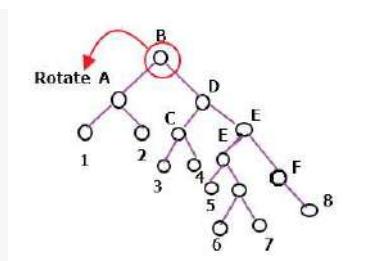
We have to merge T_1 and T_2

Only the D node will be participate in merging

All the values of the tree are in leaf node

and it should be height balanced





Complexity will be $O(n \log n)$

As here all nodes of 2nd tree needs to be insert in tree and rotation of nodes needed.

2 votes

-- srestha (87k points)

3.14.2 Trees: GATE1992-02, vii top

<https://gateoverflow.in/562>



Selected Answer

Answer (a) 4, (d) 7

4 → When each leaf has 3 childs. So $9/3 = 3$ Internal nodes, Then one internal node those internal nodes.

7 → When each leaf has 2 childs & one leaf out of 4 get 3 childs. Ex $\rightarrow 8/4 = 2$ child per internal node. Then one of that internal node get extra third child. Then 2 internal nodes to connect these 4. Then 1 internal node to connect this 2. So $4 + 2 + 1 = 7$.

No other way is possible.

15 votes

-- Akash Kanase (42.5k points)

3.14.3 Trees: GATE1994-5 top

<https://gateoverflow.in/2501>

No of nodes at level $i = 3^i$

Let height of tree be h

So total no of internal nodes $= 3^0 + 3^1 + 3^2 + \dots + 3^{h-1} = \frac{3^h - 1}{2}$

$2n = 3^h - 1$

No of leaf nodes $= 3^h = 2n + 1 = 2(n - 1) + 3$

Let us prove by induction

Base case

$n = 1$ (one internal node i.e., root node)

$$\text{No of leaves} = 2(1 - 1) + 3 = 3$$

Let it be true for n internal nodes

Now we prove for m nodes where $m = n + 1$

$$\text{We have } L(m) = 2(n + 1 - 1) + 3$$

$$\text{Also } L(m) = L(n) + 3 - 1 = 2(n - 1) + 3 + 3 - 1 = 2n + 3$$

So if $L(n)$ is true then $L(n + 1)$ is also true

Hence proved by induction.

8 votes

-- Pooja Palod (31.3k points)

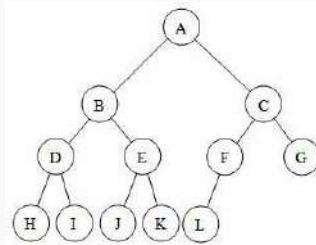
3.14.4 Trees: GATE1998-1.24 top

<https://gateoverflow.in/1661>



Selected Answer

- Tree with n nodes must have $n - 1$ edges.
- A labeled rooted binary tree can be uniquely constructed given its postorder and preorder traversal results. (inorder must be needed with either preorder or postorder for that)
- A complete binary tree with n nodes can have n leaves also
- Example:



Since: **A complete binary tree** is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible. So **false**

- The maximum number of nodes in a binary tree of height h is

$$1 + 2 + 4 + \dots + 2^h = 2^{h+1} - 1 \text{ So true}$$

Answer is **b** and **c** both.

Since, 2 answers are there I would choose b, because in some places by "complete" binary tree they mean "full binary tree" which has all the levels completely filled.

16 votes

-- Prashant Singh (59.8k points)

3.14.5 Trees: GATE1998-2.11 top

<https://gateoverflow.in/1683>



Selected Answer

Answer : → (A)

$$x(n - 1) + 1$$

Originally when we have root , there is only 1 node, which is leaf. (There is no internal node.)
From this base case "+1" part of formula comes.

When we n children to root, we make root internal. So then Total Leaves
 $= 1(n - 1) + 1 = n$.

In complete n ary tree every time you add n children to node, you add $n - 1$ leaves & make that node to which you are inserting children internal.($+n$ for leaves, -1 for node which you are attaching). So if you had originally few leaves, you add $n - 1$ "New" leaves to them. This is how $x(n - 1) + 1$ makes sense.

19 votes

-- Akash Kanase (42.5k points)

3.14.6 Trees: GATE1998-21 top

<https://gateoverflow.in/1735>



Selected Answer

- a. Consider a function $N(h)$ which represents the smallest number of nodes n for an AVL tree with height h and satisfies $n = N(h)$.

For $h = 0$ we have, number of nodes = 1. So $N(0) = 1$

For $h = 1$, we have, number of nodes = 2. We could take 3, but we require the smallest graph (a graph with smallest number of nodes) so we take 2. It means that to create a tree with height 1 we need **at least 2** nodes.

So $N(1) = 2$

Now, for $h = 2$, we need to create a node with a child subtree of height 1. This may be the right or left subtree. But since this is an AVL tree, to balance a child subtree of height let's say H_s , we need the other child to have a height of $H_s - 1$, H_s or $H_s + 1$. But we take $H_s - 1$ for minimal case. In simple words, a node with height 5 must have a child with height 4(H_s) and another child with height 3($H_s - 1$). So $N(2)$ can be obtained as:

$N(2) = N(1) + (0) + 1$ (1 is added to count the parent node, $N(1)$ or $N(H_s)$ and $N(0)$ or $N(H_s - 1)$ represent two subtrees.)

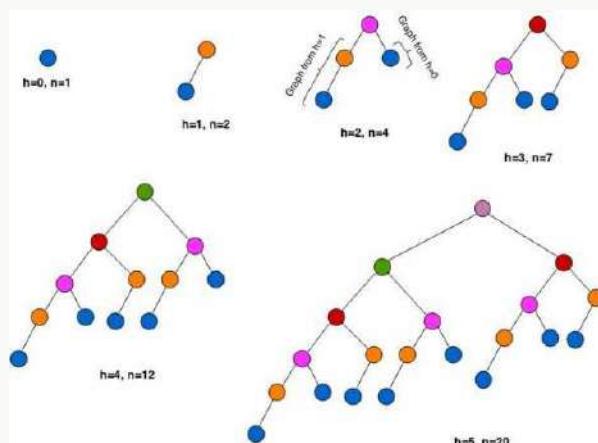
Similarly:

$$N(3) = N(2) + N(1) + 1$$

and generalizing:

$$N(h) = N(h - 1) + N(h - 2) + 1$$

This recursion can be graphically seen as below:



- b. Using the above recursion, we need to find $N(8)$

$$\begin{aligned}
 N(0) &= 1 \\
 N(1) &= 2 \\
 N(2) &= N(1) + N(0) + 1 = 1 + 2 + 1 = 4 \\
 N(3) &= N(2) + N(1) + 1 = 2 + 4 + 1 = 7 \\
 N(4) &= N(3) + N(2) + 1 = 4 + 7 + 1 = 12 \\
 N(5) &= N(4) + N(3) + 1 = 7 + 12 + 1 = 20 \\
 N(6) &= N(5) + N(4) + 1 = 12 + 20 + 1 = 33 \\
 N(7) &= N(6) + N(5) + 1 = 20 + 33 + 1 = 54 \\
 N(8) &= N(7) + N(6) + 1 = 33 + 54 + 1 = 88
 \end{aligned}$$

So, answer for **b)** is 88.

20 votes

-- Ashis Kumar Sahoo (865 points)

3.14.7 Trees: GATE2002-2.9 top

<https://gateoverflow.in/839>



L = leaf nodes

I = internal nodes

T = total nodes = $L + I$

in a tree no. of edges = $T - 1$

all edges are produced by only internal nodes so

$k \times I = T - 1 \rightarrow (1)$ (for k -ary tree, in this question $k = 3$)

$L + I = T \rightarrow (2)$

Solving (1) and (2) we get

$L = ((k - 1)T + 1)/k$

So put $k = 3, T = n$

you get $L = (2n + 1)/3$

Answer is **D**.

31 votes

-- Vikrant Singh (13.5k points)

3.14.8 Trees: GATE2003-65 top

<https://gateoverflow.in/952>



(B) is the correct answer.

Once we add G , the leaf node becomes $B G H I$, since we can have only 3 keys. the node has to split at G or H , and G or H will be added to parent node.

Since P is the parent node in options 1 and 2, its evident the 3rd element i.e. H should be selected for splitting (because after adding any key from the leftmost child node, P becomes the 3rd element in the node)

Now parent node becomes $H L P U$, select P as for splitting, and you get option B.

Hence, **answer is B.**

15 votes

-- ryan sequeira (3.4k points)

3.14.9 Trees: GATE2004-6 top

<https://gateoverflow.in/1003>



Answer is **option D.**

Breadth first search.

16 votes

-- anshu (3.3k points)

3.14.10 Trees: GATE2005-36 top

<https://gateoverflow.in/1372>



Answer :-> **C) $n(k - 1) + 1$**

Originally when we have root , there is only 1 node, which is leaf.(There is no internal node.) From that "+1" part of formula comes from this base case.

When we k children to nodes, we make root internal. So then Total Leaves
 $= n(k - 1) + 1 = (k - 1) + 1 = k$

In k complete k ary tree every time you add k children , you add $k - 1$ leaves.(+ k for leaves, -1 for node which you are attaching)

12 votes

-- Akash Kanase (42.5k points)

3.14.11 Trees: GATE2007-43 top

<https://gateoverflow.in/1241>



If you do little bit experiments on no of leaves, Internal nodes, you will realize that they have equation like following :-

No of leaves (L) = $(n - 1) * \text{Internal Nodes } (I) + 1$

Here we need to find n .

Putting values

$$\begin{aligned} 41 &= (n - 1) * 10 + 1 \\ \implies (n - 1) * 10 &= 40 \\ \implies n - 1 &= 4 \\ \implies n &= 5 \end{aligned}$$

So, answer is **C.**

20 votes

-- Akash Kanase (42.5k points)

Sum of degrees in tree = $L + I \times (n + 1) - 1 = 10n + 50$ (Each leaf node has degree 1 and all internal nodes have degree $k + 1$, except root which has degree k)

So, number of edges = $5n + 25$ (Number of edges in a graph (hence applicable for tree also) is half the sum of degrees as each edge contribute 2 to the sum of degrees)

In a tree with n nodes we have $n - 1$ edges, so with $41 + 10 = 51$ nodes, there must be 50 edges.

$$\text{So, } 5n + 25 = 50$$

$$\implies 5n = 25$$

$$\implies n = 5$$

25 votes

-- Arjun Suresh (350k points)

3.14.12 Trees: GATE2014-3-12 top

<https://gateoverflow.in/2046>


Selected Answer

(A) the inorder traversal order of a ternary tree is left → root → middle → right.

30 votes

-- Gate Keeda (19.6k points)

3.14.13 Trees: GATE2014-3-41 top

<https://gateoverflow.in/2075>


Selected Answer

Here, the condition for count value = 1 is

if ($tree \rightarrow leftMostchild == Null$)

- so, if there is no left-most child of the tree (or the sub-tree or the current node called in recursion)
- Which means there is no child to that particular node (since if there is no left-most child, there is no child at all as per the tree representation given).
- ∴ the node under consideration is a leaf node.
- The function recursively counts, and adds to value, whenever a leaf node is encountered.
- So, The function returns the number of leaf nodes in the tree.

20 votes

-- Kalpish Singhal (2.1k points)

3.14.14 Trees: GATE2017-1-20 top

<https://gateoverflow.in/118300>


Selected Answer

Tree with n vertices which means $n - 1$ edges.

$$n = 10 \therefore edges = n - 1 = 9 .$$

$$\therefore \text{Sum of degree of all vertices} \leq 2E \leq 2 * 9 \leq 18$$

👍 24 votes

-- Kantikumar (4.6k points)

3.14.15 Trees: TIFR2012-B-15 top

<https://gateoverflow.in/25212>

This post explains it nicely:

<https://www.quora.com/How-does-following-algorithm-for-finding-longest-path-in-tree-work>

👍 4 votes

-- sudepta roy (423 points)

4 Programming & DS: Programming (104) top

4.1

Aliasing(1) top

4.1.1 Aliasing: GATE2000-1.16 top

<https://gateoverflow.in/639>

Aliasing in the context of programming languages refers to

- A. multiple variables having the same memory location
- B. multiple variables having the same value
- C. multiple variables having the same identifier
- D. multiple uses of the same variable

[gate2000](#) [programming](#) [easy](#) [aliasing](#)
[Answer](#)

Answers: Aliasing

4.1.1 Aliasing: GATE2000-1.16 top

<https://gateoverflow.in/639>

Selected Answer

Option is **A.**

In computer **programming**, **aliasing** refers to the situation where the **same memory location** can be accessed using different names. For instance, if a function takes two pointers A and B which have the same value, then the name **A aliases** the name **B**.

[18 votes](#)
-- Prasanna Ranganathan (4.7k points)

4.2

Goto(2) top

4.2.1 Goto: GATE1989-3-i top

<https://gateoverflow.in/87095>

An unrestricted use of the "go to" statement is harmful because of which of the following reason (s):

- A. It makes it more difficult to verify programs.
- B. It makes programs more inefficient.
- C. It makes it more difficult to modify existing programs.
- D. It results in the compiler generating longer machine code.

[gate1989](#) [normal](#) [programming](#) [goto](#)
[Answer](#)

4.2.2 Goto: GATE1994-1.5 top

<https://gateoverflow.in/2442>

An unrestricted use of the "*goto*" statement is harmful because

- A. it makes it more difficult to verify programs
 B. it increases the running time of the programs
 C. it increases the memory required for the programs
 D. it results in the compiler generating longer machine code

gate1994 programming easy goto

[Answer](#)

Answers: Goto

4.2.1 Goto: GATE1989-3-i [top](#)

<https://gateoverflow.in/87095>



Selected Answer

Solution : A) It makes it more difficult to verify programs.

Proof of correctness : <https://en.wikipedia.org/wiki/Goto#Criticism>
(1st Paragraph, last 4 lines.)

Option B: goto has no role in making a program inefficient. Adding a "goto" introduces a branch instruction but assuming same program logic this can never be avoided and even when replaced by a proper structural construct like for/while/if -- will still have the branch instruction.

Option C: Actually using "goto" kind of makes it easy to modify an existing program as to make the control flow from say line number 'x' to line number 'y', we can simply do a "goto". This is much harder to do in a proper structural way.

Option D: This is also false as goto introduces only a single "JUMP" instruction (machine code will be its equivalent; OPCODE for JMP followed by the destination address) and equivalent structural constructs will have longer machine codes.

2 votes

-- Siddharth Mahapatra (1.3k points)

4.2.2 Goto: GATE1994-1.5 [top](#)

<https://gateoverflow.in/2442>



Selected Answer

Use of *goto* takes out the structural decomposition of the code and hence it becomes very difficult to verify or debug the code. As far as performance or memory impact is concerned, *goto* has no effect on them.

25 votes

-- Arjun Suresh (350k points)

4.3

Identify Function(4) [top](#)

4.3.1 Identify Function: GATE1995-3 [top](#)

<https://gateoverflow.in/2639>

Consider the following high level programming segment. Give the contents of the memory locations for variables *W*, *X*, *Y* and *Z* after the execution of the program segment. The values of the variables *A* and *B* are *5CH* and *92H*, respectively. Also indicate error conditions if any.

```
var
  A, B, W, X, Y :unsigned byte;
  Z             :unsigned integer, (each integer is represented by two bytes)
begin
```

```

X           :=A+B
Y           :=abs (A-B) ;
W           :=A-B
Z           :=A*B
end;

```

gate1995 | programming | normal | identify-function

Answer

4.3.2 Identify Function: GATE1998-2.13 [top](#)

<https://gateoverflow.in/1685>

What is the result of the following program?

```

program side-effect (input, output);
var x, result: integer;
function f (var x:integer:integer;
begin
  x:=x+1; f:=x;
end
begin
  x:=5;
  result:=f (x) *f (x);
  writeln(result);
end

```

- A. 5
- B. 25
- C. 36
- D. 42

gate1998 | programming | normal | identify-function

Answer

4.3.3 Identify Function: GATE2004-IT-15 [top](#)

<https://gateoverflow.in/3656>

Let x be an integer which can take a value of 0 or 1. The statement

```
if (x == 0) x = 1; else x = 0;
```

is equivalent to which one of the following ?

- A. $x = 1 + x;$
- B. $x = 1 - x;$
- C. $x = x - 1;$
- D. $x = 1 \% x;$

gate2004-it | programming | easy | identify-function

Answer

4.3.4 Identify Function: GATE2017-2-43 [top](#)

<https://gateoverflow.in/118388>

Consider the following snippet of a C program. Assume that swap ($\&x, \&y$) exchanges the content of x and y :

```

int main () {
    int array[] = {3, 5, 1, 4, 6, 2};
    int done =0;
}

```

```

int i;
while (done==0) {
    done =1;
    for (i=0; i<=4; i++) {
        if (array[i] < array[i+1]) {
            swap(&array[i], &array[i+1]);
            done=0;
        }
    }
    for (i=5; i>=1; i--) {
        if (array[i] > array[i-1]) {
            swap(&array[i], array[i-1]);
            done =0;
        }
    }
}
printf("%d", array[3]);
}

```

The output of the program is _____

[gate2017-2](#) [programming](#) [algorithms](#) [numerical-answers](#) [identify-function](#)

[Answer](#)

Answers: Identify Function

4.3.1 Identify Function: GATE1995-3 [top](#)

<https://gateoverflow.in/2639>

Max. value that can be accommodated inside an unsigned byte = 255 and unsigned int = 65535

A and B are given in Hexadecimal.

$$A = 5CH = (92)_{10}$$

$$B = 92H = (146)_{10}$$

$$X = A + B = (238)_{10} = EEH$$

$$Y = \text{abs}(A - B) = (54)_{10} = 36H$$

$$W = A - B = (-54)_{10} = CAH$$

$$Z = A * B = (13432)_{10} = 3478H$$

12 votes

-- Ravi Ranjan (3.2k points)

4.3.2 Identify Function: GATE1998-2.13 [top](#)

<https://gateoverflow.in/1685>



Selected Answer

Call by value: 36,

Call by reference: undefined behaviour for C/C++ but 42 for languages having * as a sequence point.

$$f(x) * f(x);$$

If the value of x is being modified inside the function (call by reference) we cannot be sure if this modified value or the old value will be passed as argument for the second call to $f()$. This is because left and right operand of any arithmetic expression in C/C++ can be evaluated in any order. For languages like Java, strict left-right order is maintained.

7 votes

-- Arjun Suresh (350k points)

4.3.3 Identify Function: GATE2004-IT-15 top

<https://gateoverflow.in/3656>



Selected Answer

Firstly, our requirement is for $x = 1$ it makes '0' and for $x = 0$ it makes '1'

Let's consider options one by one:

A. $X = 1 + X$
For $x = 1$, it gives 2 So, False

B. $X = 1 - X$
Here B is correct, as
For $x = 0$, it gives 1.
For $x = 1$, it gives 0.

C. $X = X - 1$
For $x = 0$, it gives -1. So, False

D. $X = 1 \% X$
For $x = 0$, it gives 1%0 . I think it is undefined.
Even if we consider $X = X \% 1$
Then for $x = 0$, it gives 0%1 = 0 But we require 1.

So, Option (B) is correct.

18 votes

-- Himanshu Agarwal (15.3k points)

4.3.4 Identify Function: GATE2017-2-43 top

<https://gateoverflow.in/118388>



Selected Answer

Well, the above program is sorting the array in **descending** order.

Initially, *while loop* starts execution by *evaluating the initial condition*

```
while(done==0)
```

For the first time the first for loop will be executed completely, the content of array will be as follows :

5,3,4,6,2,1

After the second for executed completely the content of array will be as follows:

6,5,3,4,2,1

But the value variable **done** is still 0 so while loop will execute again, so now the content of array after executing the first for loop will be 6,5,4,3,2,1 and no change in second for loop but still the

done variable is 0.

So, while loop execute again, now **done** variable is modified to 1 and there will be no change in done variable because inside first and second for loop no if condition will satisfied .

Finally, the while condition is evaluated false and value of *array*[3] will be printed which is 3.

1 31 votes

-- Manoj Kumar (38.6k points)

4.4

Loop Invariants(8) top

4.4.1 Loop Invariants: GATE1991-1,vi top

<https://gateoverflow.in/504>

Consider the following PASCAL program segment:

```
if i mod 2 = 0 then
  while i >= 0 do
    begin
      i := i div 2;
      if i mod 2 < > 0 then i := i - 1;
      else i := i - 2;
    end;
```

An appropriate loop-invariant for the while-loop is _____

gate1991 programming loop-invariants normal

Answer

4.4.2 Loop Invariants: GATE2004-32 top

<https://gateoverflow.in/1029>

Consider the following program fragment for reversing the digits in a given integer to obtain a new integer.

Let $n = d_1 d_2 \dots d_m$.

```
int n, rev;
rev = 0;
while(n > 0) {
    rev = rev * 10 + n%10;
    n = n/10;
}
```

The loop invariant condition at the end of the i^{th} iteration is:

- A. $n = d_1 d_2 \dots d_{m-i}$ and $\text{rev} = d_m d_{m-1} \dots d_{m-i+1}$
- B. $n = d_{m-i+1} \dots d_{m-1} d_m$ or $\text{rev} = d_{m-i} \dots d_2 d_1$
- C. $n \neq \text{rev}$
- D. $n = d_1 d_2 \dots d_m$ or $\text{rev} = d_m \dots d_2 d_1$

gate2004 programming loop-invariants normal

Answer

4.4.3 Loop Invariants: GATE2015-1-33 top

<https://gateoverflow.in/3276>

Consider the following pseudo code, where x and y are positive integers.

```
begin
    q := 0
    r := x
    while r ≥ y do
        begin
            r := r - y
            q := q + 1
        end
    end
```

The post condition that needs to be satisfied after the program terminates is

- A. $\{r = qx + y \wedge r < y\}$
- B. $\{x = qy + r \wedge r < y\}$
- C. $\{y = qx + r \wedge 0 < r < y\}$
- D. $\{q + 1 < r - y \wedge y > 0\}$

gate2015-1 programming loop-invariants normal

Answer

4.4.4 Loop Invariants: GATE2016-2-35 [top](#)

<https://gateoverflow.in/39578>

The following function computes X^Y for positive integers X and Y .

```
int exp (int X, int Y) {
    int res = 1, a = X, b = Y;

    while (b != 0) {
        if (b % 2 == 0) {a = a * a; b = b/2; }
        else             {res = res * a; b = b - 1; }
    }
    return res;
}
```

Which one of the following conditions is TRUE before every iteration of the loop?

- A. $X^Y = a^b$
- B. $(res * a)^Y = (res * X)^b$
- C. $X^Y = res * a^b$
- D. $X^Y = (res * a)^b$

gate2016-2 programming loop-invariants normal

Answer

4.4.5 Loop Invariants: GATE2017-2-37 [top](#)

<https://gateoverflow.in/118381>

Consider the C program fragment below which is meant to divide x by y using repeated subtractions. The variables x , y , q and r are all unsigned int.

```
while (r >= y) {
    r=r-y;
    q=q+1;
}
```

Which of the following conditions on the variables x, y, q and r before the execution of the fragment will ensure that the loop terminated in a state satisfying the condition $x == (y * q + r)$?

- A. $(q == r) \&\& (r == 0)$
 B. $(x > 0) \&\& (r == x) \&\& (y > 0)$
 C. $(q == 0) \&\& (r == x) \&\& (y > 0)$
 D. $(q == 0) \&\& (y > 0)$

gate2017-2 programming loop-invariants

Answer

4.4.6 Loop Invariants: TIFR2010-B-30 [top](#)

<https://gateoverflow.in/19042>

Consider the following program for summing the entries of the array b : array $[0..N-1]$ of integers, where N is a positive integer. (The symbol ' $\langle\neq\rangle$ ' denotes 'not equal to').

```
var
    i, s: integer;
Program
    i := 0;
    s := 0;
[*] while i <> N do
    s := s + b[i];
    i := i + 1;
od
```

Which of the following gives the invariant that holds at the beginning of each loop, that is, each time the program arrives at point [*] ?

- A. $s = \sum_{j=0}^N b[j] \& 0 \leq i \leq N$
 B. $s = \sum_{j=0}^i b[j] \& 0 \leq i < N$
 C. $s = \sum_{j=0}^i b[j] \& 0 < i \leq N$
 D. $s = \sum_{j=1}^N b[j] \& 0 \leq i < N$
 E. $s = \sum_{j=0}^{i-1} b[j] \& 0 \leq i \leq N$

tifr2010 programming loop-invariants

Answer

4.4.7 Loop Invariants: TIFR2010-B-37 [top](#)

<https://gateoverflow.in/19251>

Consider the program where a, b are integers with $b > 0$.

```
x:=a; y:=b; z:=0;
while y > 0 do
    if odd (x) then
        z:= z + x;
        y:= y - 1;
    else y:= y % 2;
        x:= 2 * x;
    fi
```

Invariant of the loop is a condition which is true before and after every iteration of the loop. In the

above program the loop invariant is given by

$$0 \leq y \text{ and } z + x * y = a * b$$

Which of the following is true of the program?

- A. The program will not terminate for some values of a, b .
- B. The program will terminate with $z = 2^b$
- C. The program will terminate with $z = a * b$.
- D. The program will not terminate for some values of a, b but when it does terminate, the condition $z = a * b$ will hold.
- E. The program will terminate with $z = a^b$

[tifr2010](#) | [programming](#) | [loop-invariants](#)

[Answer](#)

4.4.8 Loop Invariants: TIFR2017-B-5 [top](#)

<https://gateoverflow.in/95683>

Consider the following psuedocode fragment, where y is an integer that has been initialized.

```
int i=1
int j=1
while (i<10):
    j=j*i
    i=i+1
    if (i==y):
        break
    end if
end while
```

Consider the following statements:

- i. $(i == 10)$ or $(i == y)$
- ii. If $y > 10$, then $i == 10$
- iii. If $j = 6$, then $y == 4$

Which of the above statements is/are TRUE at the end of the while loop? Choose from the following options.

- A. i only
- B. iii only
- C. ii and iii only
- D. i, ii, and iii
- E. None of the above

[tifr2017](#) | [programming](#) | [loop-invariants](#)

[Answer](#)

Answers: Loop Invariants

4.4.1 Loop Invariants: GATE1991-1,vi [top](#)

<https://gateoverflow.in/504>



Selected Answer

Loop invariant is some condition which holds at the end of each iteration of the loop. i.e. it is "*invariant*" => does not vary (or change). It might change inside one iteration, but it will be true at

the end of every iteration.

We often use loop invariants to prove that our algorithm works correctly.

In the given program, a loop invariant is:

$i \bmod 2 = 0$

i.e. **i is even after every iteration.**

One can verify this as follows:

- Before the execution of first iteration the loop invariant is true, because of this line of code:

```
if i mod 2 = 0 then
```

- In every iteration, we divide i by 2, so now i will be either odd or even.
 - If odd, we subtract 1 from i

```
if i mod 2 < > 0 then i := i - 1;
```

- so it's now even.

- otherwise, if even, we subtract 2 from i

```
else i := i - 2;
```

- so, it remains even

- So, at the end of every iteration *i remains even.*

<https://stackoverflow.com/questions/3221577/what-is-a-loop-invariant>

2 votes

-- Rishabh Gupta (13.9k points)

4.4.2 Loop Invariants: GATE2004-32 top

<https://gateoverflow.in/1029>



Selected Answer

A loop invariant is something that hold at the start of a loop, across each iteration (inside an iteration it can change but before the iteration ends original condition must be true) and at the end also. So, we can check for the satisfiability of the condition at the loop header for start of the loop, for each iteration and also at the exit.

Here, in each iteration the right most digit of n, is moving to the right end of rev. So, answer is (A). i.e. the 2 conditions given in (A) choice are true on entry to loop, after each iteration (not necessarily during an iteration), and at end of loop.

17 votes

-- Arjun Suresh (350k points)

4.4.3 Loop Invariants: GATE2015-1-33 top

<https://gateoverflow.in/8276>



Selected Answer

The loop terminates when $r < y$. So, $r < y$ is one post condition.

In each iteration q is incremented by 1 and y is subtracted from r . Initial value of r is x . So, loop iterates x/y times and q will be equal to x/y and $r = x \% y \Rightarrow x = qy + r$;

So, (B) choice.

36 votes

-- Arjun Suresh (350k points)

4.4.4 Loop Invariants: GATE2016-2-35 top

<https://gateoverflow.in/39578>



Selected Answer

Take $X = 10, Y = 3$

In that case,

Before Iteration 1:

$$res = 1, a = 10, b = 3$$

All options are satisfied here.

Iteration 1:

- $while(b! = 0) \implies 3! = 0 \quad \checkmark$
- $if(b \% 2 == 0) \implies 3 \% 2 == 0 \quad \times$
- $else$
 $res = res * a \implies res = 1 * 10 = 10$
 $b = b - 1 \implies b = 3 - 1 = 2$

Before Iteration 2:

$$res = 10, a = 10, b = 2$$

$$\text{option A : } X^Y = a^b \implies 10^3 = 10^2 \quad \times$$

$$\text{option B : } (res * a)^Y = (res * X)^b \implies (10 * 10)^3 = (10 * 10)^2 \quad \times$$

$$\text{option C : } X^Y = res * a^b \implies 10^3 = 10 * 10^2 \quad \checkmark$$

$$\text{option D : } X^Y = (res * a)^b \implies 10^3 = (10 * 10)^2 \quad \times$$

Lets see one more iteration to verify option C.

Iteration 2:

$$res = 10, a = 10, b = 2$$

- $while(b! = 0) \implies 2! = 0 \quad \checkmark$
- $if(b \% 2 == 0) \implies 2 \% 2 == 0 \quad \checkmark$
 $a = a * a$
 $= 10 * 10 = 100$
 $b = \frac{b}{2}$
 $= \frac{2}{2} = 1$

Before Iteration 3:

$$res = 10, a = 100, b = 1$$

$$\text{Option C : } X^Y = res * a^b \implies 10^3 = 10 * 100^1 = 10^3 \quad \checkmark$$

Option C is answer

34 votes

-- Akash Kanase (42.5k points)

4.4.5 Loop Invariants: GATE2017-2-37 [top](#)<https://gateoverflow.in/118381>

Selected Answer

Here, $x == (y * q + r)$ says q = quotient and r = remainder.

To divide a number with repeated subtraction, quotient should be initialized to 0 and should be incremented for each subtraction.

Initially $q = 0 \Rightarrow r = x$.

\therefore Initial conditions should be C] ($q == 0$) && ($r == x$) && ($y > 0$) .

22 votes

-- Kantikumar (4.6k points)

4.4.6 Loop Invariants: TIFR2010-B-30 [top](#)<https://gateoverflow.in/19042>

Selected Answer

Whenever we encounter the [*], the variable s holds the sum of all elements $b[0]$ to $b[i - 1]$.

When we first enter the loop, $i = 0$, and s doesn't have any elements summed up.

When we last enter the loop, $i = (N - 1)$ and s contains the sum of elements $b[0]$ through $b[N - 2]$.

We leave the loop when $i = N$, and s gets the sum of elements $b[0]$ to $b[N - 1]$

The only option that matches this behavior is **option E**.

$$s = \sum_{j=0}^{i-1} b[j] \text{ & } 0 \leq i \leq N$$

13 votes

-- Pragy Agarwal (20.6k points)

4.4.7 Loop Invariants: TIFR2010-B-37 [top](#)<https://gateoverflow.in/19251>

Selected Answer

if

x is odd then -

- $z = a * b$ will be the o/p

if

x is even then -

case 1: if y is even then $x = 2 * x$ and $z = 0$ will be o/p

case 2: y is odd then loop will not terminate .

EDIT - When x and y are even, for some values of x , condition $z = ab$ is not holding.

For example - when $a = 4$ and $b = 4$, after loop terminates, $z = 8$ and $ab = 16$.

$8 \neq 16$.

So, **A** should be the correct answer.

4 votes

-- Saurav Shrivastava (1.6k points)

4.4.8 Loop Invariants: TIFR2017-B-5 top

<https://gateoverflow.in/95683>



Selected Answer

Consider the situations when loop gets broken -

1. If i becomes 10, or i becomes equal to y . \Rightarrow (i) is correct.
2. If $y > 10$, the control will come out of the loop when i becomes 10. \Rightarrow (ii) is correct.
3. During 1st iteration, $j = 1, i = 2$
 - o 2nd iteration, $j = 2, i = 3$
 - o 3rd iteration, $j = 6, i = 4$

4th iteration must not occur because j will become 24. Hence during 3rd iteration break statement must have been executed and hence, y must be equal to 4 (equal to i during the 3rd iteration).

This implies (iii) is also correct.

Hence, **Option D** is correct.

11 votes

-- tarun_svbk (1.5k points)

4.5

Parameter Passing(2) top

4.5.1 Parameter Passing: GATE2003-73 top

<https://gateoverflow.in/960>

The following program fragment is written in a programming language that allows global variables and does not allow nested declarations of functions.

```
global int i=100, j=5;
void P(x) {
    int i=10;
    print(x+10);
    i=200;
    j=20;
    print (x);
}
main() {P(i+j);}
```

If the programming language uses static scoping and call by need parameter passing mechanism, the values printed by the above program are:

- A. 115, 220
- B. 25, 220
- C. 25, 15
- D. 115, 105

[gate2003](#) [compiler-design](#) [normal](#) [runtime-environments](#) [parameter-passing](#)
[Answer](#)

4.5.2 Parameter Passing: GATE2013-42 [top](#)

<https://gateoverflow.in/60>

What is the return value of $f(p,p)$, if the value of p is initialized to 5 before the call? Note that the first parameter is passed by reference, whereas the second parameter is passed by value.

```
int f (int &x, int c) {
    c = c - 1;
    if (c==0) return 1;
    x = x + 1;
    return f(x,c) * x;
}
```

[gate2013](#) [compiler-design](#) [normal](#) [marks-to-all](#) [numerical-answers](#) [parameter-passing](#) [runtime-environments](#)
[Answer](#)

Answers: Parameter Passing

4.5.1 Parameter Passing: GATE2003-73 [top](#)

<https://gateoverflow.in/960>


Selected Answer

Call-by-name : Is a lazy evaluation (expression passed as an argument evaluated only when used) technique.

Call-by-need: Is a version of call-by-name but when an expression is evaluated during first use, is saved and reused for all later uses.

```
global int i=100, j=5;
//memory created for i and j variable and 100 and 5 stored in them respectively (1)

void P(x) { // p(i+j) (3)
    int i=10;
    // new variable created i with value 10 store in it. (4)

    print(x+10);
    // print(x+10); = print(i+j +10);= 10 + 5 +10 = 25
    // here need of i+j so i+j replaced by 15 everywhere .(5)

    i=200;
    // local i value changed to 200.

    j=20;
    //global j value changed to 20 as there is no local j.
    //if dynamic scoping used, then the scope of j comes from scope of j in main as
    //main called this function. Here, main also uses global j and hence no change.

    print (x);
    // print (x);= print (i+j); = printf(15) = 15 (7)
    // due to call by need. If call by name used answer is 200+20 = 220
}

main() {P(i+j);}
// 1st function call since it is call by need no evaluation done
//here i and j value refer to global values(2)
```

Answer is (C).

Reference : <https://www.cs.bgu.ac.il/~comp161/wiki.files/ps9.pdf>

23 votes

-- Prashant Singh (59.8k points)

4.5.2 Parameter Passing: GATE2013-42 [top](#)

<https://gateoverflow.in/60>



Selected Answer

In GATE 2013 marks were given to all as the same code in C/C++ produces undefined behavior. This is because * is not a sequence point in C/C++. The correct code must replace:

```
return f(x, c) * x;
with

res = f(x, c); // ';' forms a sequence point
//and all side-effects are guaranteed to be completed here
//-- updation of the x parameter inside f is guaranteed
//to be reflected in the caller from the next point onwards.
return res * x;
```

In this code, there will be 4 recursive calls with parameters (6,4), (7,3), (8,2) and (9,1). The last call returns 1. But due to pass by reference, *x* in all the previous functions is now 9. Hence, the value returned by *f(p,p)* will be $9 * 9 * 9 * 9 * 1 = 6561$.

Good Read:

- <http://stackoverflow.com/questions/41775973/is-this-undefined-behaviour-in-c-if-not-predict-the-output- logically>
- <https://gateoverflow.in/108445/c-programming?show=108582#a108582>

46 votes

-- Arjun Suresh (350k points)

4.6

Programming Constructs(1) [top](#)

4.6.1 Programming Constructs: GATE1999-2.5 [top](#)

<https://gateoverflow.in/1483>

Given the programming constructs

- assignment
- for loops where the loop parameter cannot be changed within the loop
- if-then-else
- forward go to
- arbitrary go to
- non-recursive procedure call
- recursive procedure/function call
- repeat loop,

which constructs will you not include in a programming language such that it should be possible to program the terminates (i.e., halting) function in the same programming language

- (ii), (iii), (iv)
- (v), (vii), (viii)
- (vi), (vii), (viii)
- (iii), (vii), (viii)

gate1999 | programming | normal | programming-constructs

Answer

Answers: Programming Constructs

4.6.1 Programming Constructs: GATE1999-2.5 top

<https://gateoverflow.in/1483>

Answer is (B).

Arbitrary goto, recursive call and repeat may enter infinite loop, and hence terminates program may not be able to answer if 'the program does terminate'.

15 votes

-- bahirNaik (3.2k points)

4.7

Programming In C(62) top

4.7.1 Programming In C: GATE2000-1.11 top

<https://gateoverflow.in/634>

The following C declarations:

```
struct node {
    int i;
    float j;
};
struct node *s[10];
```

define s to be:

- A. An array, each element of which is a pointer to a structure of type node
- B. A structure of 2 fields, each field being a pointer to an array of 10 elements
- C. A structure of 3 fields: an integer, a float, and an array of 10 elements
- D. An array, each element of which is a structure of type node

gate2000 | programming | programming-in-c | easy

Answer

4.7.2 Programming In C: GATE2000-1.12 top

<https://gateoverflow.in/635>

The most appropriate matching for the following pairs

X: m = malloc(5); m = NULL;	1: using dangling pointers
Y: free(n); n -> value = 5;	2: using uninitialized pointers
Z: char *p, *p = 'a';	3: lost memory

is:

- A. X - 1 Y - 3 Z - 2
- B. X - 2 Y - 1 Z - 3
- C. X - 3 Y - 2 Z - 1
- D. X - 3 Y - 1 Z - 2

gate2000 | programming | programming-in-c | normal

Answer

4.7.3 Programming In C: GATE2000-1.17, ISRO2015-79 [top](#)

Consider the following C declaration:

```
struct (
    short x[5];
    union {
        float y;
        long z;
    } u;
)t;
```

Assume that the objects of the type short, float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively. The memory requirement for variable *t*, ignoring alignment consideration, is:

- A. 22 bytes
- B. 14 bytes
- C. 18 bytes
- D. 10 bytes

[gate2000](#) [programming](#) [programming-in-c](#) [easy](#) [isro2015](#)

[Answer](#)

4.7.4 Programming In C: GATE2000-2.20 [top](#)

<https://gateoverflow.in/667>

The value of *j* at the end of the execution of the following C program:

```
int incr (int i)
{
    static int count = 0;
    count = count + i;
    return (count);
}
main () {
    int i, j;
    for (i = 0; i <= 4; i++)
        j = incr (i);
}
```

is:

- A. 10
- B. 4
- C. 6
- D. 7

[gate2000](#) [programming](#) [programming-in-c](#) [easy](#)

[Answer](#)

4.7.5 Programming In C: GATE2001-2.18 [top](#)

<https://gateoverflow.in/736>

Consider the following three C functions:

[P1]

```
int *g (void)
{
    int x = 10;
```

```

    return (&x);
}

```

[P2]

```

int *g(void)
{
    int *px;
    *px = 10;
    return px;
}

```

[P3]

```

int *g(void)
{
    int *px;
    px = (int*) malloc (sizeof(int));
    *px = 10;
    return px;
}

```

Which of the above three functions are likely to cause problems with pointers?

- A. Only P3
- B. Only P1 and P3
- C. Only P1 and P2
- D. P1, P2 and P3

gate2001 programming programming-in-c normal

Answer

4.7.6 Programming In C: GATE2002-1.17 [top](#)

<https://gateoverflow.in/822>

In the C language:

- A. At most one activation record exists between the current activation record and the activation record for the main
- B. The number of activation records between the current activation record and the activation records from the main depends on the actual function calling sequence.
- C. The visibility of global variables depends on the actual function calling sequence
- D. Recursion requires the activation record for the recursive function to be saved in a different stack before the recursive function can be called.

gate2002 programming programming-in-c easy descriptive

Answer

4.7.7 Programming In C: GATE2002-2.18 [top](#)

<https://gateoverflow.in/848>

The C language is:

- A. A context free language
- B. A context sensitive language
- C. A regular language
- D. Parsable fully only by a Turing machine

gate2002 programming programming-in-c normal

Answer

4.7.8 Programming In C: GATE2002-2.8 [top](#)

<https://gateoverflow.in/838>

Consider the following declaration of a two-dimensional array in C:

```
char a[100][100];
```

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0, the address of $a[40][50]$ is:

- A. 4040
- B. 4050
- C. 5040
- D. 5050

[gate2002](#) [programming-in-c](#) [programming](#) [easy](#)

[Answer](#)

4.7.9 Programming In C: GATE2003-2 [top](#)

<https://gateoverflow.in/893>

Assume the following C variable declaration:

```
int *A[10], B[10][10];
```

Of the following expressions:

- I. $A[2]$
- II. $A[2][3]$
- III. $B[1]$
- IV. $B[2][3]$

which will not give compile-time errors if used as left hand sides of assignment statements in a C program?

- A. I, II, and IV only
- B. II, III, and IV only
- C. II and IV only
- D. IV only

[gate2003](#) [programming](#) [programming-in-c](#) [easy](#)

[Answer](#)

4.7.10 Programming In C: GATE2003-89 [top](#)

<https://gateoverflow.in/972>

Consider the C program shown below:

```
#include<stdio.h>
#define print(x) printf("%d", x)

int x;
void Q(int z)
{
    z+=x;
    print(z);
}

void P(int *y)
{
```

```

        int x = *y + 2;
        Q(x);
        *y = x - 1;
        print(x);
    }
main(void) {
    x = 5;
    P(&x);
    print(x);
}

```

The output of this program is:

- A. 12 7 6
- B. 22 12 11
- C. 14 6 6
- D. 7 6 6

gate2003 programming programming-in-c normal

Answer

4.7.11 Programming In C: GATE2004-33 top

<https://gateoverflow.in/1030>

Consider the following C program segment:

```

char p[20];
char* s = "string";
int length = strlen(s);
for(i = 0; i < length; i++)
    p[i] = s[length-i];
printf("%s", p);

```

The output of the program is:

- A. gnirts
- B. string
- C. gnirt
- D. no output is printed

gate2004 programming programming-in-c easy

Answer

4.7.12 Programming In C: GATE2004-IT-58 top

<https://gateoverflow.in/3701>

Consider the following C program which is supposed to compute the transpose of a given 4×4 matrix M . Note that, there is an X in the program which indicates some missing statements. Choose the correct option to replace X in the program.

```

#include<stdio.h>
#define ROW 4
#define COL 4
int M[ROW][COL] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
main()
{
    int i, j, t;
    for (i = 0; i < 4; ++i)
    {
        X
    }
}

```

```

for (i = 0; i < 4; ++i)
    for (j = 0; j < 4; ++j)
        printf ("%d", M[i][j]);
}

```

A.

```

for(j = 0; j < 4; ++j) {
    t = M[i][j];
    M[i][j] = M[j][i];
    M[j][i] = t;
}

```

B.

```

for(j = 0; j < 4; ++j) {
    M[i][j] = t;
    t = M[j][i];
    M[j][i] = M[i][j];
}

```

C.

```

for(j = i; j < 4; ++j) {
    t = M[i][j];
    M[i][j] = M[j][i];
    M[j][i] = t;
}

```

D.

```

for(j = i; j < 4; ++j) {
    M[i][j] = t;
    t = M[j][i];
    M[j][i] = M[i][j];
}

```

gate2004-it programming easy programming-in-c

Answer

4.7.13 Programming In C: GATE2004-IT-59 [top](#)

<https://gateoverflow.in/3702>

What is the output of the following program?

```

#include<stdio.h>
int funcf (int x);
int funcg (int y);
main ()
{
    int x = 5, y = 10, count;
    for (count = 1; count <= 2; ++count) {
        y += funcf(x) + funcg(x);
        printf ("%d", y);
    }
}
funcf (int x) {
    int y;
    y = funcg(x);
    return (y);
}
funcg (int x) {
    static int y = 10;
    y += 1;
    return (y + x);
}

```

- A. 43 80
- B. 42 74
- C. 33 37

D. 32 32

gate2004-it programming programming-in-c normal

Answer

4.7.14 Programming In C: GATE2004-IT-60 [top](#)<https://gateoverflow.in/3703>

Choose the correct option to fill the ?1 and ?2 so that the program prints an input string in reverse order. Assume that the input string is terminated by a new line character.

```
#include <stdio.h>
void wrt_it (void);
int main (void)
{
    printf("Enter Text");
    printf ("\n");
    wrt_it();
    printf ("\n");
    return 0;
}
void wrt_it (void)
{
    int c;
    if (?1)
        wrt_it();
    ?2
}
```

- A. ?1 is `getchar() != '\n'`
?2 is `getchar(c);`
- B. ?1 is `(c = getchar()) != '\n'`
?2 is `getchar(c);`
- C. ?1 is `c != '\n'`
?2 is `putchar(c);`
- D. ?1 is `(c = getchar()) != '\n'`
?2 is `putchar(c);`

gate2004-it programming programming-in-c normal

Answer

4.7.15 Programming In C: GATE2004-IT-61 [top](#)<https://gateoverflow.in/3704>

Consider the following C program:

```
#include <stdio.h>
typedef struct {
    char *a;
    char *b;
} t;
void f1 (t s);
void f2 (t *p);
main()
{
    static t s = {"A", "B"};
    printf ("%s %s\n", s.a, s.b);
    f1(s);
    printf ("%s %s\n", s.a, s.b);
    f2(&s);
}
```

```

void f1 ( t s )
{
    s.a = "U";
    s.b = "V";
    printf ("%s %s\n", s.a, s.b);
    return;
}
void f2 ( t *p )
{
    p -> a = "V";
    p -> b = "W";
    printf ("%s %s\n", p -> a, p -> b);
    return;
}

```

What is the output generated by the program ?

- A. $\begin{matrix} A & B \\ U & V \\ V & W \\ V & W \end{matrix}$
- B. $\begin{matrix} A & B \\ U & V \\ A & B \\ V & W \end{matrix}$
- C. $\begin{matrix} A & B \\ U & V \\ U & V \\ V & W \end{matrix}$
- D. $\begin{matrix} A & B \\ U & V \\ V & W \\ U & V \end{matrix}$

gate2004-it programming programming-in-c normal

Answer

4.7.16 Programming In C: GATE2005-1, ISRO2017-55 [top](#)

<https://gateoverflow.in/1343>

What does the following C-statement declare?

int (*f) (int *);

- A. A function that takes an integer pointer as argument and returns an integer
- B. A function that takes an integer as argument and returns an integer pointer
- C. A pointer to a function that takes an integer pointer as argument and returns an integer
- D. A function that takes an integer pointer as argument and returns a function pointer

gate2005 programming programming-in-c easy isro2017

Answer

4.7.17 Programming In C: GATE2005-32 [top](#)

<https://gateoverflow.in/1368>

Consider the following C program:

```
double foo (double); /* Line 1 */
```

```

int main() {
    double da, db;
    //input da
    db = foo(da);
}
double foo (double a) {
    return a;
}

```

The above code compiled without any error or warning. If Line 1 is deleted, the above code will show:

- A. no compile warning or error
- B. some compiler-warnings not leading to unintended results
- C. some compiler-warnings due to type-mismatch eventually leading to unintended results
- D. compiler errors

[gate2005](#) [programming](#) [programming-in-c](#) [compiler-design](#) [easy](#)

Answer

4.7.18 Programming In C: GATE2005-IT-53 [top](#)

<https://gateoverflow.in/3814>

The following *C* function takes two ASCII strings and determines whether one is an anagram of the other. An anagram of a string *s* is a string obtained by permuting the letters in *s*.

```

int anagram (char *a, char *b) {
    int count [128], j;
    for (j = 0; j < 128; j++) count[j] = 0;
    j = 0;
    while (a[j] && b[j]) {
        A;
        B;
    }
    for (j = 0; j < 128; j++) if (count [j]) return 0;
    return 1;
}

```

Choose the correct alternative for statements *A* and *B*.

- A. *A* : $count[a[j]]++$ and *B* : $count[b[j]]--$
- B. *A* : $count[a[j]]++$ and *B* : $count[b[j]]++$
- C. *A* : $count[a[j+1]]++$ and *B* : $count[b[j]]--$
- D. *A* : $count[a[j]]++$ and *B* : $count[b[j+1]]--$

[gate2005-it](#) [programming](#) [normal](#) [programming-in-c](#)

Answer

4.7.19 Programming In C: GATE2005-IT-58 [top](#)

<https://gateoverflow.in/3819>

Let *a* be an array containing *n* integers in increasing order. The following algorithm determines whether there are two distinct numbers in the array whose difference is a specified number *S* > 0.

```

i = 0; j = 1;
while (j < n ) {
    if (E) j++;
    else if (a[j] - a[i] == S) break;
}

```

```

    else i++;
}
if (j < n) printf("yes") else printf ("no");

```

Choose the correct expression for E.

- A. $a[j] - a[i] > S$
- B. $a[j] - a[i] < S$
- C. $a[i] - a[j] < S$
- D. $a[i] - a[j] > S$

gate2005-it programming normal programming-in-c

Answer

4.7.20 Programming In C: GATE2006-57 [top](#)

<https://gateoverflow.in/1835>

Consider this C code to swap two integers and these five statements: the code

```

void swap (int *px, int *py)
{
    *px = *px - *py;
    *py = *px + *py;
    *px = *py - *px;
}

```

S1: will generate a compilation error

S2: may generate a segmentation fault at runtime depending on the arguments passed

S3: correctly implements the swap procedure for all input pointers referring to integers stored in memory locations accessible to the process

S4: implements the swap procedure correctly for some but not all valid input pointers

S5: may add or subtract integers and pointers

- A. S1
- B. S2 and S3
- C. S2 and S4
- D. S2 and S5

gate2006 programming programming-in-c normal

Answer

4.7.21 Programming In C: GATE2006-IT-49 [top](#)

<https://gateoverflow.in/3592>

Which one of the choices given below would be printed when the following program is executed ?

```

#include <stdio.h>
struct test {
    int i;
    char *c;
}st[] = {5, "become", 4, "better", 6, "jungle", 8, "ancestor", 7, "brother"};
main ()
{
    struct test *p = st;
    p += 1;
    ++p -> c;
    printf("%s,", p++ -> c);
}

```

```

printf("%c,", *++p -> c);
printf("%d,", p[0].i);
printf("%s \n", p -> c);
}

```

- A. jungle, n, 8, nclastor
- B. etter, u, 6, ungle
- C. cetter, k, 6, jungle
- D. etter, u, 8, ncestor

gate2006-it programming programming-in-c normal

Answer

4.7.22 Programming In C: GATE2006-IT-50 [top](#)

<https://gateoverflow.in/3593>

Which one of the choices given below would be printed when the following program is executed?

```

#include <stdio.h>
void swap (int *x, int *y)
{
    static int *temp;
    temp = x;
    x = y;
    y = temp;
}
void printab ()
{
    static int i, a = -3, b = -6;
    i = 0;
    while (i <= 4)
    {
        if ((i++)%2 == 1) continue;
        a = a + i;
        b = b + i;
    }
    swap (&a, &b);
    printf("a = %d, b = %d\n", a, b);
}
main()
{
    printab();
    printab();
}

```

- A. $a = 0, b = 3$
 $a = 0, b = 3$
- B. $a = 3, b = 0$
 $a = 12, b = 9$
- C. $a = 3, b = 6$
 $a = 3, b = 6$
- D. $a = 6, b = 3$
 $a = 15, b = 12$

gate2006-it programming programming-in-c normal

Answer

4.7.23 Programming In C: GATE2006-IT-51 [top](#)

<https://gateoverflow.in/3594>

Which one of the choices given below would be printed when the following program is executed?

```
#include <stdio.h>
int a1[] = {6, 7, 8, 18, 34, 67};
int a2[] = {23, 56, 28, 29};
int a3[] = {-12, 27, -31};
int *x[] = {a1, a2, a3};
void print(int *a[])
{
    printf("%d,", a[0][2]);
    printf("%d,", *a[2]);
    printf("%d,", *++a[0]);
    printf("%d,", *(++a)[0]);
    printf("%d\n", a[-1][+1]);
}
main()
{
    print(x);
}
```

- A. 8, -12, 7, 23, 8
- B. 8, 8, 7, 23, 7
- C. -12, -12, 27, -31, 23
- D. -12, -12, 27, -31, 56

gate2006-it programming programming-in-c normal

Answer

4.7.24 Programming In C: GATE2007-IT-31 [top](#)

<https://gateoverflow.in/3464>

Consider the C program given below :

```
#include <stdio.h>
int main ()
{
    int sum = 0, maxsum = 0, i, n = 6;
    int a [] = {2, -2, -1, 3, 4, 2};
    for (i = 0; i < n; i++)
    {
        if (i == 0 || a [i] < 0 || a [i] < a [i - 1])
        {
            if (sum > maxsum) maxsum = sum;
            sum = (a [i] > 0) ? a [i] : 0;
        }
        else sum += a [i];
    }
    if (sum > maxsum) maxsum = sum ;
    printf ("%d\n", maxsum);
}
```

What is the value printed out when this program is executed?

- A. 9
- B. 8
- C. 7
- D. 6

gate2007-it programming programming-in-c normal

Answer

4.7.25 Programming In C: GATE2008-18 [top](#)

<https://gateoverflow.in/416>

Which combination of the integer variables x, y and z makes the variable a get the value 4 in the following expression?

$$a = (x > y)?((x > z)?x : z) : ((y > z)?y : z)$$

- A. $x = 3, y = 4, z = 2$
- B. $x = 6, y = 5, z = 3$
- C. $x = 6, y = 3, z = 5$
- D. $x = 5, y = 4, z = 5$

gate2008 | programming | programming-in-c | easy

Answer

4.7.26 Programming In C: GATE2008-60 [top](#)

<https://gateoverflow.in/483>

What is printed by the following C program?

```
int f(int x, int *py, int **ppz)
{
    int y, z;
    **ppz += 1; z = **ppz; // corrected z = *ppz; to z = **ppz;
    *py += 2; y = *py;
    x += 3;
    return x+y+z;
}

void main()
{
    int c, *b, **a;
    c = 4; b = &c; a = &b;
    printf("%d", f(c, b, a));
}
```

- A. 18
- B. 19
- C. 21
- D. 22

gate2008 | programming | programming-in-c | normal

Answer

4.7.27 Programming In C: GATE2008-IT-49 [top](#)

<https://gateoverflow.in/3359>

What is the output printed by the following C code?

```
# include <stdio.h>
int main ()
{
    char a [6] = "world";
    int i, j;
    for (i = 0, j = 5; i < j; a [i++] = a [j--]);
    printf ("%s\n", a);
}
```

- A. dlrow
- B. Null string
- C. dlrlid
- D. worow

gate2008-it programming programming-in-c normal

Answer

4.7.28 Programming In C: GATE2008-IT-50 [top](#)

<https://gateoverflow.in/3360>

Consider the C program below. What does it print?

```
# include <stdio.h>
# define swap1 (a, b) tmp = a; a = b; b = tmp
void swap2 ( int a, int b)
{
    int tmp;
    tmp = a; a = b; b = tmp;
}
void swap3 (int*a, int*b)
{
    int tmp;
    tmp = *a; *a = *b; *b = tmp;
}
int main ()
{
    int num1 = 5, num2 = 4, tmp;
    if (num1 < num2) {swap1 (num1, num2);}
    if (num1 < num2) {swap2 (num1 + 1, num2);}
    if (num1 >= num2) {swap3 (&num1, &num2);}
    printf ("%d, %d", num1, num2);
}
```

- A. 5,5
- B. 5,4
- C. 4,5
- D. 4,4

gate2008-it programming programming-in-c normal

Answer

4.7.29 Programming In C: GATE2008-IT-51 [top](#)

<https://gateoverflow.in/3361>

Consider the C program given below. What does it print?

```
#include <stdio.h>
int main ()
{
    int i, j;
    int a [8] = {1, 2, 3, 4, 5, 6, 7, 8};
    for(i = 0; i < 3; i++) {
        a[i] = a[i] + 1;
        i++;
    }
    i--;
    for (j = 7; j > 4; j--) {
        int i = j/2;
        a[i] = a[i] - 1;
    }
    printf ("%d, %d", i, a[i]);
}
```

- A. 2,3
- B. 2,4

- C. 3,2
D. 3,3

gate2008-it programming programming-in-c normal

Answer

4.7.30 Programming In C: GATE2008-IT-52 [top](#)

<https://gateoverflow.in/3362>

C program is given below:

```
# include <stdio.h>
int main ()
{
    int i, j;
    char a [2] [3] = {{'a', 'b', 'c'}, {'d', 'e', 'f'}};
    char b [3] [2];
    char *p = *b;
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 3; j++) {
            *(p + 2*j + i) = a [i] [j];
        }
    }
}
```

What should be the contents of the array b at the end of the program?

- A. a b
c d
e f
- B. a d
b e
c f
- C. a c
e b
d f
- D. a e
d c
b f

gate2008-it programming programming-in-c normal

Answer

4.7.31 Programming In C: GATE2010-11 [top](#)

<https://gateoverflow.in/2184>

What does the following program print?

```
#include<stdio.h>

void f(int *p, int *q) {
    p=q;
    *p=2;
}

int i=0, j=1;

int main() {
```

```

    f(&i, &j);
    printf("%d %d\n", i, j);
    return 0;
}

```

- A. 2 2
- B. 2 1
- C. 0 1
- D. 0 2

gate2010 | programming | programming-in-c | easy

[Answer](#)

4.7.32 Programming In C: GATE2011-22 [top](#)

<https://gateoverflow.in/2124>

What does the following fragment of C program print?

```

char c[] = "GATE2011";
char *p = c;
printf("%s", p + p[3] - p[1]);

```

- A. GATE2011
- B. E2011
- C. 2011
- D. 011

gate2011 | programming | programming-in-c | normal

[Answer](#)

4.7.33 Programming In C: GATE2012-3 [top](#)

<https://gateoverflow.in/35>

What will be the output of the following C program segment?

```

char inChar = 'A';
switch ( inChar ) {
    case 'A' : printf ("Choice A \n");
    case 'B' :
    case 'C' : printf ("Choice B");
    case 'D' :
    case 'E' :
    default : printf ("No Choice");
}

```

- A. No Choice
- B. Choice A
- C. Choice A
Choice B No Choice
- D. Program gives no output as it is erroneous

gate2012 | programming | easy | programming-in-c

[Answer](#)

4.7.34 Programming In C: GATE2012-48 [top](#)

<https://gateoverflow.in/2176>

Consider the following C code segment.

```
int a, b, c = 0;
```

```
void prtFun(void);
main()
{
    static int a = 1;          /* Line 1 */
    prtFun();
    a += 1;
    prtFun();
    printf("\n %d %d ", a, b);
}

void prtFun(void)
{
    static int a = 2;          /* Line 2 */
    int b = 1;
    a += ++b;
    printf("\n %d %d ", a, b);
}
```

What output will be generated by the given code segment?

A.

3 1

4 1

4 2

B.

4 2

6 1

6 1

C.

4 2

6 2

2 0

D.

3 1

5 2

5 2

4.7.35 Programming In C: GATE2012-49 top

<https://gateoverflow.in/43314>

Consider the following C code segment.

```
int a, b, c = 0;
void prtFun(void);
main()
{
    static int a = 1;           /* Line 1 */
    prtFun();
    a += 1;
    prtFun();
    printf("\n %d %d ", a, b);
}

void prtFun(void)
{
    static int a = 2;           /* Line 2 */
    int b = 1;
    a += ++b;
    printf("\n %d %d ", a, b);
}
```

What output will be generated by the given code segment if:

Line 1 is replaced by **auto int**
a = 1;

Line 2 is replaced by **register int**
a = 2;

A.	
3	1
4	1
4	2
B.	
4	2
6	1
6	1
C.	
4	2

6	2
2	0
D.	
4	2
4	2
2	0

normal gate2012 programming-in-c programming

Answer

4.7.36 Programming In C: GATE2014-1-10 [top](#)

<https://gateoverflow.in/1770>

Consider the following program in C language:

```
#include <stdio.h>

main()
{
    int i;
    int*pi = &i;

    scanf("%d",pi);
    printf("%d\n", i+5);
}
```

Which one of the following statements is **TRUE**?

- A. Compilation fails.
- B. Execution results in a run-time error.
- C. On execution, the value printed is 5 more than the address of variable *i*.
- D. On execution, the value printed is 5 more than the integer value entered.

gate2014-1 programming programming-in-c easy

Answer

4.7.37 Programming In C: GATE2014-2-11 [top](#)

<https://gateoverflow.in/1965>

Suppose *n* and *p* are unsigned int variables in a C program. We wish to set *p* to nC_3 . If *n* is large, which one of the following statements is most likely to set *p* correctly?

- A. $p = n * (n - 1) * (n - 2)/6;$
- B. $p = n * (n - 1)/2 * (n - 2)/3;$
- C. $p = n * (n - 1)/3 * (n - 2)/2;$
- D. $p = n * (n - 1) * (n - 2)/6.0;$

gate2014-2 programming programming-in-c normal

Answer

4.7.38 Programming In C: GATE2014-2-42 [top](#)

<https://gateoverflow.in/2008>

Consider the C function given below.

```
int f(int j)
{
    static int i = 50;
    int k;
    if (i == j)
    {
        printf("something");
        k = f(i);
        return 0;
    }
    else return 0;
}
```

Which one of the following is **TRUE**?

- A. The function returns 0 for all values of j .
- B. The function prints the string **something** for all values of j .
- C. The function returns 0 when $j = 50$.
- D. The function will exhaust the runtime stack or run into an infinite loop when $j = 50$.

[gate2014-2](#) [programming](#) [programming-in-c](#)

Answer

4.7.39 Programming In C: GATE2015-1-11 [top](#)

<https://gateoverflow.in/8185>

The output of the following C program is_____.

```
void f1 ( int a, int b)  {
    int c;
    c = a; a = b;
    b = c;
}
void f2 ( int * a, int * b) {
    int c;
    c = * a; *a = *b; *b = c;
}
int main ()
{
    int a = 4, b = 5, c = 6;
    f1 ( a, b);
    f2 ( &b, &c);
    printf ("%d", c - a - b);
}
```

[gate2015-1](#) [programming](#) [programming-in-c](#) [easy](#) [numerical-answers](#)

Answer

4.7.40 Programming In C: GATE2015-1-35 [top](#)

<https://gateoverflow.in/8283>

What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires four bytes of memory.

```
int main ()
{
    unsigned int x [4] [3] =
        {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};
```

```
        printf ("%u, %u, %u", x + 3, *(x + 3), *(x + 2) + 3);
```

- A. 2036, 2036, 2036
- B. 2012, 4, 2204
- C. 2036, 10, 10
- D. 2012, 4, 6

gate2015-1 programming programming-in-c normal

[Answer](#)

4.7.41 Programming In C: GATE2015-2-15 [top](#)

<https://gateoverflow.in/8086>

Consider the following function written in the C programming language :

```
void foo(char *a)
{
    if (*a && *a != ' ')
    {
        foo(a+1);
        putchar(*a);
    }
}
```

The output of the above function on input "ABCD EFGH" is

- A. ABCD EFGH
- B. ABCD
- C. HGFE DCBA
- D. DCBA

gate2015-2 programming programming-in-c normal

[Answer](#)

4.7.42 Programming In C: GATE2015-3-26 [top](#)

<https://gateoverflow.in/8478>

Consider the following C program

```
#include<stdio.h>
int main()
{
    static int a[] = {10, 20, 30, 40, 50};
    static int *p[] = {a, a+3, a+4, a+1, a+2};
    int **ptr = p;
    ptr++;
    printf("%d%d", ptr-p, **ptr);
}
```

The output of the program is _____.

gate2015-3 programming programming-in-c normal numerical-answers

[Answer](#)

4.7.43 Programming In C: GATE2015-3-30 [top](#)

<https://gateoverflow.in/8486>

Consider the following two C code segments. Y and X are one and two dimensional arrays of size n and $n \times n$ respectively, where $2 \leq n \leq 10$. Assume that in both code segments, elements of Y are

initialized to 0 and each element $X[i][j]$ of array X is initialized to $i + j$. Further assume that when stored in main memory all elements of X are in same main memory page frame.

Code segment 1:

```
// initialize elements of Y to 0
// initialize elements of X[i][j] of X to i+j
for (i=0; i<n; i++)
    Y[i] += X[0][i];
```

Code segment 2:

```
// initialize elements of Y to 0
// initialize elements of X[i][j] of X to i+j
for (i=0; i<n; i++)
    Y[i] += X[i][0];
```

Which of the following statements is/are correct?

S1: Final contents of array Y will be same in both code segments

S2: Elements of array X accessed inside the for loop shown in code segment 1 are contiguous in main memory

S3: Elements of array X accessed inside the for loop shown in code segment 2 are contiguous in main memory

- A. Only S2 is correct
- B. Only S3 is correct
- C. Only S1 and S2 are correct
- D. Only S1 and S3 are correct

gate2015-3 programming-in-c normal

Answer

4.7.44 Programming In C: GATE2015-3-48 [top](#)

<https://gateoverflow.in/8557>

Consider the following C program:

```
#include<stdio.h>
int main()
{
    int i, j, k = 0;
    j=2 * 3 / 4 + 2.0 / 5 + 8 / 5;
    k=--j;
    for (i=0; i<5; i++)
    {
        switch(i+k)
        {
            case 1:
            case 2: printf("\n%d", i+k);
            case 3: printf("\n%d", i+k);
            default: printf("\n%d", i+k);
        }
    }
    return 0;
}
```

The number of times printf statement is executed is _____.

gate2015-3 programming programming-in-c normal numerical-answers

Answer**4.7.45 Programming In C: GATE2015-3-54** [top](#)<https://gateoverflow.in/8563>

Consider the following C program:

```
#include<stdio.h>
int f1(void);
int f2(void);
int f3(void);
int x=10;
int main()
{
    int x=1;
    x += f1() + f2() + f3() + f2();
    printf("%d", x);
    return 0;
}
int f1() { int x = 25; x++; return x; }
int f2() { static int x = 50; x++; return x; }
int f3() { x *= 10; return x; }
```

The output of the program is _____.

[gate2015-3](#) [programming](#) [programming-in-c](#) [normal](#) [numerical-answers](#)

Answer**4.7.46 Programming In C: GATE2015-3-7** [top](#)<https://gateoverflow.in/8401>

Consider the following C program segment.

```
# include <stdio.h>
int main()
{
    char s1[7] = "1234", *p;
    p = s1 + 2;
    *p = '0';
    printf("%s", s1);
}
```

What will be printed by the program?

- A. 12
- B. 120400
- C. 1204
- D. 1034

[gate2015-3](#) [programming](#) [programming-in-c](#) [normal](#)

Answer**4.7.47 Programming In C: GATE2016-1-12** [top](#)<https://gateoverflow.in/39638>

Consider the following "C" program.

```
void f(int, short);
void main()
{
    int i = 100;
    short s = 12;
```

```

    short *p = &s;
    _____; // call to f()
}

```

Which one of the following expressions , when placed in the blank above, will NOT result in a type checking error?

- A. $f(s, *s)$
- B. $i = f(i, s)$
- C. $f(i, *s)$
- D. $f(i, *p)$

gate2016-1 programming-in-c easy

Answer

4.7.48 Programming In C: GATE2016-1-15 [top](#)

<https://gateoverflow.in/39642>

Consider the following C program.

```

# include <stdio.h>
void mystery (int *ptr, int *ptrb) {
    int *temp;
    temp = ptrb;
    ptrb = ptr;
    ptr = temp;
}
int main () {
    int a = 2016, b=0, c= 4, d = 42;
    mystery (&a, &b);
    if (a < c)
        mystery (&c, &a);
    mystery (&a, &d);
    print f("%d\n", a);
}

```

The output of the program is _____.

gate2016-1 programming-in-c easy numerical-answers

Answer

4.7.49 Programming In C: GATE2016-1-34 [top](#)

<https://gateoverflow.in/39704>

The following function computes the maximum value contained in an integer array $P[]$ of size n ($n \geq 1$).

```

int max (int *p, int n) {
    int a = 0, b=n-1;

    while (_____) {
        if (p[a] <= p[b]) {a = a+1;}
        else {b = b-1;}
    }
    return p[a];
}

```

The missing loop condition is:

- A. $a! = n$
- B. $b! = 0$

- C. $b > (a + 1)$
 D. $b! = a$

gate2016-1 programming-in-c normal

Answer

4.7.50 Programming In C: GATE2016-2-12 top

<https://gateoverflow.in/39565>

The value printed by the following program is _____.

```
void f (int * p, int m) {
    m = m + 5;
    *p = *p + m;
    return;
}
void main () {
    int i=5, j=10;

    f (&i, j);
    print f ("%d", i+j);
}
```

gate2016-2 programming-in-c normal numerical-answers

Answer

4.7.51 Programming In C: GATE2016-2-37 top

<https://gateoverflow.in/39602>

Consider the following program:

```
int f (int * p, int n)
{
    if (n <= 1) return 0;
    else return max (f (p+1, n-1), p[0] - p[1]);
}
int main ()
{
    int a[] = {3, 5, 2, 6, 4};
    print f("%d", f(a, 5));
}
```

Note: $\max(x, y)$ returns the maximum of x and y .

The value printed by this program is _____.

gate2016-2 programming-in-c normal numerical-answers

Answer

4.7.52 Programming In C: GATE2017-1-13 top

<https://gateoverflow.in/118293>

Consider the following C code:

```
#include<stdio.h>
int *assignval (int *x, int val) {
    *x = val;
    return x;
}
void main () {
```

```

int *x = malloc(sizeof(int));
if (NULL == x) return;
x = assignval (x, 0);
if (x) {
    x = (int *)malloc(sizeof(int));
    if (NULL == x) return;
    x = assignval (x, 10);
}
printf("%d\n", *x);
free(x);
}

```

The code suffers from which one of the following problems:

- A. compiler error as the return of *malloc* is not typecast appropriately.
- B. compiler error because the comparison should be made as $x == NULL$ and not as shown.
- C. compiles successfully but execution may result in dangling pointer.
- D. compiles successfully but execution may result in memory leak.

gate2017-1 programming-in-c programming

Answer

4.7.53 Programming In C: GATE2017-1-36 [top](#)

<https://gateoverflow.in/118319>

Consider the C functions *foo* and *bar* given below:

```

int foo(int val) {
    int x=0;
    while(val > 0) {
        x = x + foo(val--);
    }
    return val;
}

```

```

int bar(int val) {
    int x = 0;
    while(val > 0) {
        x= x + bar(val-1);
    }
    return val;
}

```

Invocations of *foo(3)* and *bar(3)* will result in:

- A. Return of 6 and 6 respectively.
- B. Infinite loop and abnormal termination respectively.
- C. Abnormal termination and infinite loop respectively.
- D. Both terminating abnormally.

gate2017-1 programming-in-c programming normal

Answer

4.7.54 Programming In C: GATE2017-1-53 [top](#)

<https://gateoverflow.in/118473>

Consider the following C program.

```

#include<stdio.h>
#include<string.h>

```

```

void printlength(char *s, char *t) {
    unsigned int c=0;
    int len = ((strlen(s) - strlen(t)) > c) ? strlen(s) : strlen(t);
    printf("%d\n", len);
}

void main() {
    char *x = "abc";
    char *y = "defgh";
    printlength(x,y);
}

```

Recall that *strlen* is defined in *string.h* as returning a value of type *size_t*, which is an unsigned int. The output of the program is _____.

gate2017-1 programming programming-in-c normal numerical-answers

Answer

4.7.55 Programming In C: GATE2017-1-55 [top](#)

<https://gateoverflow.in/118442>

The output of executing the following C program is _____.

```

#include<stdio.h>

int total(int v) {
    static int count = 0;
    while(v) {
        count += v&1;
        v >>= 1;
    }
    return count;
}

void main() {
    static int x=0;
    int i=5;
    for(; i>0; i--) {
        x = x + total(i);
    }
    printf("%d\n", x);
}

```

gate2017-1 programming programming-in-c normal numerical-answers

Answer

4.7.56 Programming In C: GATE2017-2-14 [top](#)

<https://gateoverflow.in/118245>

Consider the following function implemented in C:

```

void printxy(int x, int y) {
    int *ptr;
    x=0;
    ptr=&x;
    y=*ptr;
    *ptr=1;
    printf("%d, %d", x, y);
}

```

The output of invoking *printxy(1,1)* is:

- A. 0,0
 B. 0,1
 C. 1,0
 D. 1,1

gate2017-2 programming-in-c

Answer**4.7.57 Programming In C: GATE2017-2-2** [top](#)<https://gateoverflow.in/118171>

Match the following:

P. static char var;	i. Sequence of memory locations to store addresses
Q. m=malloc(10); m=NULL;	ii. A variable located in data section of memory
R. char *ptr[10];	iii. Request to allocate a CPU register to store data
S. register int varl;	iv. A lost memory which cannot be freed

- A. P-ii; Q-iv; R-i; S-iii
 B. P-ii; Q-i; R-iv; S-iii
 C. P-ii; Q-iv; R-iii; S-i
 D. P-iii; Q-iv; R-i; S-ii

gate2017-2 programming programming-in-c

Answer**4.7.58 Programming In C: GATE2017-2-54** [top](#)<https://gateoverflow.in/118272>

Consider the following C program.

```
#include<stdio.h>
int main () {
    int m=10;
    int n, n1;
    n=++m;
    n1=m++;
    n--;
    --n1;
    n==n1;
    printf("%d", n);
    return 0;
}
```

The output of the program is _____

gate2017-2 programming-in-c numerical-answers

Answer**4.7.59 Programming In C: GATE2017-2-55** [top](#)<https://gateoverflow.in/118335>

Consider the following C program.

```
#include<stdio.h>
#include<string.h>
int main() {
    char* c="GATECSIT2017";
    char* p=c;
```

```

    printf("%d", (int)strlen(c+2[p]-6[p]-1));
    return 0;
}

```

The output of the program is _____

gate2017-2 programming-in-c numerical-answers

Answer

4.7.60 Programming In C: GATE2018-32 [top](#)

<https://gateoverflow.in/204106>

Consider the following C code. Assume that unsigned long int type length is 64 bits.

```

unsigned long int fun(unsigned long int n) {
    unsigned long int i, j, sum = 0;
    for( i=n; i>1; i=i/2) j++;
    for( ; j>1; j=j/2) sum++;
    return sum;
}

```

The value returned when we call fum with the input 2^{40} is:

- A. 4
- B. 5
- C. 6
- D. 40

gate2018 programming-in-c normal

Answer

4.7.61 Programming In C: GATE2018-45 [top](#)

<https://gateoverflow.in/204120>

Consider the following program written in pseudo-code. Assume that x and y are integers.

```

Count (x, y) {
    if (y != 1) {
        if (x != 1) {
            print("*");
            Count (x/2, y);
        }
        else {
            y=y-1;
            Count (1024, y);
        }
    }
}

```

The number of times that the *print* statement is executed by the call *Count(1024, 1024)* is _____

gate2018 programming-in-c numerical-answers

Answer

4.7.62 Programming In C: TIFR2018-A-7 [top](#)

<https://gateoverflow.in/179276>

```

Consider the following function definition.
void greet(int n)
{
    if (n>0)

```

```
{
    printf("hello");
    greet(n-1);
}
printf("world");
}
```

If you run `greet(n)` for some non-negative integer n , what would it print ?

- A. n times "hello", followed by $n+1$ times "world"
- B. n times "hello", followed by n times "world"
- C. n times "helloworld"
- D. $n+1$ times "helloworld"
- E. n times "helloworld", followed by "world"

[tifr2018](#) [programming-in-c](#)

[Answer](#)

Answers: Programming In C

4.7.1 Programming In C: GATE2000-1.11 [top](#)

<https://gateoverflow.in/634>



Selected Answer

(A) is the answer. `[]` has greater precedence than `*` in C. So, `s` becomes an array of pointers.

33 votes

-- gatecse (1.8k points)

4.7.2 Programming In C: GATE2000-1.12 [top](#)

<https://gateoverflow.in/635>



Selected Answer

Answer is (D).

$X : m = NULL$; makes the pointer m point to $NULL$. But the memory created using `malloc` is still there and cannot be used as we don't have a link to it. Hence, lost memory

$Y : n$ is freed and so pointer n is now pointing to an invalid memory making it a Dangling pointer.

$Z : p$ is not initialized. $p = malloc(sizeof(char))$; should have been used before assigning ' a ' to $*p$.

44 votes

-- Aditi Dan (5.2k points)

4.7.3 Programming In C: GATE2000-1.17, ISRO2015-79 [top](#)

<https://gateoverflow.in/640>



Selected Answer

Answer is (C).

Here, structure creates the memory for 'array and union', but union only creates the memory for only 'long z' which is the largest size data type inside it.

Hence,

$\text{short} \times [5] = 5 * 2 = 10$ bytes [shorts take 2 bytes]

$\text{long } z = 8$ bytes

So, $(10 + 8) = 18$ bytes

19 votes

-- Kalpana Bhargav (3.3k points)

4.7.4 Programming In C: GATE2000-2.20 top

<https://gateoverflow.in/667>



Answer: (A)

At $i = 0, j = 0$

At $i = 1, j = 1$

At $i = 2, j = 3$

At $i = 3, j = 6$

At $i = 4, j = 10$

19 votes

-- Rajarshi Sarkar (34.1k points)

4.7.5 Programming In C: GATE2001-2.18 top

<https://gateoverflow.in/736>



[P1] may cause an error because function is returning the address of locally declared variable.

[P2] will cause a problem because px is an int pointer that is not assigned with any address and we are doing dereferencing.

[P3] will work because memory in bytes of size of int will be reserved and its address will be stored in px that can be further use, once function execution completes, this m/m will still exist in Heap until we free it using *free()* function.

Hence, answer is (C).

48 votes

-- Manu Thakur (39.6k points)

4.7.6 Programming In C: GATE2002-1.17 top

<https://gateoverflow.in/822>



- Each function call starts a new activation record and since C allows nested function calls more than once activation record can exist between the current activation record and main.
- TRUE**
- Since, C uses static scoping, the actual function calling sequence has no impact on the visibility of global variables. If a variable is not found in the current activation record, it is looked in

- global address space and this is independent of the calling sequence.
- D. All function calls- whether recursive or not uses the same stack for saving the activation record. There is no need for a different stack as for C compiler a recursive function call and a normal function call make no difference.

47 votes

-- Arjun Suresh (350k points)

4.7.7 Programming In C: GATE2002-2.18 top

<https://gateoverflow.in/848>



Answer is (B).

All modern programming languages are CSL. Because they contain two features which cannot be handled by PDA.

The features are:

- variable declared before use and
- matching formal and actual parameters of functions.

26 votes

-- Sachin Mittal (15.7k points)

4.7.8 Programming In C: GATE2002-2.8 top

<https://gateoverflow.in/838>



Answer is (B).

In C, arrays are always stored in row-major form.

Formula to evaluate 2-D array's location is:----

$$loc(a[i][j]) = BA + [(i - lb_1) \times NC + (j - lb_2)] \times c$$

Where,

BA - Base Address

NC - no. of columns

c - memory size allocated to data type of array

$$a[lb_1 \dots ub_1][lb_2 \dots ub_2]$$

Here, $BA = 0, NC = 100, c = 1, a[0 \dots 99][0 \dots 99]$ so $lb_1 = 0, lb_2 = 0$

$$\begin{aligned} loc(a[40][50]) &= 0 + [(40 - 0) \times 100 + (50 - 0)] \times 1 \\ &= 0 + [4000 + 50] \times 1 = 4050 \end{aligned}$$

25 votes

-- Kalpana Bhargav (3.3k points)

4.7.9 Programming In C: GATE2003-2 top

<https://gateoverflow.in/893>



A is an array of pointers to int, and B is a 2-D array.

$A[2] =$
can take a pointer

$A[2][3] =$
can take an int

$B[1] =$
 $B[1]$ is the base address of array and it cannot be changed as array in C is a constant pointer.

$B[2][3] =$
can take an integer

So, (A) is the answer.

44 votes

-- Arjun Suresh (350k points)

4.7.10 Programming In C: GATE2003-89 top

<https://gateoverflow.in/972>



Selected Answer

```
main: x = 5; //Global x becomes 5
P: int x = *y + 2; //local x in P becomes 5+2 = 7
Q: z+=x; //local z in Q becomes 7 + 5 = 12
Q: print(z); //prints 12
P: *y = x - 1;
//content of address of local variable y
(same as global variable x) becomes 7 - 1 = 6
P: print(x); //prints local variable x in P = 7
main: print(x); //prints the global variable x = 6
```

26 votes

-- Arjun Suresh (350k points)

4.7.11 Programming In C: GATE2004-33 top

<https://gateoverflow.in/1030>



Selected Answer

Here,

$p[0] = s[length] = '\0';$ //compiler puts a '\0' at the end of all string literals

Now, for any string function in C, it checks till the first '\0' to identify the end of string. So, since the first char is '\0', printf %s, will print empty string. If we use printf("%s", p+1); we will get option (C) with some possible garbage until some memory location happens to contain "\0". For the given code, answer is (D).

26 votes

-- Arjun Suresh (350k points)

4.7.12 Programming In C: GATE2004-IT-58 top

<https://gateoverflow.in/3701>



Selected Answer

Option C:

look at the initial value of j , if j starts with 0, then double *for* loop will swap $M[i][j]$ with $M[j][i]$ and also $M[j][i]$ and $M[i][j]$ so the matrix M will remain unchanged, so to avoid this double swapping we need to initialize $j = i$ and swap only upper triangular matrix with lower triangular matrix.

```
for(j = i; j < 4; ++j) {
    // code for swapping M[i][j] with M[j][i]
    t = M[i][j];
    M[i][j] = M[j][i];
    M[j][i] = t;
}
```

29 votes

-- Vikrant Singh (13.5k points)

4.7.13 Programming In C: GATE2004-IT-59 [top](#)

<https://gateoverflow.in/3702>



funcf(x) + funcg(x)

funcf or *funcg* can be executed first. Lets assume *funcf* is executed first. It calls *funcg* - so even if the order of call is reversed, result will be same.

In first call of *funcg*, y becomes 11 and it returns $5 + 11 = 16$.

In second call of *funcg*, y becomes 12 and it returns $5 + 12 = 17$.

So, in main y is incremented by $16 + 17 = 33$ to become $10 + 33 = 43$. (Choice A)

In the second iteration y will be incremented by $18 + 19 = 37$ to give $43 + 37 = 80$.

30 votes

-- Arjun Suresh (350k points)

4.7.14 Programming In C: GATE2004-IT-60 [top](#)

<https://gateoverflow.in/3703>



getchar() - reads a single character at a time from the *stdin*.

putchar(c) - writes a character specified by the argument to *stdout*.

As

***getchar()* and *putchar()* both are needed to read the string and print its reverse and only option D contains both the function. D is the answer.** :P

Now coming to the code.

wrt_it(void) is calling itself recursively. when \n is encountered *putchar()* gets executed and prints the last character and then the function returns to its previous call and prints last 2nd character and so on.

8 votes

-- Soumya Jain (1.0k points)

4.7.15 Programming In C: GATE2004-IT-61 top

<https://gateoverflow.in/3704>



Selected Answer

First print $A\ B$

$f1$ is call by value the changes applicable only for local

from $f1\ U\ V$ is printed

back in main $A\ B$ is printed

then in $f2\ V\ W$ is printed

Hence, answer is (B).

1 21 votes

-- Sankaranarayanan P.N (11.5k points)

4.7.16 Programming In C: GATE2005-1, ISRO2017-55 top

<https://gateoverflow.in/1343>



Selected Answer

A. A function that takes an integer pointer as argument and returns an integer $\Rightarrow \text{int } f(\text{int}^*)$

B. A function that takes an integer as argument and returns an integer pointer $\Rightarrow \text{int } * f(\text{int})$

C. A pointer to a function that takes an integer pointer as argument and returns an integer \Rightarrow

```
int (*f) (int * );
```

So, answer is C.

1 30 votes

-- Akash Kanase (42.5k points)

4.7.17 Programming In C: GATE2005-32 top

<https://gateoverflow.in/1368>



Selected Answer

Answer is (D).

When a function is called without being defined, C compiler assumes it to return "int" but here foo is returning "double" and hence the compiler will throw type mis-match error.

From C99 on ward, implicit declaration of functions is not even allowed.

1 33 votes

-- Aditi Dan (5.2k points)

4.7.18 Programming In C: GATE2005-IT-53 top

<https://gateoverflow.in/3814>



Selected Answer

The answer is (D).

```

#include <stdio.h>

int main(void) {
    return 0;
}

int anagram (char *a, char *b) {
/*
ASCII characters are of 7-bits
so we use count array to represent all the ASCII characters
(ranging 0-127)
*/
int count [128], j;

/*
so this loop will initialize count of all the ASCII characters to be
0 (zero)
*/
for (j = 0; j < 128; j++) count[j] = 0;

j = 0;
/*
"a[j] && b[j]" ensures that anagram returns 0 (false) in case both
strings have different length. Because different length strings cannot
be anagram of each other
*/
/*
Logic:

Below while loop increments ASCII equivalent position for its occurrence
in array 'a' in count array; and decrements ASCII equivalent position
for its occurrence in array 'b' in count array.

Example: a = "ISS" and b = "SIS"
ASCII equivalent of:
I - 73
S - 83

j = 0: Statement A will increment count[ASCII of 'I'] ==> count[73]
count[73] = 0 --> 1
Statement B will decrement count[ASCII of 'S'] ==> count[83]
count[83] = 0 --> -1 and will increment j j = 0 --> 1

j = 1: Statement A will increment count[ASCII of 'S'] ==> count[83]
count[83] = -1 --> 0
Statement B will decrement count[ASCII of 'I'] ==> count[73]
count[73] = 1 --> 0 and will increment j j = 1 --> 2

j = 2: Statement A will increment count[ASCII of 'S'] ==> count[83]
count[83] = 0 --> 1
Statement B will decrement count[ASCII of 'S'] ==> count[83]
count[83] = 1 --> 0 and will increment j j = 2 --> 3

*** END OF LOOP ***

*/
while (a[j] && b[j]) {

A; //count [a[j]]++
/*

```

```
Note: j will be increment after count[]-- will execute
Resource: http://www.c4learn.com/c-programming/increment-operator-inside-printf
*/
B; //count[b[j++]]--
}

/*
This loop checks that the number of occurrences of the individual ASCII
characters is same or not.
If count[i] = 0 ---> same number of occurrences for ASCII character i
---> return 1 (true)

if count[i]!= 0 ---> different number of occurrences for ASCII character i
---> return 0 (false)
*/
for (j = 0; j < 128; j++) if (count [j]) return 0;
return 1;
}
```

32 votes

-- Sohil Ladhani (181 points)

4.7.19 Programming In C: GATE2005-IT-58 top

<https://gateoverflow.in/3819>

Selected Answer

Answer is (B)

For some ' i ' if we find that difference of $(A[j] - A[i] < S)$ we increment ' j ' to make this difference wider so that it becomes equal to S .

If at times difference becomes greater than S we know that it wont reduce further for same ' i ' and so we increment the ' i '.

We do it for each ' i ' if not found in previous iteration. until $i = n$

26 votes

-- Sandeep_Uniyal (7.5k points)

4.7.20 Programming In C: GATE2006-57 top

<https://gateoverflow.in/1835>

Selected Answer

S1 is false.

S2 is true, depending on the argument passed it may generate segmentation fault.

S3 is false because implementation is having some problem. Let $x = 3$ and I want to implement $SWAP[x,x]$. Now ans would be 0 but that must be x . Problem is because we are not checking whether both pointer are pointing the same address or different So, S4 is true.

S5 is obviously false so, option (**C**) is right.

44 votes

-- Kalpana Bhargav (3.3k points)

4.7.21 Programming In C: GATE2006-IT-49 top

<https://gateoverflow.in/3592>



code :

```
#include <stdio.h>

struct test {
    int i;
    char *c;
} st[] = {5, "become", 4, "better", 6, "jungle", 8, "ancestor", 7, "brother"};

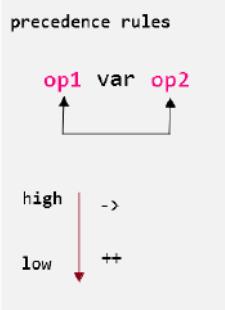
int main () {
//printf("size = %d\n", sizeof(struct test) );
    struct test *p = st;
    p += 1;
    ++p->c;           // ++(p->c)
    printf("%s,", p++->c); // (p++)->c
    printf("%c,", *++p->c); // *(++(p->c))
    printf("%d,", p[0].i);
    printf("%s \n", p->c);
}
```

We will assume few things:

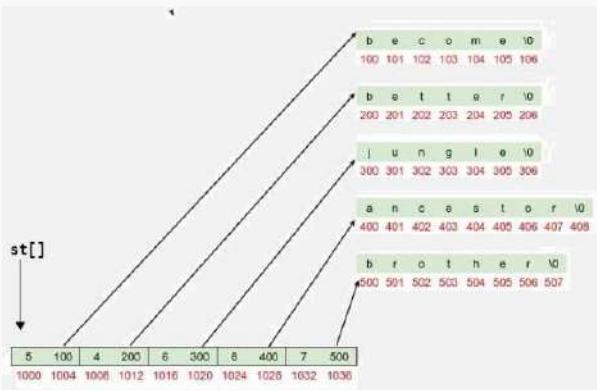
- Size of integer **4 Bytes**.
- Size of a pointer **4 Bytes**.

Neglecting any [alignment issues](#) with the storage of this structure we will have **8 Bytes per structure**.

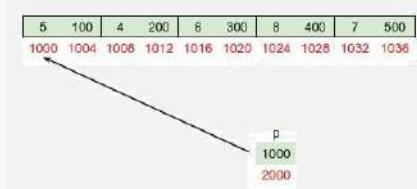
And one precedence rule we need to use:



Initial situation :

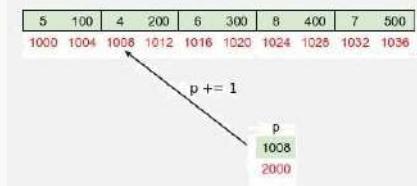


```
struct test *p = st;
```

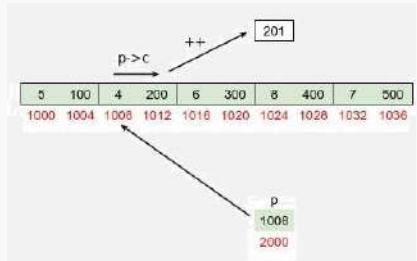


```
p += 1;
```

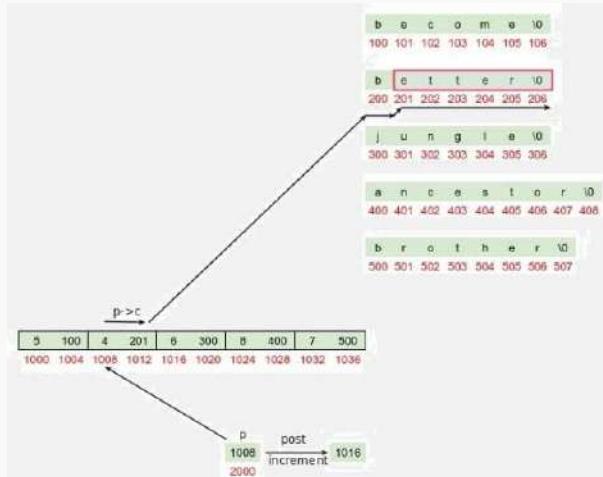
We know that if ptr is a pointer then, $\text{ptr} + x = \text{ptr} + x * \text{sizeof}(*\text{ptr})$;



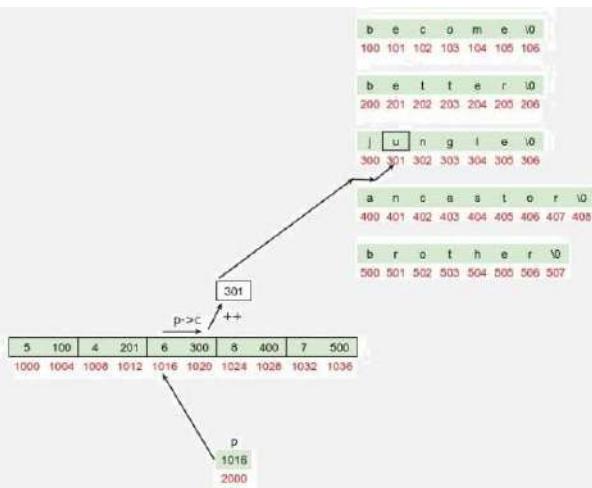
```
++p->c;
```



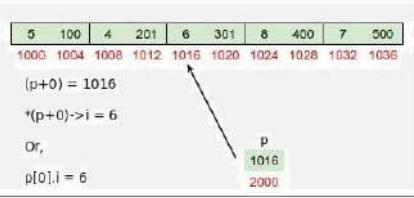
```
printf("%s, ", p++->c); // (p++)->c
```



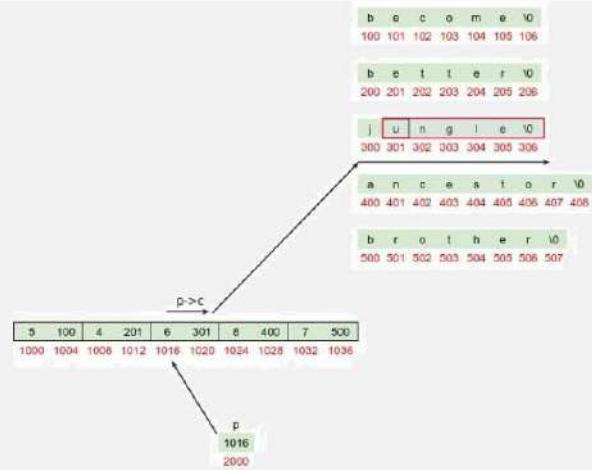
```
printf("%c, ", *++p->c); // * (++(p->c))
```



```
printf("%d, ", p[0].i);
```



```
printf("%s \n", p->c);
```



94 votes

-- Debashish Deka (56.7k points)

4.7.22 Programming In C: GATE2006-IT-50 top

<https://gateoverflow.in/3593>



Selected Answer

First of all, the swap function just swaps the pointers inside the function and has no effect on the variables being passed.

Inside printab, a and b are added odd integers from 1-5, i.e., $1 + 3 + 5 = 9$. So, in first call to printab, $a = -3 + 9 = 6$ and $b = -6 + 9 = 3$.

Static variables have one memory throughout program run (initialized during program start) and they keep their values across function calls. So, during second call to printab, $a = 6 + 9 = 15$,

$b = 3 + 9 = 12$.

Hence, (D) is choice.

Upvote 39 votes

-- Arjun Suresh (350k points)

4.7.23 Programming In C: GATE2006-IT-51 top

<https://gateoverflow.in/3594>



`a = {a1,a2,a3};`

`printf("%d, ", a[0][2]);`

$a[0]$ is $a1$. So, this will print $a1[2] = 8$;

`printf("%d, ", *a[2]);`

$a[2]$ is $a3$. So, this will print $*a3 = a3[0] = -12$ (has greater precedence than $*$)

`printf("%d, ", *++a[0]);`

$a[0]$ which is $a1$ is incremented. $a1$ is a pointer to int (base address of an integer array) and so increment means adding `sizeof(int)` and hence $a1$ now points to the second element in the array. So, $*++a[0]$ prints second element of $a1$ which is 7 and now $a1$ starts from 7.

`printf("%d, ", *(++a)[0]);`

$++a$ will increment a , which being a pointer (In C, an array when passed to a function becomes a pointer) to pointer (to int) will add `sizeof(pointer)` to a . So, a now contains $\{a2, a3\}$ and $a[0]$ will be $a2$ and $*a2$ will be the first element in $a2$ which is 23

`printf("%d\n", a[-1][+1]);`

$a[-1]$ will subtract a size of pointer from the base address of a . Normally this results in invalid memory access, but since we have incremented a previously, $a[-1]$ is valid and will point to $a1$. So, $a[-1][+1]$ will be $a1[1]$ which has the value 8.

($a1$ was incremented in 3rd printf and hence starts from 7 and not 6. $+1$ is same as 1, just given to create confusion)

Upvote 55 votes

-- Arjun Suresh (350k points)

4.7.24 Programming In C: GATE2007-IT-31 top

<https://gateoverflow.in/3464>



Answer is C.

I have tried to explain this question in a better way and the explanation is here...

i	$A[i]$	for → if - ----- satisfied?	maxsum	sum
-----	--------	-----------------------------------	--------	-----

-	-	-	0	0
0	$A[0] = 2$	Yes ($i == 0$)	0	2
1	$A[1] = -2$	Yes ($a[i] < 0$)	2	0
2	$A[2] = -1$	Yes ($a[i] < 0$) (for → if → if ----- not satisfied)	--	0
3	$A[3] = 3$	No (else executed) (for → if → if ----- not satisfied)	--	3
4	$A[4] = 4$	No (else executed) (for → if → if ----- not satisfied)	--	7
5	$A[5] = 2$	Yes ($a[i] < a[i - 1]$)	7	2

[End of for loop]

```
If (sum (i.e. 2) > maxsum(i.e. 7)) // No
    maxsum = sum; // Not Executed
printf will output maxsum = 7
```

19 votes

-- MKT (189 points)

4.7.25 Programming In C: GATE2008-18 top

<https://gateoverflow.in/416>



Selected Answer

Using option (A) : $x = 3, y = 4, z = 2$

$a = (3 > 4)? \text{No}$

therefore don't evaluate the first part and check second part $((y > z)?y : z)$

$(4 > 2)? \text{Yes}$

$a = \text{value of } y = 4$

Answer is (A)

$x = 3, y = 4, z = 2$

20 votes

-- Keith Kr (6.1k points)

4.7.26 Programming In C: GATE2008-60 top

<https://gateoverflow.in/483>



Selected Answer

c	4	5	7
b			
a			
x	4	7	
y	7		
z	5		

Return $x + y + z = \text{return } 7 + 7 + 5 = \text{return } 19$

So, **option B = 19** is correct.

25 votes

-- Amar Vashishth (30.5k points)

4.7.27 Programming In C: GATE2008-IT-49 top

<https://gateoverflow.in/3359>



Selected Answer

Char $a[6] =$

w	o	r	l	d	\0
---	---	---	---	---	----

$a[0]$ 1 2 3 4 5

After the loop will execute first time,

$a[0] = a[5]$

$a[0] = \text{'\0'}$

Next two more iterations of the loop till $i < j$ condition becomes false are not important for the output as the first position is '\0';

`printf("%s", a)`, the string starting at address a prints the string starting with '\0' and it indicates the end of string ,so it will print null string.

So, option (B).

25 votes

-- Mitali (201 points)

4.7.28 Programming In C: GATE2008-IT-50 [top](#)

<https://gateoverflow.in/3360>



Answer is C.

Only:

```
if (num1 > = num2) {swap3 (&num1, &num2);}
```

Statement works, which in turn swaps num1 and num2.

18 votes

-- Rajarshi Sarkar (34.1k points)

4.7.29 Programming In C: GATE2008-IT-51 [top](#)

<https://gateoverflow.in/3361>



Answer is (C) 3,2

First 2 variable integer type declared named *i* and *j*

Then int type array *a*[8] declared and initialized.

a[0] = 1, *a*[1] = 2, *a*[2] = 3, *a*[3] = 4, *a*[4] = 5, *a*[5] = 6, *a*[6] = 7, *a*[7] = 8

Then for loop started

i = 0, *i* < 3 (true)

a[0] = *a*[0] + 1 = 1 + 1 = 2

i ++ (outside for loop), *i* ++ (inside for loop);

i = 2, *i* < 3 (true)

a[2] = *a*[2] + 1 = 3 + 1 = 4

i ++, *i* ++ (outside for loop),

i = 4, *i* < 3 (false) //Now come out of loop

i --; (so *i* = 3)

Now another for loop started where in loop integer type variable named *i* declared

Block Scope: A Block is a set of statements enclosed within left and right braces ({ and }) respectively. Blocks may be nested in C (a block may contain other blocks inside it). A variable declared in a block is accessible in the block and all inner blocks of that block, but not accessible outside the block.

What if the inner block itself has one variable with the same name?

If an inner block declares a variable with the same name as the variable declared by the outer block, then the visibility of the outer block variable ends at the point of declaration by inner block.

So here inner block **int i** has the scope in this block only and outer block int i visibility is not allowed in that block

$j = 7, j > 4$ (true)

$\text{int } i = 7/2 = 3$

$a[3] = a[3] - 1 = 4 - 1 = 3$

$j = 6, j > 4$ (true)

$\text{int } i = 6/2 = 3$

$a[3] = a[3] - 1 = 3 - 1 = 2$

$j = 5, j > 4$ (true)

$\text{int } i = 5/2 = 2$

$a[2] = a[2] - 1 = 4 - 1 = 3$

$j = 4, j > 4$ (false)

Now when the for loop ends its variable named *i* scope is also end and the outer block variable now visible. So in printf outer variable *i* is used.

So, the output would be: 3, 2.

46 votes

-- Kalpana Bhargav (3.3k points)

4.7.30 Programming In C: GATE2008-IT-52 top

<https://gateoverflow.in/3362>



Selected Answer

The correct answer is option (B).

first integer type two variables declared *i* and *j*

then an integer type $2-d$ array $a[2][3]$ is declared and initialized and $2-d$ array $b[3][2]$ is created but not initialized. i.e

	address	value		address	value
$a[0][0]$	2000	<i>a</i>	$b[0][0]$	3000	garbage value
$a[0][1]$	2001	<i>b</i>	$b[0][1]$	3001	garbage value
$a[0][2]$	2002	<i>c</i>	$b[1][0]$	3002	garbage value
$a[1][0]$	2003	<i>d</i>	$b[1][1]$	3003	garbage value
$a[1][1]$	2004	<i>e</i>	$b[2][0]$	3004	garbage value
$a[1][2]$	2005	<i>f</i>	$b[2][1]$	3005	garbage value

now the char type pointer is declared and the base address of array *b* is put in it. so $p = 3000$

now the for loop is started where *i* is initialized to 0 ,so

```

i = 0 : i < 2 (true)
j = 0; j < 3 (true)
*(3000 + 2 * 0 + 0) = a[0][0] ⇒ *(3000) = a
j ++
j = 1; j < 3 (true)
*(3000 + 2 * 1 + 0) = a[0][1] ⇒ *(3002) = b
j ++
j = 2; j < 3 (true)
*(3000 + 2 * 2 + 0) = a[0][2] ⇒ *(3004) = c
j ++
j = 3; j < 3 (false)

i ++
i = 1 : i < 2 (true)
j = 0; j < 3 (true)
*(3000 + 2 * 0 + 1) = a[1][0] ⇒ *(3001) = d
j ++
j = 1; j < 3 (true)
*(3000 + 2 * 1 + 1) = a[1][1] ⇒ *(3003) = e
j ++
j = 2; j < 3 (true)
*(3000 + 2 * 2 + 1) = a[1][2] ⇒ *(3005) = f
j ++
j = 3; j < 3 (false)

i ++

```

now the values in array *b* is

<i>b[0][0]</i>	3000	<i>a</i>
<i>b[0][1]</i>	3001	<i>d</i>
<i>b[1][0]</i>	3002	<i>b</i>
<i>b[1][1]</i>	3003	<i>e</i>
<i>b[2][0]</i>	3004	<i>c</i>
<i>b[2][1]</i>	3005	<i>f</i>

Hence, the output will be (**B**) choice.

Note:

```
* (p + 2*j + i)
```

p + size of inner dimension $*j + i$, hence is same as $p[j][i]$. Hence with this statement we can identify that the code is transposing the matrix a and storing in b using pointer p .

Upvote 35 votes

-- Kalpana Bhargav (3.3k points)

4.7.31 Programming In C: GATE2010-11 top

<https://gateoverflow.in/2184>



Selected Answer

```
p=q; // now p and q are pointing to same address i.e. address of j
*p=2; // value of j will be updated to 2
```

Hence, answer is (D) 0 2

Upvote 28 votes

-- Manu Thakur (39.6k points)

4.7.32 Programming In C: GATE2011-22 top

<https://gateoverflow.in/2124>



Selected Answer

2011 is the answer.

In C, there is a rule that whatever character code be used by the compiler, codes of all alphabets and digits must be in order. So, if character code of '*A*' is x , then for '*B*' it must be $x + 1$.

Now %s means printf takes address and prints all bytes starting from that address as characters till any byte becomes the code for '\0'. Now, the passed value to printf here is $p + p[3] - p[1]$

p is the starting address of array c. $p[3] = 'E'$ and $p[1] = 'A'$. So, $p[3] - p[1] = 4$, and $p + 4$ will be pointing to the fifth position in the array c. So, printf starts printing from 2 and prints 2011.

(Here "GATE2011" is a string literal and by default a '\0' is added at the end of it by the compiler).

NB: In this question %s is not required.

```
printf(p + p[3] - p[1]);
```

Also gives the same result as first argument to printf is a character pointer and only if we want to pass more arguments we need to use a format string.

Upvote 47 votes

-- Arjun Suresh (350k points)

4.7.33 Programming In C: GATE2012-3 top

<https://gateoverflow.in/35>



Selected Answer

There is a `space` in between the `\'` and `n`. ([see-Q-no.-3](#))

```
case 'A' : printf ("Choice A\ n"); ^
```

So, [output](#) of the given program is:

```
Choice A nChoice BNo Choice
```

Which includes:

```
'n'
```

And there is no new line or spaces between outputs. Hence, there is no option matching.

<http://stackoverflow.com/questions/33694700/im-missing-something>

28 votes

-- Arjun Suresh (350k points)

4.7.34 Programming In C: GATE2012-48 [top](#)

<https://gateoverflow.in/2176>



main
 $a = 1$
prtFun()
 $a = 2$
 $b = 1$
 $a = a + ++b = 2 + 2 = 4$
 $b = 2$
 $\text{printf } \rightarrow 4\ 2$
back to main
 $a = a + 1 \rightarrow 1 + 1 \rightarrow 2$ (local static a is taken)
prtFun()
 $a = 4$ // previous value in the function is retained because of static
 $b = 1$
 $a = a + ++b = 4 + 2 = 6$
 $b = 2$
 $\text{printf } \rightarrow 6\ 2$
back to main
 $a = 2$
 $b = 0$ (initial value of global b. in *prtFun* local b is only updated)
 $\text{printf } \rightarrow 2\ 0$

Answer is **C**.

36 votes

-- Sankaranarayanan P.N (11.5k points)

4.7.35 Programming In C: GATE2012-49 [top](#)

<https://gateoverflow.in/43314>



```
main  

a = 1  

prtFun()
```

```

a = 2
b = 1
a = a + ++b = 2 + 2 = 4
b = 2
printf → 4 2
back to main
a = a + 1 → 1 + 1 → 2
prtFun()
a = 1 //previous a is lost
b = 1
a = a + ++b = 2 + 2 = 4
b = 2
printf → 4 2
back to main
a = 2b = 0 (initial value of global b. in prtFun local b is only updated)
printf → 2 0

```

Answer is **D.**

19 votes

-- Sankaranarayanan P.N (11.5k points)

4.7.36 Programming In C: GATE2014-1-10 top

<https://gateoverflow.in/1770>

Selected Answer

```

int i; //i is declared
int*pi = &i; //pi is a pointer variable
//and is assigned the address of i

scanf("%d",pi); //i is overwritten with the value
//we provided because pi is pointing to i earlier

printf("%d\n", i+5) //it will print the value stored in i+5

```

input=3; output=8

Option D is answer.

30 votes

-- Bhagirathi Nayak (14.1k points)

4.7.37 Programming In C: GATE2014-2-11 top

<https://gateoverflow.in/1965>

Selected Answer

Answer is (B).

In c, * and / have the same precedence and are left associative.

Evaluation of $n * (n - 1) * (n - 2)$ might exceed the unsigned int range.
 So, **(A)** and **(D)** are eliminated.
 $n * (n - 1)$ is always divisible by 2.(Gives an integer value). Where as it is not always divisible by 3.(You don't always get an integer..truncation possible, less accuracy)
(C) eliminated.

In option **(B)**
 $n * (n - 1)/2$ gives an integer value.

This integer value multiplied by $(n - 2)$ again gives an integer value.
Which when divided by 3 gives an integer value(Sets p correctly).

Reason : $n * (n - 1) * (n - 2)$ is the multiplication of 3 consecutive numbers. which is divisible by 2 as well as 3.
Hence, $(n * (n - 1) / 2 * (n - 2))$ is divisible by 3.

98 votes

-- Srinath Jayachandran (3.7k points)

4.7.38 Programming In C: GATE2014-2-42 top

<https://gateoverflow.in/2008>



Selected Answer

There is no updation for i and j in the function. so if we call function with $j = 50$ the recursive call will be continued infinitely. There is no terminating condition for recursion. hence answer D

24 votes

-- Sankaranarayanan P.N (11.5k points)

4.7.39 Programming In C: GATE2015-1-11 top

<https://gateoverflow.in/8185>



Selected Answer

```
void f1 ( int a, int b) {    //This code is call by value
// hence no effect of actual values when run.
    int c;
    c = a;
    a = b;
    b = c;
}
void f2 ( int * a, int * b) {    /*a= address of b
//and *b = address of c
    int c;                //int c = garbage
    c = * a;              //c = value at address a = 5;
    *a = *b;              /*a = Exchange original
// variable value of c to b = b= 6
    *b = c;               /*b = c = 5;
}
int main () {
    int a = 4, b = 5, c = 6;
    f1 ( a, b); This has no effect on actual values
// of a ,b since Call by value.
    f2 (&b, &c); Here change will be happen.
    At this point int a = 4, b = 6, c = 5;
    printf ("%d", c - a - b);      = (5-4)-6 = 1-6 = -5
}
```

3 votes

-- Anu007 (17k points)

Here, $f1$ will not change any values bcz it is call by value but $f2$ is call by reference and it swaps values of b and c and changes are also reflected in main function. So, $5 - 4 - 6 = -5$ is the answer.

30 votes

-- target gate (149 points)

4.7.40 Programming In C: GATE2015-1-35 top

<https://gateoverflow.in/8283>



Address of x is 2000.

x being a $2D$ array,

$$x + 3 = x + 3 * \text{sizeof its inner dimension}$$

$$= 2000 + 3 * 3 * 4 \text{ (as inner dimension is 3 integers of size 4)}$$

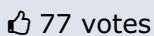
$$= 2000 + 36 = 2036.$$

$*(x + 3)$ returns the value at address 2036. But since x is $2 - D$ array, one $*$ will just return the $1D$ array which is the starting address of it, which is 2036 only.

$$(x + 2) = 2000 + 2 * 3 * 4 = 2024$$

$*(x + 2) + 3 = 2024 + 3 * 4 = 2036$ (The $*$ changes the data type from $2D$ to $1D$ and hence $+3$ will add $3 * 4$ and not $3 * 3 * 4$)

So, A.



77 votes

-- Arjun Suresh (350k points)

4.7.41 Programming In C: GATE2015-2-15 top

<https://gateoverflow.in/8086>



Answer D as priority of $!=$ is greater than that of $\&\&$ in C. The execution happens as:

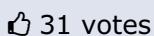
```
if (( *a) && (*a != ' '))
```

So, the if breaks either when $*a = 0$ (not '0' but ASCII 0 or null character '\0'), or when $*a = '$ '.

So, the recursive call goes like

'A' - 'B' - 'C' - 'D' - '' (breaks) and then starts outputting

DCBA



31 votes

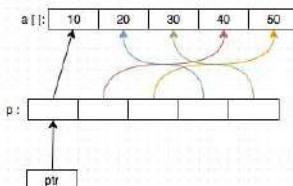
-- Vikrant Singh (13.5k points)

4.7.42 Programming In C: GATE2015-3-26 top

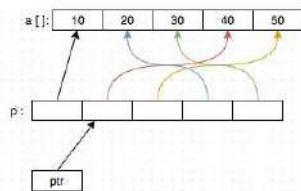
<https://gateoverflow.in/8478>



```
static int a[] = {10, 20, 30, 40, 50};
static int *p[] = {a, a+3, a+4, a+1, a+2};
int **ptr = p;
```



```
ptr++;
```



$$\text{ptr} - p = \frac{\text{address of } \text{ptr} - \text{address of } p}{\text{sizeof}(*\text{ptr})} = 1$$

$$**\text{ptr} = p[2] = *(\text{a} + 3) = 40$$

```
printf ("%d%d", ptr-p, **ptr); // 140
```

36 votes

-- Salman (963 points)

4.7.43 Programming In C: GATE2015-3-30 top

<https://gateoverflow.in/8486>



Selected Answer

Option is **C**. Only *S1* and *S2* are correct because *Y* have same element in both code and in code 1.

```
Y[i] += X[0][i];
```

This row major order (In C, arrays are stored in row-major order) which gives address of each element in sequential order ($1, 2, 3, \dots, n$) means we cross single element each time to move next shows contiguous in main memory but in code2 for:

```
Y[i] += X[i][0];
```

We are crossing n element (row crossing with n element) to move next.

26 votes

-- Anoop Sonkar (5k points)

4.7.44 Programming In C: GATE2015-3-48 top

<https://gateoverflow.in/8557>



Selected Answer

```
j=2 * 3 / 4 + 2.0 / 5 + 8 / 5;
```

$j = (((2 * 3) / 4) + (2.0 / 5)) + (8 / 5)$; //As associativity of +, * and / are from left to right and + has less precedence than * and /.

$j = ((6 / 4) + 0.4) + 1$; //2.0 is double value and hence 5 is implicitly typecast to double and we

get 0.4. But 8 and 5 are integers and hence $8/5$ gives 1 and not 1.6

$j = (1 + 0.4) + 1; // 6/4$ also gives 1 as both are integers

$j = 1.4 + 1; // 1 + 0.4$ gives 1.4 as 1 will be implicitly typecast to 1.4

$j = 2.4; //$ since j is integer when we assign 2.4 to it, it will be implicitly typecast to int.

So, $j = 2;$

$k = --j;$

This makes $j = 1$ and $k = -1$.

The variables j and k have values 1 and -1 respectively before the for loop. Inside the for loop, the variable i is initialized to 0 and the loop runs from 0 to 4.

$i = 0, k = -1, i + k = -1$, default case is executed, printf $count = 1$

$i = 1, k = -1, i + k = 0$, default case is executed, printf $count = 2$

$i = 2, k = -1, i + k = 1$, case 2, case 3 and default case is executed, printf $count = 5$

$i = 3, k = -1, i + k = 2$, case 2, case 3 and default case is executed, printf $count = 8$

$i = 4, k = -1, i + k = 3$, case 3 and default case is executed, printf $count = 10$

$i = 5$, loop exits and the control returns to main

Answer is **10**.

Upvote 58 votes

-- Shyam Singh (1.6k points)

4.7.45 Programming In C: GATE2015-3-54 top

<https://gateoverflow.in/8563>



Selected Answer

The variable x is initialized to 1. First and only call to $f1()$ returns 26. First call to $f2()$ returns 51. First and only call to $f3()$ returns 100. Second call to $f2()$ returns 52 (The value of local static variable x in $f2()$ retains its previous value 51 and is incremented by 1).

$$x = 1 + 26 + 51 + 100 + 52 = 230$$

Answer: 230

Upvote 38 votes

-- Shyam Singh (1.6k points)

4.7.46 Programming In C: GATE2015-3-7 top

<https://gateoverflow.in/8401>



Selected Answer

`p = s1 + 2;`

p now points to the third element in $s1$.

`*p = '0';`

The third element in s1 is made 0. So, 1234 becomes 1204. **C** choice.

40 votes

-- Arjun Suresh (350k points)

4.7.47 Programming In C: GATE2016-1-12 top

<https://gateoverflow.in/39638>



Selected Answer

- A. $f(s,*s)$ - 1st argument is short and not int, 2nd is type error(since s is a short variable). So, WRONG.
- B. $i = f(i,s)$ - return type is not void. So, WRONG.
- C. $f(i,*s)$ - 1st argument is int, second is again syntax error. So, WRONG
- D. $f(i,*p)$ - Both the arguments and return type match.. p is a pointer to short, so $*p$ is value of short. So, **D** is ANSWER.

46 votes

-- Abhilash Panicker (9.5k points)

4.7.48 Programming In C: GATE2016-1-15 top

<https://gateoverflow.in/39642>



Selected Answer

The mystery about mystery function is it does not affect values in main. As in C, parameters are passed by value- even if they are pointer. So, here the pointer values are exchanged within the function only. (we can use * operator to exchange the values at the location of the pointers and this will affect the values in main).

So, NO CHANGES in a,b,c,d .
And ANSWER is 2016

51 votes

-- Abhilash Panicker (9.5k points)

4.7.49 Programming In C: GATE2016-1-34 top

<https://gateoverflow.in/39704>



Selected Answer

Answer is (**D**).

Hint : Given in the question itself that we start comparing the contents of an array from $a[0]$ and $a[n-1]$ (converging from both side) then condition must be till both meet at a point and that point will be $a = b$.

Hence loop condition should be $a != b$.

Option C fails for $n = 2, p = [1,2]$.

31 votes

-- sukanyac (171 points)

4.7.50 Programming In C: GATE2016-2-12 top

<https://gateoverflow.in/39565>



Selected Answer

i is called by reference and j is called by value.

So, in function $f()$ only value of i might change,
 Now, in function $f(*p, m)$
 $*p$ is pointing to i
 thus $*p$ is 5.
 m is 10 because of call by value of j .

1. $m = 10 + 5$ hence $m = 15$
2. $*p = 5 + 15$ hence $*p = 20$, that is, value of variable i is now 20
3. returns nothing

Now, back to main
 $i = 20$ and j is as it is 10

Hence, output of printf will be $i + j = 20 + 10 = 30$

Check code in action ⇒ <http://codepad.org/mDLsWYp5>.

32 votes

-- Shashank Chavan (3.3k points)

4.7.51 Programming In C: GATE2016-2-37 top

<https://gateoverflow.in/39602>



Selected Answer

$f(a, 5)$

$p, n = 5$

3	5	2	6	4	
---	---	---	---	---	--

$\max(f(p+1, 5-1), 3-5)$ or $\max(f(p+1, 4), -2)$

$p, n = 4$

5	2	6	4	
---	---	---	---	--

$\max(\max(f(p+1, 4-1), 5-2), -2)$ or $\max(\max(f(p+1, 3), 3), -2)$

$p, n = 3$

2	6	4	
---	---	---	--

$\max(\max(\max(f(p+1, 3-1), 2-6), 3), -2)$ or
 $\max(\max(\max(f(p+1, 2), -4), 3), -2)$

$p, n = 2$

6	4	
---	---	--

$\max(\max(\max(\max(f(p+1), 1), 2), -4), 3), -2)$

$n = 1$, return 0

$\max(\max(\max(\max(0, 2), -4), 3), -2)$

$\max(\max(\max(2, -4), 3), -2)$

$\max(\max(2, 3), -2)$

$\max(3, -2)$

3

Check this code ⇒ <http://codepad.org/vVicmkN7>.

40 votes

-- Praveen Saini (54.6k points)

4.7.52 Programming In C: GATE2017-1-13 top

<https://gateoverflow.in/118293>



Selected Answer

Answer is D.

Option A: In C++ we need to do typecasting. C does automatic implicit typecasting.

See the screenshot for C & C++ compilers below. C compiler is working fine but C++ compiler is giving error.

Option B: Null means address 0. if ($a == 0$) if ($0 == a$) There is no difference.

Option C: Do it step by step, always x is pointing to a valid memory location. Dangling Pointer means if it points to a memory location which is deleted(freed). So no dangling pointer.

<http://www.geeksforgeeks.org/dangling-void-null-wild-pointers/>

Option D: $\$x\$$ will lose the previous address it was pointing to. So it will result in memory leak. <http://www.geeksforgeeks.org/what-is-memory-leak-how-can-we-avoid/>

Proof for **Option A:**

C Compiler:

```
Compile | Execute | Share Code main.c x
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main()
5 {
6     int *x=malloc(sizeof(int));
7     *x=1;
8     printf("%d\n",*x);
9     return 0;
10 }
```

DEV C++

```
Untitled1.cpp
1 #include <stdio.h>
2 #include<stdlib.h>
3
4 main()
5 {
6     int *x=malloc(sizeof(int));
7 }
```

sh-4.2\$ gcc -o main *.c
sh-4.2\$./main
1

[Error] invalid conversion from 'void*' to 'int*' [-fpermissive]

27 votes

-- Ahwan Mishra (10.5k points)

4.7.53 Programming In C: GATE2017-1-36 top

<https://gateoverflow.in/118319>



Selected Answer

Answer should be C.

Consider foo

Initially $val = 3$

$foo(val--)$

is equivalent to

1. $foo(val)$
2. $val = val - 1$

So,

1. *foo(3)*
2. *val = 2*

foo(3) calls *foo(3)* which in turn calls *foo(3)* this goes on

So, here we can see that *foo(3)* is called infinite number of times which causes memory overflow and abrupt termination

and one more thing to observe is infinite loop is not there since the val is decremented in the first iteration.

Consider bar.

Here, we can see the val is not decrementing in the loop

So,

1. *bar(3)* will call *bar(2)*
2. *bar(2)* will call *bar(1)*
3. *bar(1)* will call *bar(0)* → Here, *bar(0)* return 0
4. *bar(1)* will call *bar(0)*
5. *bar(1)* will call *bar(0)*

This will go on so here there is a problem of infinite loop but not abrupt termination since it does not cause memory overflow.

35 votes

-- (points)

4.7.54 Programming In C: GATE2017-1-53 top

<https://gateoverflow.in/118473>



Selected Answer

$$(strlen(s) - strlen(t)) = 3 - 5 = -2$$

But in C, when we do operation with two unsigned integers, result is also unsigned. (*strlen* returns *size_t* which is unsigned in most systems). So, this result " -2 " is treated as unsigned and its value is $INT_MAX - 2$ (not sure about all systems, but at least on systems using 2's complement representation). Now, the comparison is between this large number and another unsigned number *c* which is 0. So, the comparison return TRUE here.

Even if '*c*' is declared as "*int*", while doing an operation with an unsigned int, it gets promoted to unsigned int and we get the same result.

Hence $(strlen(s) - strlen(t)) > 0$ will return 1 on execution thus the conditional operator will return the true statement which is $strlen(abc) = 3$.

Ans should be 3.

46 votes

-- Arnabi Bej (7.9k points)

4.7.55 Programming In C: GATE2017-1-55 top

<https://gateoverflow.in/118442>



Selected Answer

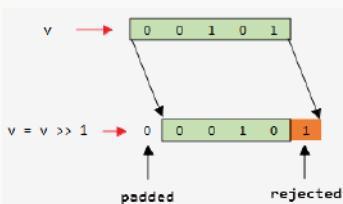
```
// inside total()
```

```
while (v) {
```

```

        count += v&1;    \\\ check the lowest bit of v
        v >>= 1;        \\\ or v = v >> 1 : right shift the bit pattern of v
    }
}

```



This piece of code will count the no of set bits in v .

In the main function, i values goes from 5 to 1, So there will be 5 calls to total().

Each call to total(i) counts the no of set bits in i . But the **count** is a static variable,

So, $\text{total}(i) = \text{Total no of set bits in all } i \leq 5$.

```

x = 0

i = 5 :: x = 0 + total(5); // x becomes = 2 :: 5 = 0101 // total(5) = 2
i = 4 :: x = 2 + total(4); // x becomes = 5 :: 4 = 0100 // total(4) = 1 + 2 = 3
i = 3 :: x = 5 + total(3); // x becomes = 10 :: 3 = 0011 // total(3) = 2 + 3 = 5
i = 2 :: x = 10 + total(2); // x becomes = 16 :: 2 = 0010 // total(2) = 1 + 5 = 6
i = 1 :: x = 16 + total(1); // x becomes = 23 :: 1 = 0001 // total(1) = 1 + 6 = 7

```

$$x = \sum_{i=1}^5 [\text{total}(i)] = 2 + 3 + 5 + 6 + 7 = 23$$

29 votes

-- Debasish Deka (56.7k points)

4.7.56 Programming In C: GATE2017-2-14 top

<https://gateoverflow.in/118245>



Selected Answer

At first in loop we are giving $x = 0$ then ptr is pointing to X .

So, $*ptr = 0$

Now, we copying the value of ptr to y , so $Y = 0$

```

x=0;          //value of x = 0
ptr= &x;        // ptr points to variable x
y= *ptr;      // Y contain value pointed by ptr i.e. x= 0;

```

Now, value of ptr is changed to 1. so the location of X itself got modified

```
*ptr=1;
```

As it is pointing to x so x will also be changed to 1

So, 1,0 will be the value

C is correct answer here.

24 votes

-- **Aboveallplayer** (18.4k points)

4.7.57 Programming In C: GATE2017-2-2 top

<https://gateoverflow.in/118171>



Selected Answer

static char var = A variable **located in data section** of memory

$m = malloc(10); m = null;$ **Here, free(m) is missing:** So, a lost memory which cannot be freed

$char * Ptr[10];$ Sequence of **10 memory location** to store addresses

register int var1; Request to allocate a **CPU register** to store data

Answer is **A**.

16 votes

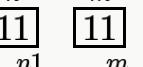
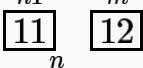
-- **Prashant Singh** (59.8k points)

4.7.58 Programming In C: GATE2017-2-54 top

<https://gateoverflow.in/118272>



Selected Answer

- $m = 10;$ 
 - $n = ++m;$ 
 - $n1 = m++;$ 
 - $n -;$ 
 - $-- n1;$ 
 - $n -= n1;$ 
- $(\because n = n - n1 = 10 - 10 = 0)$

So,

```
printf ("%d", n);
```

prints "0".

Hence, answer is **0**.

16 votes

-- **Arnabi Bej** (7.9k points)

4.7.59 Programming In C: GATE2017-2-55 top

<https://gateoverflow.in/118335>



Selected Answer

```
char c []="GATECSIT2017";
char *p=c;
printf ("%d",strlen(c+2 [p]-6 [p]-1));
```

$$2[p] = *(2+p) = p[2]$$

$$6[p] = *(6+p) = p[6]$$

$c + 2[p] - 6[p] - 1 = c + 'T' - 'I' - 1 = c + 11 - 1 = c + 10$ (In any character coding all alphabet letters are assigned consecutive int values as per C)

printf will print 2 which is the length of "17".

44 votes

-- Arjun Suresh (350k points)

4.7.60 Programming In C: GATE2018-32 top

<https://gateoverflow.in/204106>

Selected Answer

```
unsigned long int fun(unsigned long int n) {
    unsigned long int i, j, sum = 0;
    for( i=n; i>1; i=i/2) j++;
    for( ; j>1; j=j/2) sum++;
    return sum;
}
```

First for loop will make $j = 40$

Next for loop will divide j value (which is 40 now) by 2 each time unless $j \leq 1$.
 j loop starts:

```
j = 40 and sum =1,
j = 20 and sum=2,
j = 10 and sum=3,
j = 5 and sum=4,
j = 2 and sum=5,
j=1 break
```

So, Sum=5

14 votes

-- Digvijay (54.9k points)

4.7.61 Programming In C: GATE2018-45 top

<https://gateoverflow.in/204120>

Selected Answer

```
Count (x, y) {
    if (y != 1) {
        if (x != 1) {
            print("*");
            Count (x/2, y);
        }
        else {
            y=y-1;
            Count (1024, y);
        }
    }
}
```

Here, for each y value `print("*")` will run 10 times. Once x value reaches 1, `count(1024, y - 1)` will be called. Variable y can take values [2 1024] i.e. total 1023 values. So,

The total number of times '*' will be printed = $(\text{Number of times '*' printed per } y \text{ value}) * (\text{number of values } y \text{ takes})$.

Number of times '*' printed = $10 * 1023 = 10230$

8 votes

-- Digvijay (54.9k points)

4.7.62 Programming In C: TIFR2018-A-7 [top](#)

<https://gateoverflow.in/179276>



Selected Answer

Take $n = 2$:

<pre>void greet(int 1) if(2>0) rintf("hello"); print number 1 greet(1); printf("world"); 3rd time</pre>		
	<pre>void greet(int 1) if(1>0) printf("hello"); print number 2 greet(0); printf("world"); 2nd time print</pre>	
		<pre>void greet(int 0) if(0>0) fail printf("world"); 1st time print</pre>

Output = hellohelloworldworldworld i.e. n times hello and n+1 times world.

3 votes

-- Anu007 (17k points)

4.8

Programming Paradigms(2) [top](#)

4.8.1 Programming Paradigms: GATE2004-1 [top](#)

<https://gateoverflow.in/998>

The goal of structured programming is to:

- A. have well indented programs
- B. be able to infer the flow of control from the compiled code
- C. be able to infer the flow of control from the program text
- D. avoid the use of GOTO statements

[gate2004](#) [programming](#) [easy](#) [programming-paradigms](#)

Answer

4.8.2 Programming Paradigms: GATE2004-90 [top](#)

<https://gateoverflow.in/1084>

Choose the best matching between the programming styles in Group 1 and their characteristics in Group 2.

Group 1	Group 2
P. Functional	1. Common-based, procedural
Q. Logic	2. Imperative, abstract data types
R. Object-oriented	3. Side-effect free, declarative, expression evaluations
S. Imperative	4. Declarative, clausal representation, theorem proving

- A. P-2 Q-3 R-4 S-1
- B. P-4 Q-3 R-2 S-1
- C. P-3 Q-4 R-1 S-2
- D. P-3 Q-4 R-2 S-1

[gate2004](#) [programming](#) [normal](#) [programming-paradigms](#)

[Answer](#)

Answers: Programming Paradigms

4.8.1 Programming Paradigms: GATE2004-1 [top](#)

<https://gateoverflow.in/998>



Selected Answer

Answer is (**C**). The goal of structured programming is to able to infer the flow of control from the program text . It means user can execute the code according to his requirement. C and Pascal are good example of structured programming. In structured programming control passes one instruction to another instruction in sequential manner.

Avoiding the use of GOTO statements is not the goal of structured programming, it (avoiding the use of GOTO) is one of the requirements for a program to be structured.

22 votes

-- Kalpana Bhargav (3.3k points)

4.8.2 Programming Paradigms: GATE2004-90 [top](#)

<https://gateoverflow.in/1084>



Selected Answer

Answer: (D) P-3 Q-4 R-2 S-1

P. Functional	3. Side-effect free, declarative, expression evaluations
Q. Logic	4. Declarative, clausal representation, theorem proving
R. Object oriented	2. Imperative, abstract data types
S. Imperative	1. Common-based, procedural

Explanation:

P: Functional Programming is declarative in nature, involves expression evaluation, & side effect free.

Q: Logic is also declarative but involves theorem proving.

R: Object oriented is imperative statement based & have abstract (general) data types.

S: Imperative The programs are made giving commands & follows definite procedure & sequence

Ref: <https://www.geeksforgeeks.org/gate-gate-cs-2004-question-90/>

11 votes

-- Siddharth Mahapatra (1.3k points)

4.9

Pseudo Code(1) top

4.9.1 Pseudo Code: GATE2008-61 top

<https://gateoverflow.in/484>

Choose the correct option to fill ?1 and ?2 so that the program below prints an input string in reverse order. Assume that the input string is terminated by a new line character.

```
void reverse(void)
{
    int c;
    if(?1) reverse();
    ?2
}
main()
{
    printf("Enter text");
    printf("\n");
    reverse();
    printf("\n");
}
```

- A. ?1 is (*getchar()*) != '\n'
?2 is *getchar(c)*;
- B. ?1 is ((*c = getchar()*) != '\n')
?2 is *getchar(c)*;
- C. ?1 is (*c!* != '\n')
?2 is *putchar(c)*;
- D. ?1 is ((*c = getchar()*) != '\n')
?2 is *putchar(c)*;

[gate2008](#) [programming](#) [normal](#) [pseudo-code](#)

Answer

Answers: Pseudo Code

4.9.1 Pseudo Code: GATE2008-61 top

<https://gateoverflow.in/484>



Selected Answer

Here, we are using the '==' operator which has less priority than '!=>' operator. So *(c = getchar())* has to be in brackets and after reversing the string we use function *putchar(c)* for printing the character.

So, option (D) is the right answer.

21 votes

-- Kalpana Bhargav (3.3k points)

4.10

Recursion(18) top

4.10.1 Recursion: GATE1991-01,X top

<https://gateoverflow.in/507>

Consider the following recursive definition of *fib*:

```
fib(n) := if n = 0 then 1
          else if n = 1 then 1
          else fib(n-1) + fib(n-2)
```

The number of times *fib* is called (including the first call) for evaluation of *fib*(7) is _____.

gate1991 programming recursion normal

Answer

4.10.2 Recursion: GATE1994-21 top

<https://gateoverflow.in/2517>

Consider the following recursive function:

```
function fib (n:integer);integer;
begin
if (n=0) or (n=1) then fib := 1
else fib := fib(n-1) + fib(n-2)
end;
```

The above function is run on a computer with a stack of 64 bytes. Assuming that only return address and parameter are passed on the stack, and that an integer value and an address takes 2 bytes each, estimate the maximum value of *n* for which the stack will not overflow. Give reasons for your answer.

gate1994 programming recursion normal

Answer

4.10.3 Recursion: GATE1995-2.9 top

<https://gateoverflow.in/2621>

A language with string manipulation facilities uses the following operations

```
head(s): first character of a string
tail(s): all but exclude the first character of a string
```

```
concat(s1, s2): s1s2
```

For the string "acbc" what will be the output of

```
concat(head(s), head(tail(tail(s))))
```

- A. ac
- B. bc
- C. ab
- D. cc

gate1995 algorithms normal recursion

Answer**4.10.4 Recursion: GATE2000-16** [top](#)<https://gateoverflow.in/687>

A recursive program to compute Fibonacci numbers is shown below. Assume you are also given an array $f[0.....m]$ with all elements initialized to 0

```
fib(n) {
    if (n > M) error ();
    if (n == 0) return 1;
    if (n == 1) return 1;
    if (...)
        return _____ (1)
    t = fib(n - 1) + fib(n - 2);
    _____ (2)
    _____ (3)
    return t;
}
```

- A. Fill in the boxes with expressions/statements to make $fib()$ store and reuse computed Fibonacci values. Write the box number and the corresponding contents in your answer book.
- B. What is the time complexity of the resulting program when computing $fib(n)$?

[gate2000](#) [algorithms](#) [normal](#) [descriptive](#) [recursion](#)
Answer**4.10.5 Recursion: GATE2001-13** [top](#)<https://gateoverflow.in/754>

Consider the following C program:

```
void abc(char*s)
{
    if(s[0]=='\0') return;
    abc(s+1);
    abc(s+1);
    printf("%c", s[0]);
}

main()
{
    abc("123");
}
```

- A. What will be the output of the program?
- B. If $abc(s)$ is called with a null-terminated string s of length n characters (not counting the null ('\0') character), how many characters will be printed by $abc(s)$?

[gate2001](#) [programming](#) [recursion](#) [normal](#) [descriptive](#)
Answer**4.10.6 Recursion: GATE2002-11** [top](#)<https://gateoverflow.in/864>

The following recursive function in C is a solution to the Towers of Hanoi problem.

```
void move(int n, char A, char B, char C) {
    if (.....) {
        move (.....);
```

```

        printf("Move disk %d from pole %c to pole %c\n", n, A,C);
        move (.....);
    }
}

```

Fill in the dotted parts of the solution.

gate2002 programming recursion normal descriptive

Answer

4.10.7 Recursion: GATE2004-31, ISRO2008-40 [top](#)

<https://gateoverflow.in/1028>

Consider the following C function:

```

int f(int n)
{
    static int i = 1;
    if(n >= 5) return n;
    n = n+i;
    i++;
    return f(n);
}

```

The value returned by $f(1)$ is:

- A. 5
- B. 6
- C. 7
- D. 8

gate2004 programming programming-in-c recursion easy isro2008

Answer

4.10.8 Recursion: GATE2005-81a [top](#)

<https://gateoverflow.in/1403>

```

double foo(int n)
{
    int i;
    double sum;
    if(n == 0)
    {
        return 1.0;
    }
    else
    {
        sum = 0.0;
        for(i = 0; i < n; i++)
        {
            sum += foo(i);
        }
        return sum;
    }
}

```

The space complexity of the above code is?

- A. $O(1)$
- B. $O(n)$

- C. $O(n!)$
D. n^n

gate2005 programming recursion normal

Answer

4.10.9 Recursion: GATE2005-81b top

<https://gateoverflow.in/82146>

```
double foo(int n)
{
    int i;
    double sum;
    if(n == 0)
    {
        return 1.0;
    }
    else
    {
        sum = 0.0;
        for(i = 0; i < n; i++)
        {
            sum += foo(i);
        }
        return sum;
    }
}
```

Suppose we modify the above function $foo()$ and stores the value of $foo(i)$ $0 \leq i < n$, as and when they are computed. With this modification the time complexity for function $foo()$ is significantly reduced. The space complexity of the modified function would be:

- A. $O(1)$
B. $O(n)$
C. $O(n^2)$
D. $n!$

gate2005 programming recursion normal

Answer

4.10.10 Recursion: GATE2007-42 top

<https://gateoverflow.in/1240>

Consider the following C function:

```
int f(int n)
{
    static int r = 0;
    if (n <= 0) return 1;
    if (n > 3)
    {
        r = n;
        return f(n-2) + 2;
    }
    return f(n-1) + r;
}
```

What is the value of $f(5)$?

- A. 5
B. 7

- C. 9
D. 18

gate2007 programming recursion normal

Answer

4.10.11 Recursion: GATE2007-IT-27 [top](#)

<https://gateoverflow.in/3460>

The function f is defined as follows:

```
int f (int n) {
    if (n <= 1) return 1;
    else if (n % 2 == 0) return f(n/2);
    else return f(3n - 1);
}
```

Assuming that arbitrarily large integers can be passed as a parameter to the function, consider the following statements.

- i. The function f terminates for finitely many different values of $n \geq 1$.
- ii. The function f terminates for infinitely many different values of $n \geq 1$.
- iii. The function f does not terminate for finitely many different values of $n \geq 1$.
- iv. The function f does not terminate for infinitely many different values of $n \geq 1$.

Which one of the following options is true of the above?

- A. i and iii
B. i and iv
C. ii and iii
D. ii and iv

gate2007-it programming recursion normal

Answer

4.10.12 Recursion: GATE2014-2-40 [top](#)

<https://gateoverflow.in/2000>

Consider the following function.

```
double f(double x) {
    if (abs(x*x - 3) < 0.01)
        return x;
    else
        return f(x/2 + 1.5/x);
}
```

Give a value q (to 2 decimals) such that $f(q)$ will return q :_____.

gate2014-2 programming recursion numerical-answers normal

Answer

4.10.13 Recursion: GATE2016-1-35 [top](#)

<https://gateoverflow.in/39730>

What will be the output of the following C program?

```
void count (int n) {
    static int d=1;
```

```

printf ("%d", n);
printf ("%d", d);
d++;
if (n>1) count (n-1);
printf ("%d", d);

}

void main(){
    count (3);
}

```

- A. 3 1 2 2 1 3 4 4 4
 B. 3 1 2 1 1 1 2 2 2
 C. 3 1 2 2 1 3 4
 D. 3 1 2 1 1 1 2

gate2016-1 programming-in-c recursion normal

Answer

4.10.14 Recursion: GATE2017-1-35 [top](#)

<https://gateoverflow.in/118317>

Consider the following two functions.

```

void fun1(int n) {
    if(n == 0) return;
    printf("%d", n);
    fun2(n - 2);
    printf("%d", n);
}

void fun2(int n) {
    if(n == 0) return;
    printf("%d", n);
    fun1(++n);
    printf("%d", n);
}

```

The output printed when `fun1(5)` is called is

- A. 53423122233445
 B. 53423120112233
 C. 53423122132435
 D. 53423120213243

gate2017-1 programming normal tricky recursion

Answer

4.10.15 Recursion: GATE2018-21 [top](#)

<https://gateoverflow.in/204095>

Consider the following C program:

```

#include<stdio.h>

int counter=0;

int calc (int a, int b) {
    int c;
    counter++;
    if(b==3) return (a*a*a);
}

```

```

        else {
            c = calc(a, b/3);
            return (c*c*c);
        }
    }

int main() {
    calc(4, 81);
    printf("%d", counter);
}

```

The output of this program is _____.

gate2018 programming-in-c numerical-answers recursion

Answer

4.10.16 Recursion: ISICAL MTech 2014 CS [top](#)

<https://gateoverflow.in/27885>

How many asterisks (*) in terms of k will be printed by the following C function, when called as $\text{count}(m)$ where $m = 3^k$? Justify your answer.

Assume that 4 bytes are used to store an integer in C and k is such that 3^k can be stored in 4 bytes.

```

void count(int n) {
    printf("*");

    if (n>1) {
        count(n/3);
        count(n/3);
        count(n/3);
    }
}

```

programming-in-c recursion isi-2014

Answer

4.10.17 Recursion: TIFR2010-B-31 [top](#)

<https://gateoverflow.in/26484>

Consider the following computation rules. **Parallel-outermost rule:** Replace all the outermost occurrences of F (i.e., all occurrences of F which do not occur as arguments of other F 's) simultaneously. **Parallel - innermost rule:** Replace all the innermost occurrences of F (i.e., all occurrences of F with all arguments free of F 's) simultaneously. Now consider the evaluations of the recursive program over the integers.

```

F(x, y) <== if x = 0 then 0 else
              [ F(x + 1, F(x, y)) * F(x - 1, F(x, y)) ]

```

where the multiplication functions $*$ is extended as follows:

```

0 * w & w * 0 are 0
a * w & w * a are w (for any non-zero integer a)
w * w is w

```

We say that $F(x, y) = w$ when the evaluation of $F(x, y)$ does not terminate. Computing $F(1, 0)$ using the parallel - innermost and parallel - outermost rule yields

- A. w and 0 respectively
- B. 0 and 0 respectively
- C. w and w respectively

- D. w and 1 respectively
 E. none of the above

[tifr2010](#) [programming](#) [recursion](#)

[Answer](#)

4.10.18 Recursion: TIFR2011-B-38 [top](#)

<https://gateoverflow.in/20923>

Consider the class of recursive and iterative programs. Which of the following is false?

- A. Recursive programs are more powerful than iterative programs.
 B. For every iterative program there is an equivalent recursive program.
 C. Recursive programs require dynamic memory management.
 D. Recursive programs do not terminate sometimes.
 E. Iterative programs and recursive programs are equally expressive.

[tifr2011](#) [programming-in-c](#) [recursion](#)

[Answer](#)

Answers: Recursion

4.10.1 Recursion: GATE1991-01,X [top](#)

<https://gateoverflow.in/507>



Selected Answer

The recurrence relation for the no. of calls is

$$T(n) = T(n-1) + T(n-2) + 2$$

$T(0) = T(1) = 0$ (for $\text{fib}(0)$ and $\text{fib}(1)$, there are no recursive calls).

$$T(2) = 2$$

$$T(3) = 4$$

$$T(4) = 8$$

$$T(5) = 14$$

$$T(6) = 24$$

$$T(7) = 40.$$

Counting the initial call we get $40 + 1 = 41$.

28 votes

-- Arjun Suresh (350k points)

4.10.2 Recursion: GATE1994-21 [top](#)

<https://gateoverflow.in/2517>



Selected Answer

Size of an activation record = $2 + 2 = 4$ bytes.

So, no. of possible activation records which can be live at a time = $64/4 = 16$.

So, we can have 16 function calls live at a time (recursion depth = 16), meaning the maximum value for n without stack overflow is 16 (calls from 1 – 16). For $n = 17$, stack will overflow.

This is different from the total no. of recursive calls which will be as follows:

n	No. of calls
1	1
2	3
3	5
4	9
5	15
6	25

拇指图标 27 votes

-- Arjun Suresh (350k points)

4.10.3 Recursion: GATE1995-2.9 top

<https://gateoverflow.in/2621>



Selected Answer

Answer is C.

concat(a, head(tail(tail(acbc))))

concat(a, head(tail(cbc)))

concat(a, head(bc))

concat(a, b)

ab.

拇指图标 17 votes

-- Gate Keeda (19.6k points)

4.10.4 Recursion: GATE2000-16 top

<https://gateoverflow.in/687>



Selected Answer

Array f is used to store the $fib()$ values calculated in order to save repeated calls. Since $n = 0$ and $n = 1$ are special cases we can store $fib(2)$ to $f[0]$, $fib(3)$ to $f[1]$ and so on. The missing code completed would be:

```
if (f[n - 2] != 0) {
    return f[n-2];
}
t = fib(n-1) + fib(n-2);
f[n-2] = t;
return t;
```

In this code, $fib(i)$ will do a recursion only once as once $fib(i)$ is calculated it is stored in array.

So, the time complexity for $fib(n)$ would be $\Theta(n)$.

PS: We can also store $fib(n)$ in $f(n)$, the above code just saves 2 elements' space in the array.

23 votes

-- Arjun Suresh (350k points)

4.10.5 Recursion: GATE2001-13 top

<https://gateoverflow.in/754>



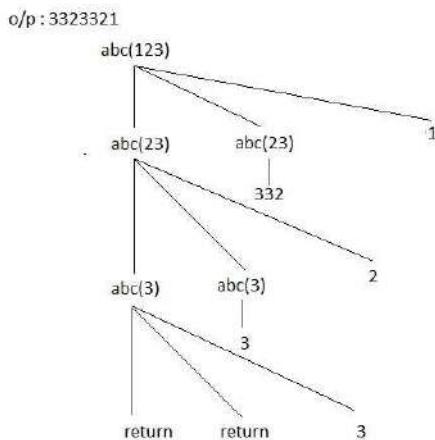
Selected Answer

Answer is a) 332 332 1

Answer b) $2^n - 1$

Q. a) O/p :

3 3 2 3 3 2 1



$$\text{b)} : T(n) = 2T(n-1) + 1; n > 0$$

$$= 0; n = 0 \text{ [Since for length zero string no character will be printed]}$$

After solving it by substitution,

$$\begin{aligned} T(n) &= 2T(n-1) + 1 \\ &= 2(2T(n-2) + 1) + 1 \\ &= 2^2T(n-2) + 2 + 1 \\ &= 2^2(2T(n-3) + 1) + 2 + 1 \\ &= 2^3T(n-3) + 2^2 + 2 + 1 \end{aligned}$$

Finally, it will expand like

$$\begin{aligned} &= 2^nT(n-n) + 2^{n-1} + 2^{n-2} + \dots + 2^2 + 2 + 1 \\ &= 2^nT(0) + 2^{n-1} + 2^{n-2} + \dots + 2^2 + 2 + 1 \\ &= \frac{1 \cdot (2^n - 1)}{(2-1)} \\ &= 2^n - 1 \end{aligned}$$

24 votes

-- jayendra (8.2k points)

4.10.6 Recursion: GATE2002-11 top

<https://gateoverflow.in/864>



Selected Answer

```
void move(int n, char A, char B, char C) {
    if (n > 0) {
        move (n-1, A, C, B);
        printf("Move disk %d from pole %c to pole %c\n", n, A, C);
        move (n-1, B, A, C);
    }
}
```

16 votes

-- sonam vyas (14.8k points)

4.10.7 Recursion: GATE2004-31, ISRO2008-40 top

<https://gateoverflow.in/1028>



Selected Answer

Answer is 7. As,

$f(1) : n = 2, i = 2$

$f(2) : n = 4, i = 3$

$f(4) : n = 7, i = 4$

$f(7) : print(n) \Rightarrow 7 <\text{ans}>$

26 votes

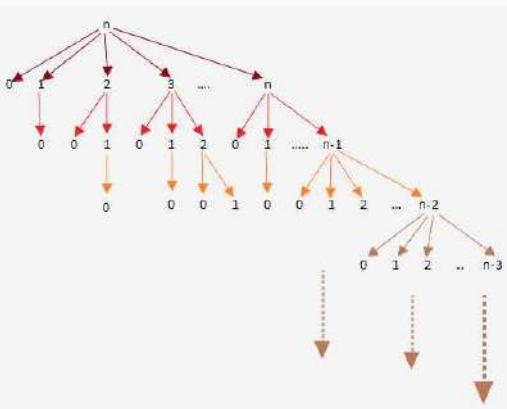
-- sumit kumar singh dixit (2.1k points)

4.10.8 Recursion: GATE2005-81a top

<https://gateoverflow.in/1403>



Selected Answer



- A. The code here is storing only local variables. So, the space complexity will be the recursion depth- maximum happening for the last iteration of the for loop $-foo(n-1) - foo(n-2) - \dots - foo(0)$ all live giving space complexity $O(n)$.

- B. To store the n values we need space complexity $O(n)$. So, the space complexity won't change and will be $O(n)$.

34 votes

-- Arjun Suresh (350k points)

4.10.9 Recursion: GATE2005-81b top

<https://gateoverflow.in/82146>



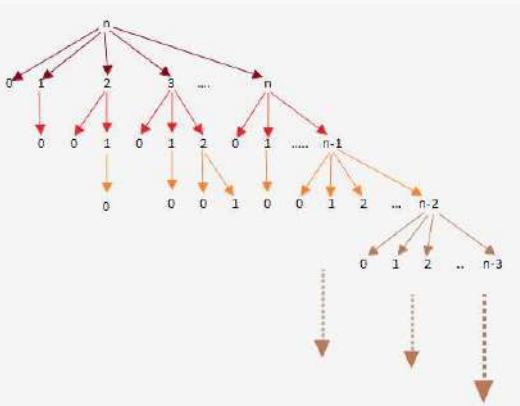
Selected Answer

Given program:

```
#include <stdio.h>
double foo(int n) {
    int i;
    double sum;
    if(n == 0) {
        return 1.0;
    }
    else {
        sum = 0.0;
        for(i = 0; i < n; i++) {
            sum += foo(i);
        }
        return sum;
    }
}

int main() {
    double a = foo(5);
    printf("%.2f\n", a);
}
```

And here is the present situation :



Here we can see that, we have lots of overlapping subproblems. Too many function calls.

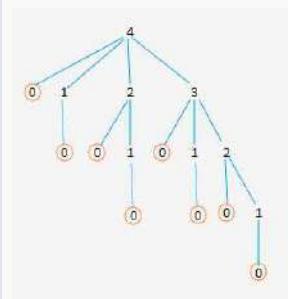
1. No of function calls = 2^n
2. stack depth used = $O(n)$

Therefore space is linear and time is exponential.

If we take a small number say 4, then we would have 8.0 as answer, or we can see that $foo(n) = 2^{n-1}$

and

1. stack depth used = 5.
2. No of function calls = $2^4 = 16$.



```

    printf ("%lf\n", a);
    free (memo_table);
}

```

Improvements over given program:

1. Stack space because of function calls reduced to **two level** only.
2. Extra space used for the **$1D$** array = $O(n)$
3. More importantly, **Time** is reduced to $O(n^2)$. (**Exponential to Quadratic !!**)

Overall space complexity = stack + extra = $O(1) + O(n) = O(n)$

Answer is **B**.

15 votes

-- Debashish Deka (56.7k points)

4.10.10 Recursion: GATE2007-42 top

<https://gateoverflow.in/1240>



Selected Answer

The answer is **D**.

$$f(5) = 18.$$

$$f(3) + 2 = 16 + 2 = 18$$

$$f(2) + 5 = 11 + 5 = 16$$

$$f(1) + 5 = 6 + 5 = 11$$

$$f(0) + 5 = 1 + 5 = 6$$

Consider from last to first. Since it is recursive function.

22 votes

-- Gate Keeda (19.6k points)

4.10.11 Recursion: GATE2007-IT-27 top

<https://gateoverflow.in/3460>



Selected Answer

The function terminates for all powers of **2** (which is infinite), hence (i) is false and (ii) is TRUE.

Let $n = 5$.

Now, recursive calls will go like $5 - 14 - 7 - 20 - 10 - 5 -$

And this goes into infinite recursion. And if we multiply **5** with any power of **2**, also result will be infinite recursion. Since, there are infinite powers of **2** possible, there are infinite recursions possible (even considering this case only). So, (iv) is TRUE and (iii) is false.

So, correct answer is **(D)**.

34 votes

-- Arjun Suresh (350k points)

4.10.12 Recursion: GATE2014-2-40 top

<https://gateoverflow.in/2000>



(We can directly go to the "if" part to get one answer, but we need to solve "else" part too to get all possible answers which though is not asked in question)

Solving the else part:

$$\frac{x}{2} + \frac{3}{2x} = \frac{x^2+3}{2x}$$

So, the new value of x will be $\frac{x^2+3}{2x}$ and we need it equal to x .

$$\frac{x^2+3}{2x} = x \implies x^2 + 3 = 2x^2 \implies x^2 = 3 \implies x = 1.732$$

Now solving the if part.

```
abs(x*x - 3) < 0.01
```

So,
 $x^2 - 3 < 0.01$ and $-(x^2 - 3) < 0.01 \implies x^2 < 3.01$ and $x^2 > 2.99 \implies x < 1.735$ and $x > 1.735$

Corrected to 2 decimal places answer should be 1.73 or 1.74.

43 votes

-- Arjun Suresh (350k points)

4.10.13 Recursion: GATE2016-1-35 top

<https://gateoverflow.in/39730>



Here, **D** is **Static**, so the value of **D** will persists between the function calls.

1. $count(3)$ will print the value of n and d and increments d and call $count(2) = prints\ 3\ 1$.
2. $count(2)$ will print the value of n and d and increments d and call $count(1) = prints\ 2\ 2$.
3. $count(1)$ will print the value of n and d and increments $d \Rightarrow prints\ 1\ 3$.

Now, it will return and prints the final incremented value of d which is
4, 3 times.

So, option (A) is correct = 3 1 2 2 1 3 4 4 4

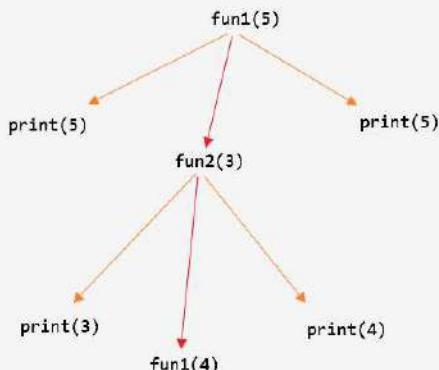
31 votes

-- Monanshi Jain (9.3k points)

4.10.14 Recursion: GATE2017-1-35 top

<https://gateoverflow.in/118317>





- Unroll recursion up to a point where we can distinguish the given options and choose the correct one!
- Options **B** and **D** are eliminated.
- **A** is the answer.

31 votes

-- Debarshish Deka (56.7k points)

4.10.15 Recursion: GATE2018-21 top

<https://gateoverflow.in/204095>



Selected Answer

```
int main() {
    calc(4, 81);
    printf("%d", counter);
}
```

printf("%d", counter);
So we need only counter value.

Each function increments counter value by 1. Goal is to find the number of function call sequence of function calls:

`calc(4, 81) ---> calc(4, 27) ---> calc(4, 9) ---> calc(4, 3) ---> return`

4 function calls.

counter = 4

12 votes

-- Digvijay (54.9k points)

4.10.16 Recursion: ISICAL MTech 2014 CS top

<https://gateoverflow.in/27885>



Selected Answer

Recurrence relation goes like this:

$$T(n) = 3T\left(\frac{n}{3}\right) + 1 \quad \rightarrow (1)$$

Using substitution,

$$T(n) = 3 \left(3T\left(\frac{\frac{n}{3}}{3}\right) + 1 \right) + 1$$

$$\implies T(n) = 3^2 T\left(\frac{n}{3^2}\right) + 3 + 1 \quad \rightarrow (2)$$

Use substitution again,

$$T(n) = 3^2 \left(3T\left(\frac{n}{3^3}\right) + 1 \right) + 3 + 1$$

$$\implies T(n) = 3^3 T\left(\frac{n}{3^3}\right) + 3^2 + 3 + 1$$

$$\implies T(n) = 3^3 T\left(\frac{n}{3^3}\right) + 3^2 + 3^1 + 3^0 \quad \rightarrow (3)$$

Continuing like this up to ' k ' times we get,

$$T(n) = 3^k T\left(\frac{n}{3^k}\right) + 3^{k-1} + 3^{k-2} + \dots + 3^1 + 3^0 \quad \rightarrow (4)$$

put $n = 3^k$ which is what question says i.e., $n = 3^k$ (I am considering ' n ' in place of ' m ' as name doesn't matter here)

$$T(n) = 3^k + 3^{k-1} + 3^{k-2} + 3^{k-3} + \dots + 3^1 + 3^0$$

Solving above G.P. series with $a = 1, r = 3$ we get

$$T(n) = \frac{3^{k+1} - 1}{3 - 1} = \frac{3^{k+1} - 1}{2}$$

4 votes

-- Arjun (177 points)

4.10.17 Recursion: TIFR2010-B-31 top

<https://gateoverflow.in/26484>

Answer is **A**) w and 0

If we solve using parallel innermost rule

$$\begin{aligned} F(1,0) &= F(2, F(1,0)) * F(0, F(1,0)) \\ &= F(2, F(2, F(1,0))) * F(0, F(2, F(1,0))) * F(0, F(1,0)) \end{aligned}$$

Since computation of $F(1,0)$ goes on

We assign $F(1,0)$ to w .

So, $F(1,0) = w$

Using parallel outermost rule:

$$\begin{aligned} F(1,0) &= F(2,F(1,0)) * F(0,F(1,0)) \\ &= F(2,F(1,0)) * 0 \\ &= 0 \end{aligned}$$

thumb up 2 votes

-- zambus (279 points)

4.10.18 Recursion: TIFR2011-B-38 top

<https://gateoverflow.in/20923>



Answer is **option E**.

Computable function: those which can be incorporated in a program using for/while loops.

Total Function: Defined for all possible inputs

Well Defined: if its definition assigns it a unique value.

It was a belief in early 1900s that every Computable function was also Primitively Recursive. But the discovery of Ackermann function provided a counter to it.

The class of primitive recursive functions is a small subclass of the class of recursive functions. This means that there are some functions which are Well-Defined Total Functions and are Computable BUT **Not** primitively recursive; eg. Ackermann function.

This makes all options from option A to option D as True.

But **option E** as FALSE. As iterative programs are equivalent to only Primitively Recursive class.

thumb up 12 votes

-- Amar Vashishth (30.5k points)

4.11

Structures(1) top

4.11.1 Structures: GATE2018-2 top

<https://gateoverflow.in/204076>

Consider the following C program:

```
#include<stdio.h>
struct Ournode{
    char x, y, z;
};
int main() {
    struct Ournode p={'1', '0', 'a'+2};
    struct Ournode *q=&p;
    printf("%c, %c", *((char*)q+1), *((char*)q+2));
    return 0;
}
```

The output of this program is:

- A. 0, c
- B. 0, a+2
- C. '0', 'a+2'
- D. '0', 'c'

gate2018 programming-in-c programming structures pointers normal

[Answer](#)

Answers: Structures

4.11.1 Structures: GATE2018-2 [top](#)

<https://gateoverflow.in/204076>



char x = 'a' + 2
i.e. x == 'c'

So, p = {'1', '0', 'c'}

((char)q + 1) == p[1]
((char)q + 2) == p[2]

printf("%c, %c", * ((char*) q+1), * ((char*) q+2));

Output: 0, c

14 votes

-- Digvijay (54.9k points)

4.12

Type Checking(1) [top](#)

4.12.1 Type Checking: GATE2003-24 [top](#)

<https://gateoverflow.in/914>

Which of the following statements is FALSE?

- A. In statically typed languages, each variable in a program has a fixed type
- B. In un-typed languages, values do not have any types
- C. In dynamically typed languages, variables have no types
- D. In all statically typed languages, each variable in a program is associated with values of only a single type during the execution of the program

gate2003 programming normal type-checking

[Answer](#)

Answers: Type Checking

4.12.1 Type Checking: GATE2003-24 [top](#)

<https://gateoverflow.in/914>



Answer is (C). In dynamically typed languages variables do have type- just that it is inferred during runtime.

14 votes

-- Arjun Suresh (350k points)

4.13

Variable Binding(1) top

4.13.1 Variable Binding: GATE2007-IT-34, UGCNET-Dec2012-III-52 top

<https://gateoverflow.in/3467>

Consider the program below in a hypothetical programming language which allows global variables and a choice of static or dynamic scoping.

```
int i ;
program main ()
{
    i = 10;
    call f();
}

procedure f()
{
    int i = 20;
    call g();
}
procedure g()
{
    print i;
}
```

Let x be the value printed under static scoping and y be the value printed under dynamic scoping. Then, x and y are:

- A. $x = 10, y = 20$
- B. $x = 20, y = 10$
- C. $x = 10, y = 10$
- D. $x = 20, y = 20$

[gate2007-it](#) [programming](#) [variable-binding](#) [normal](#) [ugcnetdec2012iii](#)
Answer

Answers: Variable Binding

4.13.1 Variable Binding: GATE2007-IT-34, UGCNET-Dec2012-III-52 top

<https://gateoverflow.in/3467>

Selected Answer

In static scoping, the scope of an identifier is determined by its location in the code, and since that doesn't change, the scope doesn't either. In dynamic scoping, the scope is determined by the sequence of calls that has led to the use of an identifier, and since that can be different each time that use is reached, is dynamic.

So, here:

Option A must be the answer.

As,

under static scoping: $x=10$ (global i)

under dynamic scoping: $y=20$ (according to the sequence of calls,i.e 20)

11 votes

-- sumit kumar singh dixit (2.1k points)

5 Theory of Computation (281) top

5.1 Closure Property(10) top

5.1.1 Closure Property: GATE1989-3-ii top

<https://gateoverflow.in/87117>

Context-free languages and regular languages are both closed under the operation (s) of :

- A. Union
- B. Intersection
- C. Concatenation
- D. Complementation

[gate1989](#) [easy](#) [theory-of-computation](#) [closure-property](#)

Answer

5.1.2 Closure Property: GATE1992-16 top

<https://gateoverflow.in/595>

Which of the following three statements are true? Prove your answer.

- i. The union of two recursive languages is recursive.
- ii. The language $\{O^n \mid n \text{ is a prime}\}$ is not regular.
- iii. Regular languages are closed under infinite union.

[gate1992](#) [theory-of-computation](#) [normal](#) [closure-property](#)

Answer

5.1.3 Closure Property: GATE2002-2.14 top

<https://gateoverflow.in/844>

Which of the following is true?

- A. The complement of a recursive language is recursive
- B. The complement of a recursively enumerable language is recursively enumerable
- C. The complement of a recursive language is either recursive or recursively enumerable
- D. The complement of a context-free language is context-free

[gate2002](#) [theory-of-computation](#) [easy](#) [closure-property](#)

Answer

5.1.4 Closure Property: GATE2006-IT-32 top

<https://gateoverflow.in/3571>

Let L be a context-free language and M a regular language. Then the language $L \cap M$ is

- A. always regular
- B. never regular
- C. always a deterministic context-free language
- D. always a context-free language

[gate2006-it](#) [theory-of-computation](#) [closure-property](#) [easy](#)

Answer

5.1.5 Closure Property: GATE2013-17 [top](#)

<https://gateoverflow.in/1439>

Which of the following statements is/are **FALSE**?

1. For every non-deterministic Turing machine, there exists an equivalent deterministic Turing machine.
 2. Turing recognizable languages are closed under union and complementation.
 3. Turing decidable languages are closed under intersection and complementation.
 4. Turing recognizable languages are closed under union and intersection.
- A. 1 and 4 only
 B. 1 and 3 only
 C. 2 only
 D. 3 only

[gate2013](#) [theory-of-computation](#) [normal](#) [closure-property](#)

Answer

5.1.6 Closure Property: GATE2016-2-18 [top](#)

<https://gateoverflow.in/39574>

Consider the following types of languages: L_1 : Regular, L_2 : Context-free, L_3 : Recursive, L_4 : Recursively enumerable. Which of the following is/are **TRUE** ?

- I. $L_3 \cup L_4$ is recursively enumerable.
 - II. $L_2 \cup L_3$ is recursive.
 - III. $L_1^* \cap L_2$ is context-free.
 - IV. $L_1 \cup L_2$ is context-free.
- A. I only.
 B. I and III only.
 C. I and IV only.
 D. I, II and III only.

[gate2016-2](#) [theory-of-computation](#) [regular-languages](#) [context-free-language](#) [closure-property](#) [normal](#)

Answer

5.1.7 Closure Property: GATE2017-2-04 [top](#)

<https://gateoverflow.in/118143>

Let L_1, L_2 be any two context-free languages and R be any regular language. Then which of the following is/are CORRECT?

- I. $L_1 \cup L_2$ is context-free
 - II. $\overline{L_1}$ is context-free
 - III. $L_1 - R$ is context-free
 - IV. $L_1 \cap L_2$ is context-free
- A. I, II and IV only
 B. I and III only
 C. II and IV only
 D. I only

[gate2017-2](#) [theory-of-computation](#) [closure-property](#)

Answer

5.1.8 Closure Property: GATE2018-7 [top](#)

<https://gateoverflow.in/204081>

The set of all recursively enumerable languages is:

- A. closed under complementation
- B. closed under intersection
- C. a subset of the set of all recursive languages
- D. an uncountable set

gate2018 theory-of-computation closure-property easy

[Answer](#)

5.1.9 Closure Property: TIFR2013-B-11 [top](#)

<https://gateoverflow.in/25772>

Which of the following statements is FALSE?

- A. The intersection of a context free language with a regular language is context free.
- B. The intersection of two regular languages is regular.
- C. The intersection of two context free languages is context free
- D. The intersection of a context free language and the complement of a regular language is context free.
- E. The intersection of a regular language and the complement of a regular language is regular.

tifr2013 theory-of-computation closure-property

[Answer](#)

5.1.10 Closure Property: TIFR2014-B-14 [top](#)

<https://gateoverflow.in/27321>

Which the following is FALSE?

- A. Complement of a recursive language is recursive.
- B. A language recognized by a non-deterministic Turing machine can also be recognized by a deterministic Turing machine.
- C. Complement of a context free language can be recognized by a Turing machine.
- D. If a language and its complement are both recursively enumerable then it is recursive.
- E. Complement of a non-recursive language can never be recognized by any Turing machine.

tifr2014 theory-of-computation closure-property

[Answer](#)

Answers: Closure Property

5.1.1 Closure Property: GATE1989-3-ii [top](#)

<https://gateoverflow.in/87117>



Selected Answer

Regular language is closed under: **Union, Intersection, Concatenation, Complementation.**

CFL is closed under **Union Concatenation but not under Intersection or Complementation.**

So, both closed under Union Concatenation.

21 votes

-- Prashant Singh (59.8k points)

5.1.2 Closure Property: GATE1992-16 top

<https://gateoverflow.in/595>



Selected Answer

- i. True. Recursive languages are closed under union.
- ii. True. The language is Turing Machine acceptable.
- iii. False. Regular languages are closed under finite union.

16 votes

-- Rajarshi Sarkar (34.1k points)

5.1.3 Closure Property: GATE2002-2.14 top

<https://gateoverflow.in/844>



Selected Answer

Complement of recursive language is always recursive.

19 votes

-- Bhagirathi Nayak (14.1k points)

5.1.4 Closure Property: GATE2006-IT-32 top

<https://gateoverflow.in/3571>



Selected Answer

Let $\Sigma = \{a, b\}$.

$L_1 = \Sigma^*$ is a regular language

$L_2 = \{ww^R \mid w \in (a+b)^*\}$ is a context free language.

$L_1 \cap L_2 = \{ww^R \mid w \in (a+b)^*\}$ which is clearly context free language and not DCFL or Regular

Hence, the answer is option D.

25 votes

-- Mari Ganesh Kumar (2k points)

5.1.5 Closure Property: GATE2013-17 top

<https://gateoverflow.in/1439>



Selected Answer

Recursive enumerable languages are not closed under complement . while recursive languages are.

Both Recursive and Recursive enumerable languages are closed under intersection, union, and kleene star.

http://gatcse.in/wiki/Closure_Property_of_Language_Families

Non-Deterministic TM is equivalent to DTM

Only 2 is false. Option C is correct.

Note: Turing decidable language mean Recursive language and Turing recognizable language mean recursive enumerable language.

31 votes

-- Praveen Saini (54.6k points)

5.1.6 Closure Property: GATE2016-2-18 top

<https://gateoverflow.in/39574>



I. I. $\overline{L_3} \cup L_4$

L_3 is recursive, so $\overline{L_3}$ is also recursive (closed under complement),

So, $\overline{L_3}$ is recursive enumerable.

L_4 is recursive enumerable,

so, $\overline{L_3} \cup L_4$ is also recursive enumerable (closed under union).

II. $\overline{L_2} \cup L_3$

L_2 is Context-free, so $\overline{L_2}$, may or may not be Context-free (not closed under complement), but definitely $\overline{L_2}$ is Recursive.

L_3 is recursive.

so $\overline{L_2} \cup L_3$ is also recursive (closed under union).

III. $L_1^* \cap L_2$

L_1 is Regular, so L_1^* is also regular (closed under kleene star)

L_2 is Context-free

so, $L_1^* \cap L_2$ is also context-free (closed under intersection with regular).

IV. $L_1 \cup \overline{L_2}$

L_1 is regular.

L_2 is context-free, so $\overline{L_2}$ may or may not be Context-free (not closed under complement).

so, $L_1 \cup \overline{L_2}$ may or may not be Context-free.

Here, answer is **D**.

Upvote 55 votes

-- Praveen Saini (54.6k points)

5.1.7 Closure Property: GATE2017-2-04 top

<https://gateoverflow.in/118143>



Statement I is **TRUE** as CFGs are closed under union.

Statement II is **FALSE** as CFGs are not enclosed under complementation.

Statement III is **TRUE** as $L_1 - R$ can be written as $L_1 \cap \bar{R}$. Regular language are closed under complementation and intersection of CFG and Regular is CFG.

Statement IV is **FALSE** as CFGs are not enclosed under intersection.

So, I and III are correct. Option B

Upvote 10 votes

-- Prateek Kumar (1.2k points)

5.1.8 Closure Property: GATE2018-7 top

<https://gateoverflow.in/204081>



Answer is **B**.

Reference: <https://gatecse.in/closure-property-of-language-families/>

C is false as the set of all recursively enumerable languages (semi-decidable) is a STRICT super set of the set of all recursive languages (decidable).

D is false as the set of all recursively enumerable languages (set of all Turing machines) is an infinite but countable set.

22 votes

-- Arjun Suresh (350k points)

5.1.9 Closure Property: TIFR2013-B-11 top

<https://gateoverflow.in/25772>



Context language are not closed under intersection so option **c**.

11 votes

-- Umang Raman (15.7k points)

5.1.10 Closure Property: TIFR2014-B-14 top

<https://gateoverflow.in/27321>



- A. True. For a recursive language, we have a TM which says for any string "w", "yes" if it belongs to the language and "no" if it does not. For the complement of a recursive language we just have to reverse 'yes' and 'no' conditions and this is possible with a slight modification to the original TM. So, the new language is also recursive.
- B. True. Non-determinism does not add any recognizing power to a Turing machine though it can affect the time complexity in solving a problem.
- C. True. Complement of CFL is recursive. and recursive language is recognized by Turing machine. So, complement of a context free language can be recognized by TM
- D. True. If a language is r.e., we have a TM which says "yes" whenever a given string belongs to L. Similarly, if its complement is r.e., we have a TM which says "yes", whenever a string does not belong to the language. So, combining both we can always say "yes" if a string belongs to the language and "no" if a string does not belong to L \implies L is recursive.
- E. False.

Non-recursive Language can be:- (a) **r.e.** or (b) **Non r.e.**

As seen in option D, if the language is RE but not recursive, its complement cannot be r.e. (recognized by a TM). But, if the language is not r.e. its complement may be r.e. (which **Can be recognized by TM**). **option E is false.**

Example:-

$L = \{\langle M, w \rangle \mid M \text{ is a TM and it does not halt on string } w\}$. //Non RE (complement of halting problem)

$L^c = \{\langle M, w \rangle \mid M \text{ is a TM and it halts on string } w\}$ //RE but not Recursive //recognized by TM (halting problem)

So, answer is (E).

11 votes

-- srestha (87k points)

5.2

Context Free Language(31) top

5.2.1 Context Free Language: GATE1987-1-xii top

<https://gateoverflow.in/80291>

A context-free grammar is ambiguous if:

- A. The grammar contains useless non-terminals.
- B. It produces more than one parse tree for some sentence.
- C. Some production has two non terminals side by side on the right-hand side.
- D. None of the above.

[gate1987](#) [theory-of-computation](#) [context-free-language](#) [ambiguous](#)

[Answer](#)

5.2.2 Context Free Language: GATE1987-2k [top](#)

<https://gateoverflow.in/80599>

State whether the following statements are TRUE or FALSE:

The intersection of two CFL's is also a CFL.

[gate1987](#) [theory-of-computation](#) [context-free-language](#)

[Answer](#)

5.2.3 Context Free Language: GATE1992-02,xix [top](#)

<https://gateoverflow.in/572>

02. Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

(xix) Context-free languages are:

- A. closed under union
- B. closed under complementation
- C. closed under intersection
- D. closed under Kleene closure

[gate1992](#) [context-free-language](#) [theory-of-computation](#) [normal](#)

[Answer](#)

5.2.4 Context Free Language: GATE1992-02,xviii [top](#)

<https://gateoverflow.in/576>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

If G is a context free grammar and w is a string of length l in $L(G)$, how long is a derivation of w in G , if G is in Chomsky normal form?

- A. $2l$
- B. $2l + 1$
- C. $2l - 1$
- D. l

[gate1992](#) [theory-of-computation](#) [context-free-language](#) [easy](#)

[Answer](#)

5.2.5 Context Free Language: GATE1995-2.20 [top](#)

<https://gateoverflow.in/2632>

Which of the following definitions below generate the same language as L , where $L = \{x^n y^n \text{ such that } n \geq 1\}$?

I. $E \rightarrow xEy \mid xy$

II. $xy \mid (x^+xyy^+)$

III. x^+y^+

- A. I only
- B. I and II
- C. II and III
- D. II only

gate1995 theory-of-computation easy context-free-language

[Answer](#)

5.2.6 Context Free Language: GATE1996-2.8 [top](#)

<https://gateoverflow.in/2737>

If L_1 and L_2 are context free languages and R a regular set, one of the languages below is not necessarily a context free language. Which one?

- A. $L_1 \cdot L_2$
- B. $L_1 \cap L_2$
- C. $L_1 \cap R$
- D. $L_1 \cup L_2$

gate1996 theory-of-computation context-free-language easy

[Answer](#)

5.2.7 Context Free Language: GATE1996-2.9 [top](#)

<https://gateoverflow.in/2738>

Define a context free languages $L \in \{0,1\}^*$, $\text{init}(L) = \{u \mid uv \in L \text{ for some } v \text{ in } \{0,1\}^*\}$ (in other words, $\text{init}(L)$ is the set of prefixes of L)

Let $L = \{w \mid w \text{ is nonempty and has an equal number of 0's and 1's}\}$

Then $\text{init}(L)$ is:

- A. the set of all binary strings with unequal number of 0's and 1's
- B. the set of all binary strings including null string
- C. the set of all binary strings with exactly one more 0 than the number of 1's or one more 1 than the number of 0's
- D. None of the above

gate1996 theory-of-computation context-free-language normal

[Answer](#)

5.2.8 Context Free Language: GATE1999-1.5 [top](#)

<https://gateoverflow.in/1459>

Context-free languages are closed under:

- A. Union, intersection
- B. Union, Kleene closure
- C. Intersection, complement
- D. Complement, Kleene closure

gate1999 theory-of-computation context-free-language easy

[Answer](#)

5.2.9 Context Free Language: GATE1999-7 [top](#)

<https://gateoverflow.in/1506>

Show that the language

$$L = \{x c x \mid x \in \{0,1\}^* \text{ and } c \text{ is a terminal symbol}\}$$

is not context free. c is not 0 or 1.

gate1999 theory-of-computation context-free-language normal

[Answer](#)

5.2.10 Context Free Language: GATE2000-7 [top](#)

<https://gateoverflow.in/678>

- A. Construct as minimal finite state machine that accepts the language, over $\{0,1\}$, of all strings that contain neither the sub string 00 nor the sub string 11.
- B. Consider the grammar

$S \rightarrow$	aSAb
$S \rightarrow$	ϵ
$A \rightarrow$	bA
$A \rightarrow$	ϵ

where S, A are non-terminal symbols with S being the start symbol; a, b are terminal symbols and ϵ is the empty string. This grammar generates strings of the form $a^i b^j$ for some $i, j \geq 0$, where i and j satisfy some condition. What is the condition on the values of i and j ?

gate2000 theory-of-computation descriptive regular-languages context-free-language

[Answer](#)

5.2.11 Context Free Language: GATE2001-1.5 [top](#)

<https://gateoverflow.in/698>

Which of the following statements is true?

- A. If a language is context free it can always be accepted by a deterministic push-down automaton
- B. The union of two context free languages is context free
- C. The intersection of two context free languages is a context free
- D. The complement of a context free language is a context free

gate2001 theory-of-computation context-free-language easy

[Answer](#)

5.2.12 Context Free Language: GATE2003-51 [top](#)

<https://gateoverflow.in/940>

Let $G = (\{S\}, \{a,b\}, R, S)$ be a context free grammar where the rule set R is $S \rightarrow aSb \mid SS \mid \epsilon$

Which of the following statements is true?

- A. G is not ambiguous
- B. There exist $x, y \in L(G)$ such that $xy \notin L(G)$
- C. There is a deterministic pushdown automaton that accepts $L(G)$
- D. We can find a deterministic finite state automaton that accepts $L(G)$

gate2003 theory-of-computation context-free-language normal

[Answer](#)

5.2.13 Context Free Language: GATE2005-57 [top](#)

<https://gateoverflow.in/1380>

Consider the languages:

- $L_1 = \{ww^R \mid w \in \{0,1\}^*\}$
- $L_2 = \{w\#w^R \mid w \in \{0,1\}^*\}$, where $\#$ is a special symbol
- $L_3 = \{ww \mid w \in \{0,1\}^*\}$

Which one of the following is TRUE?

- A. L_1 is a deterministic CFL
- B. L_2 is a deterministic CFL
- C. L_3 is a CFL, but not a deterministic CFL
- D. L_3 is a deterministic CFL

gate2005 theory-of-computation context-free-language easy

[Answer](#)

5.2.14 Context Free Language: GATE2006-19 [top](#)

<https://gateoverflow.in/980>

Let

$$L_1 = \{0^{n+m}1^n0^m \mid n, m \geq 0\},$$

$$L_2 = \{0^{n+m}1^{n+m}0^m \mid n, m \geq 0\} \text{ and}$$

$$L_3 = \{0^{n+m}1^{n+m}0^{n+m} \mid n, m \geq 0\}.$$

Which of these languages are NOT context free?

- A. L_1 only
- B. L_3 only
- C. L_1 and L_2
- D. L_2 and L_3

gate2006 theory-of-computation context-free-language normal

[Answer](#)

5.2.15 Context Free Language: GATE2006-IT-34 [top](#)

<https://gateoverflow.in/3573>

In the context-free grammar below, S is the start symbol, a and b are terminals, and ϵ denotes the empty string.

- $S \rightarrow aSAb \mid \epsilon$
- $A \rightarrow bA \mid \epsilon$

The grammar generates the language

- $((a+b)^*b)$
- $\{a^m b^n \mid m \leq n\}$
- $\{a^m b^n \mid m = n\}$
- a^*b^*

gate2006-it theory-of-computation context-free-language normal

Answer

5.2.16 Context Free Language: GATE2006-IT-4 top

<https://gateoverflow.in/3543>

In the context-free grammar below, S is the start symbol, a and b are terminals, and ϵ denotes the empty string

$$S \rightarrow aSa \mid bSb \mid a \mid b \mid \epsilon$$

Which of the following strings is NOT generated by the grammar?

- $aaaa$
- $baba$
- $abba$
- $babaabab$

gate2006-it theory-of-computation context-free-language easy

Answer

5.2.17 Context Free Language: GATE2007-IT-46 top

<https://gateoverflow.in/3481>

The two grammars given below generate a language over the alphabet $\{x, y, z\}$

$$\begin{aligned} G1 : S &\rightarrow x \mid z \mid xS \mid zS \mid yB \\ B &\rightarrow y \mid z \mid yB \mid zB \end{aligned}$$

$$\begin{aligned} G2 : S &\rightarrow y \mid z \mid yS \mid zS \mid xB \\ B &\rightarrow y \mid yS \end{aligned}$$

Which one of the following choices describes the properties satisfied by the strings in these languages?

- $G1$: No y appears before any x
 $G2$: Every x is followed by at least one y
- $G1$: No y appears before any x
 $G2$: No x appears before any y
- $G1$: No y appears after any x
 $G2$: Every x is followed by at least one y
- $G1$: No y appears after any x
 $G2$: Every y is followed by at least one x

gate2007-it theory-of-computation normal context-free-language

Answer

5.2.18 Context Free Language: GATE2007-IT-48 top

<https://gateoverflow.in/3490>

Consider the grammar given below:

$$\begin{aligned} S &\rightarrow xB \mid yA \\ A &\rightarrow x \mid xS \mid yAA \\ B &\rightarrow y \mid yS \mid xBB \end{aligned}$$

Consider the following strings.

- i. $xyyx$
- ii. $xyyxy$
- iii. $xyxy$
- iv. yxx
- v. yxx
- vi. xyx

Which of the above strings are generated by the grammar ?

- A. i, ii and iii
- B. ii, v and vi
- C. ii, iii and iv
- D. i, iii and iv

[gate2007-it](#) [theory-of-computation](#) [context-free-language](#) [normal](#)

[Answer](#)

5.2.19 Context Free Language: GATE2007-IT-49 [top](#)

<https://gateoverflow.in/3491>

Consider the following grammars. Names representing terminals have been specified in capital letters.

$$\begin{array}{l} G1 \quad \text{stmtnt} \rightarrow \text{WHILE} \\ : \quad \text{stmtnt} \rightarrow (\text{expr}) \text{stmtnt} \\ \quad \text{stmtnt} \rightarrow \text{OTHER} \\ \quad \text{expr} \rightarrow \text{ID} \end{array}$$

$$\begin{array}{l} G2 \quad \text{stmtnt} \rightarrow \text{WHILE} \\ : \quad \text{stmtnt} \rightarrow (\text{expr}) \text{stmtnt} \\ \quad \text{stmtnt} \rightarrow \text{OTHER} \\ \quad \text{expr} \rightarrow \text{expr} + \text{expr} \\ \quad \text{expr} \rightarrow \text{expr} * \text{expr} \\ \quad \text{expr} \rightarrow \text{ID} \end{array}$$

Which one of the following statements is true?

- A. G_1 is context-free but not regular and G_2 is regular
- B. G_2 is context-free but not regular and G_1 is regular
- C. Both G_1 and G_2 are regular
- D. Both G_1 and G_2 are context-free but neither of them is regular

[gate2007-it](#) [theory-of-computation](#) [context-free-language](#) [normal](#)

[Answer](#)

5.2.20 Context Free Language: GATE2008-IT-34 [top](#)

<https://gateoverflow.in/3344>

Consider a CFG with the following productions.

$$\begin{aligned} S &\rightarrow AA \mid B \\ A &\rightarrow 0A \mid A0 \mid 1 \end{aligned}$$

$$B \rightarrow 0B00 \mid 1$$

S is the start symbol, A and B are non-terminals and 0 and 1 are the terminals. The language generated by this grammar is:

- A. $\{0^n 1 0^{2n} \mid n \geq 1\}$
- B. $\{0^i 1 0^j 1 0^k \mid i, j, k \geq 0\} \cup \{0^n 1 0^{2n} \mid n \geq 0\}$
- C. $\{0^i 1 0^j \mid i, j \geq 0\} \cup \{0^n 1 0^{2n} \mid n \geq 0\}$
- D. The set of all strings over $\{0, 1\}$ containing at least two 0's

gate2008-it theory-of-computation context-free-language normal

Answer

5.2.21 Context Free Language: GATE2008-IT-78 top

<https://gateoverflow.in/3392>

A $CFG G$ is given with the following productions where S is the start symbol, A is a non-terminal and a and b are terminals.

$$S \rightarrow aS \mid AA \rightarrow aAb \mid bAa \mid \epsilon$$

Which of the following strings is generated by the grammar above?

- A. $aabbaba$
- B. $aabaaba$
- C. $abababb$
- D. $aabbaab$

gate2008-it theory-of-computation normal context-free-language

Answer

5.2.22 Context Free Language: GATE2009-12, ISRO2016-37 top

<https://gateoverflow.in/1304>

$$S \rightarrow aSa \mid bSb \mid a \mid b$$

The language generated by the above grammar over the alphabet $\{a, b\}$ is the set of:

- A. all palindromes
- B. all odd length palindromes
- C. strings that begin and end with the same symbol
- D. all even length palindromes

gate2009 theory-of-computation context-free-language easy isro2016

Answer

5.2.23 Context Free Language: GATE2015-3-32 top

<https://gateoverflow.in/8489>

Which of the following languages are context-free?

$$L_1 : \{a^m b^n a^n b^m \mid m, n \geq 1\}$$

$$L_2 : \{a^m b^n a^m b^n \mid m, n \geq 1\}$$

$$L_3 : \{a^m b^n \mid m = 2n + 1\}$$

- A. L_1 and L_2 only
 B. L_1 and L_3 only
 C. L_2 and L_3 only
 D. L_3 only

gate2015-3 theory-of-computation context-free-language normal

[Answer](#)

5.2.24 Context Free Language: GATE2016-1-16 [top](#)

<https://gateoverflow.in/39640>

Which of the following languages is generated by the given grammar?

$$S \rightarrow aS \mid bS \mid \epsilon$$

- A. $\{a^n b^m \mid n, m \geq 0\}$
 B. $\{w \in \{a, b\}^* \mid w \text{ has equal number of } a's \text{ and } b's\}$
 C. $\{a^n \mid n \geq 0\} \cup \{b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$
 D. $\{a, b\}^*$

gate2016-1 theory-of-computation context-free-language normal

[Answer](#)

5.2.25 Context Free Language: GATE2016-1-42 [top](#)

<https://gateoverflow.in/39705>

Consider the following context-free grammars;

$$G_1 : S \rightarrow aS \mid B, B \rightarrow b \mid bB$$

$$G_2 : S \rightarrow aA \mid bB, A \rightarrow aA \mid B \mid \epsilon, B \rightarrow bB \mid \epsilon$$

Which one of the following pairs of languages is generated by G_1 and G_2 , respectively?

- A. $\{a^m b^n \mid m > 0 \text{ or } n > 0\}$ and $\{a^m b^n \mid m > 0 \text{ and } n > 0\}$
 B. $\{a^m b^n \mid m > 0 \text{ and } n > 0\}$ and $\{a^m b^n \mid m > 0 \text{ or } n \geq 0\}$
 C. $\{a^m b^n \mid m \geq 0 \text{ or } n > 0\}$ and $\{a^m b^n \mid m > 0 \text{ and } n > 0\}$
 D. $\{a^m b^n \mid m \geq 0 \text{ and } n > 0\}$ and $\{a^m b^n \mid m > 0 \text{ or } n > 0\}$

gate2016-1 theory-of-computation context-free-language normal

[Answer](#)

5.2.26 Context Free Language: GATE2016-2-43 [top](#)

<https://gateoverflow.in/39605>

Consider the following languages:

$$L_1 = \{a^n b^m c^{n+m} : m, n \geq 1\}$$

$$L_2 = \{a^n b^n c^{2n} : n \geq 1\}$$

Which one of the following is TRUE?

- A. Both L_1 and L_2 are context-free.
 B. L_1 is context-free while L_2 is not context-free.
 C. L_2 is context-free while L_1 is not context-free.
 D. Neither L_1 nor L_2 is context-free.

[gate2016-2](#) [theory-of-computation](#) [context-free-language](#) [normal](#)
Answer

5.2.27 Context Free Language: GATE2017-1-10 [top](#)

<https://gateoverflow.in/118290>

Consider the following context-free grammar over the alphabet $\Sigma = \{a, b, c\}$ with S as the start symbol:

$$S \rightarrow abScT \mid abcT$$

$$T \rightarrow bT \mid b$$

Which one of the following represents the language generated by the above grammar?

- A. $\{(ab)^n(cb)^n \mid n \geq 1\}$
- B. $\{(ab)^n cb^{m_1} cb^{m_2} \dots cb^{m_n} \mid n, m_1, m_2, \dots, m_n \geq 1\}$
- C. $\{(ab)^n (cb^m)^n \mid m, n \geq 1\}$
- D. $\{(ab)^n (cb^n)^m \mid m, n \geq 1\}$

[gate2017-1](#) [theory-of-computation](#) [context-free-language](#) [normal](#)
Answer

5.2.28 Context Free Language: GATE2017-1-34 [top](#)

<https://gateoverflow.in/118316>

If G is a grammar with productions

$$S \rightarrow SaS \mid aSb \mid bSa \mid SS \in$$

where S is the start variable, then which one of the following strings is not generated by G ?

- A. $abab$
- B. $aaab$
- C. $abbaa$
- D. $babba$

[gate2017-1](#) [theory-of-computation](#) [context-free-language](#) [normal](#)
Answer

5.2.29 Context Free Language: GATE2017-1-38 [top](#)

<https://gateoverflow.in/118321>

Consider the following languages over the alphabet $\Sigma = \{a, b, c\}$. Let $L_1 = \{a^n b^n c^m \mid m, n \geq 0\}$ and $L_2 = \{a^m b^n c^n \mid m, n \geq 0\}$.

Which of the following are context-free languages?

- I. $L_1 \cup L_2$
- II. $L_1 \cap L_2$
- A. I only
- B. II only
- C. I and II
- D. Neither I nor II

gate2017-1 theory-of-computation context-free-language normal

Answer

5.2.30 Context Free Language: GATE2017-2-16 [top](#)

<https://gateoverflow.in/118243>

Identify the language generated by the following grammar, where S is the start variable.

- $S \rightarrow XY$
- $X \rightarrow aX \mid a$
- $Y \rightarrow aYb \mid \epsilon$

- A. $\{a^m b^n \mid m \geq n, n > 0\}$
 B. $\{a^m b^n \mid m \geq n, n \geq 0\}$
 C. $\{a^m b^n \mid m > n, n \geq 0\}$
 D. $\{a^m b^n \mid m > n, n > 0\}$

gate2017-2 theory-of-computation context-free-language

Answer

5.2.31 Context Free Language: ISI2015-CS-5b [top](#)

<https://gateoverflow.in/47330>

Construct a context free grammar (CFG) to generate the following language:

$$L = \{a^n b^m c^{n+m} : n, m \text{ are integers, and } n \geq 1, m \geq 1\}$$

descriptive isi2015 theory-of-computation context-free-language

Answer

Answers: Context Free Language

5.2.1 Context Free Language: GATE1987-1-xii [top](#)

<https://gateoverflow.in/80291>



Selected Answer

An ambiguous grammar produces more than one parse tree for any string.. that's B..

16 votes

-- kirti singh (3.8k points)

5.2.2 Context Free Language: GATE1987-2k [top](#)

<https://gateoverflow.in/80599>



Selected Answer

No intersection of two CFLs may or may not be a CFL i.e. CFL is not closed under intersection operation.

Example:

$$L_1 : \{a^n b^n c^m \mid n, m \geq 1\} \cap L_2 : \{a^n b^m c^m \mid n, m \geq 1\}$$

$$L_3 = \{a^m b^m c^m \mid m \geq 1\} \text{ Which is CSL}$$

16 votes

-- Prashant Singh (59.8k points)

5.2.3 Context Free Language: GATE1992-02,xix top

<https://gateoverflow.in/572>



Answer: **a, d.**

Context Free languages are not closed under intersection and complementation.

14 votes

-- Rajarshi Sarkar (34.1k points)

5.2.4 Context Free Language: GATE1992-02,xviii top

<https://gateoverflow.in/576>



Chomsky Normal Form (If all of its production rules are of the form):

$$A \rightarrow BC \text{ or}$$

$$A \rightarrow a \text{ or}$$

$$S \rightarrow \epsilon$$

where A, B and C are nonterminal symbols, a is a terminal symbol (a symbol that represents a constant value), S is the start symbol, and ϵ is the empty string. Also, neither B nor C may be the start symbol, and the third production rule can only appear if ϵ is in $L(G)$, namely, the language produced by the context-free grammar G .

Applying productions of the first form will increase the number of nonterminals from k to $k+1$, since you replace one nonterminal (-1) with two nonterminals ($+2$) for a net gain of $+1$ nonterminal. Since you start with one nonterminal, this means you need to do $l-1$ productions of the first form. You then need l more of the second form to convert the nonterminals to terminals, giving a total of $l + (l-1) = 2l-1$ productions.

19 votes

-- Rajarshi Sarkar (34.1k points)

5.2.5 Context Free Language: GATE1995-2.20 top

<https://gateoverflow.in/2632>



(A) In the other two you can have any number of x and y . There is no such restriction over the number of both being equal.

19 votes

-- Gate Keeda (19.6k points)

5.2.6 Context Free Language: GATE1996-2.8 top

<https://gateoverflow.in/2737>



(B) CFL's are not closed under intersection.

15 votes

-- Gate Keeda (19.6k points)

5.2.7 Context Free Language: GATE1996-2.9 top

<https://gateoverflow.in/2738>



(B) is the answer. Because for any binary string of 0's and 1's we can append another string to make it contain equal number of 0's and 1's. i.e., any string over $\{0, 1\}$ is a prefix of a string in L .

Example:

01111 - is prefix of 011110000 which is in L .

1111- is prefix of 11110000 which is in L .

01010- is prefix of 010101 which is in L .

20 votes

-- Arjun Suresh (350k points)

5.2.8 Context Free Language: GATE1999-1.5 top

<https://gateoverflow.in/1459>



Cfl are not closed under intersection and complement now choose the correct option, so (B) union and klenne closure.

16 votes

-- Bhagirathi Nayak (14.1k points)

5.2.9 Context Free Language: GATE1999-7 top

<https://gateoverflow.in/1506>



Language contains strings where sub string on left of c is same as that on right of c

say 01100 c 01100

sub string on left of c and right of c cannot be matched with one Stack

while that can be done using two stack

if we push all 0's and 1's on left of c in stack 1 , and all 0's and 1' on right of c in stack 2

then **Top of stack** of both stack will have same symbol. That can be matched .

20 votes

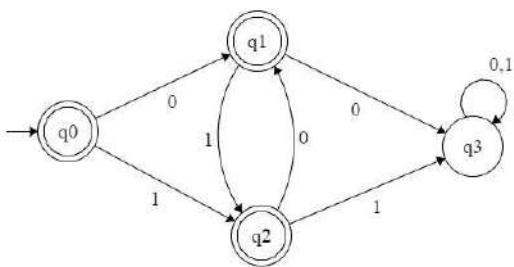
-- Praveen Saini (54.6k points)

5.2.10 Context Free Language: GATE2000-7 top

<https://gateoverflow.in/678>



- a. langauge $L = (0 + 1)^* - (0 + 1)^*(00 + 11)(0 + 1)^*$ is it true ?? DFA contains 4 states , 3 are final , 1 is dead state



- b. $i \leq j$
as $S \rightarrow aSAb$

there will be always for one a in left and minimum one b in right and $A \rightarrow bA \mid \Lambda$ can generate any no of b 's including null , if A is Λ then $i = j$ and if A is generate any b then $j > i$ so condition is $i \leq j$

23 votes

-- Mithlesh Upadhyay (5.8k points)

5.2.11 Context Free Language: GATE2001-1.5 top

<https://gateoverflow.in/698>



Selected Answer

Answer is (B).

(A) is wrong as a language can be context free even if it is being accepted by non-deterministic PDA for ex- $\{ww^r : w \in \Sigma^* \text{ and } w^r \text{ is reverse of } w\}$

(C) and (D) not always true as Context free languages are not closed under Complement or Intersection.

21 votes

-- Prateeksha Keshari (2.2k points)

5.2.12 Context Free Language: GATE2003-51 top

<https://gateoverflow.in/940>



Selected Answer

It will be easy to analyze the problem if we replace terminal a and b by (and) respectively.

$$S \rightarrow (S) \mid SS \mid \epsilon$$

$L(G)$ = balanced parentheses [each left parenthesis has a matching right parenthesis and are well nested]

example $((), ()(), (()), ((())())$.

A. $S \Rightarrow (S) \Rightarrow ()$
 $S \Rightarrow SS \Rightarrow S \Rightarrow (S) \Rightarrow ()$
 $S \Rightarrow SS \Rightarrow S \Rightarrow (S) \Rightarrow ()$

String $()$ can be derived by above three way each having different derivation tree.
So G is Ambiguous

- B. Concatenation of two balance parenthesis will be balanced also . eq.
 $x = (())$ $y = ()$ $xy = ((())()$.
C. We can design Deterministic PDA for L. put left parenthesis (only) in stack and pop with right parenthesis.

- D. We cannot design DFA for L because we need a stack to match left parenthesis with right parenthesis.
only option C is true.

36 votes

-- Praveen Saini (54.6k points)

5.2.13 Context Free Language: GATE2005-57 top

<https://gateoverflow.in/1380>



(B) is TRUE.

Reference: http://gatecse.in/wiki/Identify_the_class_of_the_language

13 votes

-- Gate Keeda (19.6k points)

5.2.14 Context Free Language: GATE2006-19 top

<https://gateoverflow.in/980>



L_1 is context-free. We count the number of 0's and check if the remaining number of 1's followed by 0's count to the initial number of 0's.

L_2 is not context-free. Here the number of 0's and the following 1's must be same, which can be checked using a PDA. But after that we must also ensure that the following number of 0's must be less than the previous count of 0's and 1's (otherwise $n < 0$, which violates the condition for acceptance) and we cannot do these two checks using a single PDA.

L_3 is again not context-free as it is nothing but equal number of 0's followed by equal number of 1's followed by equal number of 0's.

29 votes

-- Arjun Suresh (350k points)

5.2.15 Context Free Language: GATE2006-IT-34 top

<https://gateoverflow.in/3573>



$$A \rightarrow bA \mid \epsilon$$

$$\therefore A = b^*$$

$$S \rightarrow aSAb \mid \epsilon$$

$$\equiv S \rightarrow aSb^*b \mid \epsilon$$

$$\equiv S \rightarrow aSb^+ \mid \epsilon$$

$$S = a^n(b^+)^n, \quad n \geq 0$$

$$S = a^n b^n b^*, \quad n \geq 0$$

$$S = a^m b^n, \quad m \leq n$$

Hence, option B is correct.

37 votes

-- Pragy Agarwal (20.6k points)

5.2.16 Context Free Language: GATE2006-IT-4 top

<https://gateoverflow.in/3543>



$$L(G) = \text{PALINDROME}$$

baba does not belong to palindrome , so B is the answer.

18 votes

-- Praveen Saini (54.6k points)

5.2.17 Context Free Language: GATE2007-IT-46 top

<https://gateoverflow.in/3481>



From Above grammar:

Regular expression for G1 : $(x + z)^+ + (x + z)^*y(y + z)^+$

Regular expression for G2 : $(y + z + xy)^+$

Option (A) is correct .

15 votes

-- Praveen Saini (54.6k points)

5.2.18 Context Free Language: GATE2007-IT-48 top

<https://gateoverflow.in/3490>



ii, iii and iv.

So, option (C) is correct.

Above grammar is for **equal no of x and y**

from Non-terminal $S \rightarrow xB$

$$\Rightarrow xy \quad [\text{as } B \rightarrow y \text{ one } y \text{ for one } x]$$

$$S \rightarrow xB$$

$$\Rightarrow xxBB \quad [\text{as } B \rightarrow yBB \text{ one } B \text{ result in one } y \text{ for one } x]$$

$$S \rightarrow xB$$

$$\Rightarrow xyS \quad [\text{as } B \rightarrow yS \text{ one } y \text{ for one } x \text{ and start again}]$$

Note: Same applies for string start with y i.e . $S \rightarrow yA$.

20 votes

-- Praveen Saini (54.6k points)

5.2.19 Context Free Language: GATE2007-IT-49 top

<https://gateoverflow.in/3491>



Selected Answer

Regular grammar is either right linear or left linear. A left linear grammar is one in which there is at most 1 non-terminal on the right side of any production, and it appears at the left most position. Similarly, in right linear grammar non-terminal appears at the right most position.

Here, we can write a right linear grammar for $G1$ as

$$\begin{aligned} S &\rightarrow w(E \\ E &\rightarrow id)S \\ S &\rightarrow o \end{aligned}$$

(w - WHILE, o - OTHER)

So, $L(G1)$ is regular.

Now for $G2$ also we can write a right linear grammar:

$$\begin{aligned} S &\rightarrow w(E \\ E &\rightarrow id)S \\ E &\rightarrow id + E \\ E &\rightarrow id * E \\ S &\rightarrow o \end{aligned}$$

making its language regular.

So, both $G1$ and $G2$ have an equivalent regular grammar. But given in the question both these grammars are neither right linear nor left linear and hence not a regular grammar. So, D must be the answer.

<http://www.cs.odu.edu/~toida/nerzic/390teched/regular/grammar/reg-grammar.html>

1 33 votes

-- Arjun Suresh (350k points)

5.2.20 Context Free Language: GATE2008-IT-34 top

<https://gateoverflow.in/3344>



Selected Answer

$$B \rightarrow 0B00 \mid 1$$

generates $\{0^n 1 0^{2n} \mid n \geq 0\}$

$$\begin{aligned} S &\rightarrow AA, \\ A &\rightarrow 0A \mid A0 \mid 1 \end{aligned}$$

generates $0A0A \rightarrow 00A0A \rightarrow 00101$, which is there in only B and D choices. D is not the answer as "00" is not generated by the given grammar. So, only option left is B and if we see carefully, non-terminal B is generating the second part of B choice and AA is generating the first part.

$$\{0^i 1 0^j 1 0^k \mid i, j, k \geq 0\} \cup \{0^n 1 0^{2n} \mid n \geq 0\}$$

1 22 votes

-- Arjun Suresh (350k points)

5.2.21 Context Free Language: GATE2008-IT-78 top

<https://gateoverflow.in/3392>



$S \rightarrow aS$
 $S \rightarrow aA$
 $S \rightarrow aaAb$
 $S \rightarrow aabAab$
 $S \rightarrow aabbAaab$
 $S \rightarrow aabbaab$

Hence, (D) is the answer.

33 votes

-- Shreyans Dhankhar (2.5k points)

5.2.22 Context Free Language: GATE2009-12, ISRO2016-37 top

<https://gateoverflow.in/1304>



Answer is B. String generated by this language is a,b,aba,bab,aabaa,.....

All this strings are odd length palindromes.

23 votes

-- neha pawar (4.2k points)

5.2.23 Context Free Language: GATE2015-3-32 top

<https://gateoverflow.in/8489>



first check for L_1 . now look a^m & b^m and a^n & b^n must be comparable using one stack for CFL. now take a stack push all a^m in to the stack then push all b^n in to stack now a^n is coming so pop b^n for each a^n by this b^n and a^n will be comparable. now we have left only a^m in stack and b^m is coming so pop a^m for each b^m by which we can compare a^m to b^m ..we conclude that we are comparing this L_1 using a single stack so this is CFG.

now for L_2 .this can not be done in to a single stack because m and n are not comparable we can not find when to push or pop so this is CSL.

now for L_3 .push all a 's into stack and pop 2 a 's for every b and at last we left with a single a . bcz here $aaaaabb$ is a valid string where $m = 2n + 1$ and $n = 2$. So realized using single stack hence L_3 is CFG.

so the option is B.. L_1 and L_3 are CFG

39 votes

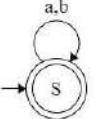
-- Anoop Sonkar (5k points)

5.2.24 Context Free Language: GATE2016-1-16 top

<https://gateoverflow.in/39640>

 Selected Answer

$S \rightarrow aS \mid bS \mid \epsilon$
is $(a+b)^*$



42 votes -- Praveen Saini (54.6k points)

5.2.25 Context Free Language: GATE2016-1-42 [top](#)

<https://gateoverflow.in/39705>

 Selected Answer

Answer is (D).

G_1 results in strings $b, ab, bb, aab, abb, bbb, \dots$ i.e $a^m b^n, m \geq 0$ and $n > 0$ (and because only a 's are not possible but only b 's are)

G_2 result in strings $a, b, aa, ab, bb, aaa, aab, abb, bbb, \dots$ i.e $a^m b^n, m > 0$ or $n > 0$ (or because only b 's is possible $b, bb, bbb, ,$ only a 's are possible)

43 votes -- juxtapose (455 points)

5.2.26 Context Free Language: GATE2016-2-43 [top](#)

<https://gateoverflow.in/39605>

 Selected Answer

$L_1 = \{a^n b^m c^{n+m} : m, n \geq 1\}$ is Context-free language

(push a 's into stack, then push b 's into stack , read c 's and pop b 's , when no b 's left on stack, keep reading c 's and pop a 's , when no c 's left in input , and stack is empty, then accepted).

$L_2 = \{a^n b^n c^{2n} : n \geq 1\}$ is Context-sensitive language and not context-free. (cannot implemented by one stack)

So, answer is option B.

43 votes -- Praveen Saini (54.6k points)

5.2.27 Context Free Language: GATE2017-1-10 [top](#)

<https://gateoverflow.in/118290>

 Selected Answer

Answer should be B

Consider the 1st production $S \rightarrow abScT$

This production generates equal number of (ab) 's and c 's but after each c there is T which goes to

$T \rightarrow bT$

so with each c there can be one or more b 's (one because of production $T \rightarrow b$ and more because of $T \rightarrow bT$)

and these b 's are independent.

For example

$ababcbbcb$ is the part of the language

and $ababcbccccb$ is also the part of the language so we rule out A and C both say equal number of b 's after each c

In option D equal number of (ab) 's and c 's is not satisfied . The only option that satisfies these 2 conditions is B

uddipto's edit:

people who says this will be the string at any case

$S \rightarrow abcT$

$\rightarrow abcb$

no problem for i B in this case also here $cm^n = cm^1$ means n value 1 thats it

and remember in B c is not coming eternally.it is coming until m value reaches n of b..right there

.if $m = n = 1$ then cb stops

29 votes

-- (points)

5.2.28 Context Free Language: GATE2017-1-34 top

<https://gateoverflow.in/118316>



Selected Answer

$S \rightarrow SaS \mid aSb \mid bSa \mid SS \in$

A. $abab$

$S \rightarrow aSb$
 $\rightarrow abSab \rightarrow ab \in ab \rightarrow abab$

B. $aaab$

$S \rightarrow SS$
 $\rightarrow SaSS \rightarrow SaSaSS$
 $\rightarrow aaaSbS \rightarrow aaa \in bS$
 $\rightarrow aaab$

C. $abbaa$

$S \rightarrow SS$
 $\rightarrow aSbS$
 $\rightarrow a \in bS$
 $toabbSa$
 $toabbSaSa$
 $abbaa$

(Putting $S \rightarrow \infty$)

Hence all A,B,C are possible to generate except D. Answer is D.

20 votes

-- Arnabi Bej (7.9k points)

5.2.29 Context Free Language: GATE2017-1-38 top <https://gateoverflow.in/118321>



Answer: Option (A)

1. Here, let's say $L = L_1 \cup L_2$.

Here as we see, L_1 and L_2 both are DCFLs. And hence CFLs.

And from the Closure Properties of CFLs, we can conclude that CFLs are Closed under Union Operation.

And hence, $L = L_1 \cup L_2$ is CFL.

2. Now, let's take $L = L_1 \cap L_2$, and that can be expressed as,

$$L = \{a^i b^j c^k \mid i = j \text{ AND } j = k, i, j, k \geq 0\}, \text{ in other words}$$

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

And we know that it comes under CSL (As we will require Two Stacks at a Time). So L is CSL and also NOT a CFL.

25 votes

-- Tilak D. Nanavati (3.6k points)

5.2.30 Context Free Language: GATE2017-2-16 top <https://gateoverflow.in/118243>



$$S \rightarrow XY$$

$$X \rightarrow aX \mid a$$

$$Y \rightarrow aYb \mid \epsilon$$

X generates atleast one 'a'. While Y generates equal no of a's and b's(including epsilon).

$$L = \{a, ab, aab, aabb, aaabb, \dots\}$$

Hence, answer should be [Option C.](#)

11 votes

-- tvkkk (1.5k points)

5.2.31 Context Free Language: ISI2015-CS-5b top <https://gateoverflow.in/47330>



Given Context free language is :

$$L = \{a^n b^m c^{n+m} : n, m \text{ are integers, } n \geq 1, m \geq 1\}.$$

For clarity given language can be interpreted as

$$L = \{a^n b^m c^m c^n : n, m \text{ are integers, } n \geq 1, m \geq 1\}.$$

Now the given language is clearly **DCFL**.

And context free grammar (CFG) of the following language:

$$S \rightarrow aSc \mid aAc$$

$$A \rightarrow bAc \mid bc$$

11 votes

-- Manoj Kumar (38.6k points)

5.3

Countable Set(1) top

5.3.1 Countable Set: GATE1997-3.4 top

<https://gateoverflow.in/2235>

Given $\Sigma = \{a, b\}$, which one of the following sets is not countable?

- A. Set of all strings over Σ
- B. Set of all languages over Σ
- C. Set of all regular languages over Σ
- D. Set of all languages over Σ accepted by Turing machines

gate1997 theory-of-computation normal countable-set

Answer

Answers: Countable Set

5.3.1 Countable Set: GATE1997-3.4 top

<https://gateoverflow.in/2235>



Selected Answer

(b) Set of all languages over Σ is uncountable.

Ref: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2011/lecture-notes/MIT6_045JS11_lec05.pdf

Power set of an infinite set is uncountable. Set of languages over Σ is the power set of set of strings over Σ which is an infinite set. Hence the set of languages becomes an uncountable set.

21 votes

-- Arjun Suresh (350k points)

5.4

Decidability(23) top

5.4.1 Decidability: GATE1987-2I top

<https://gateoverflow.in/80603>

State whether the following statement are TRUE or FALSE.

A is recursive if both A and its complement are accepted by Turing machines.

gate1987 decidability

Answer

5.4.2 Decidability: GATE1989-3-iii top

<https://gateoverflow.in/87123>

Answer the following questions:

Which of the following problems are undecidable?

- Membership problem in context-free languages.
- Whether a given context-free language is regular.
- Whether a finite state automation halts on all inputs.
- Membership problem for type 0 languages.

gate1989 | normal | theory-of-computation | decidability

Answer

5.4.3 Decidability: GATE1990-3-vii [top](#)

<https://gateoverflow.in/84835>

Choose the correct alternatives (More than one may be correct).

It is undecidable whether:

- An arbitrary Turing machine halts after 100 steps.
- A Turing machine prints a specific letter.
- A Turing machine computes the products of two numbers
- None of the above.

gate1990 | normal | theory-of-computation | decidability

Answer

5.4.4 Decidability: GATE1995-11 [top](#)

<https://gateoverflow.in/2647>

Let L be a language over Σ i.e., $L \subseteq \Sigma^*$. Suppose L satisfies the two conditions given below.

- L is in NP and
- For every n , there is exactly one string of length n that belongs to L .

Let L^c be the complement of L over Σ^* . Show that L^c is also in NP.

gate1995 | theory-of-computation | normal | decidability

Answer

5.4.5 Decidability: GATE1996-1.9 [top](#)

<https://gateoverflow.in/2713>

Which of the following statements is false?

- The Halting Problem of Turing machines is undecidable
- Determining whether a context-free grammar is ambiguous is undecidable
- Given two arbitrary context-free grammars G_1 and G_2 it is undecidable whether $L(G_1) = L(G_2)$
- Given two regular grammars G_1 and G_2 it is undecidable whether $L(G_1) = L(G_2)$

gate1996 | theory-of-computation | decidability | easy

Answer

5.4.6 Decidability: GATE1997-6.5 [top](#)

<https://gateoverflow.in/2261>

Which one of the following is not decidable?

- A. Given a Turing machine M , a string s and an integer k , M accepts s within k steps
- B. Equivalence of two given Turing machines
- C. Language accepted by a given finite state machine is not empty
- D. Language generated by a context free grammar is non-empty

gate1997 theory-of-computation decidability easy

Answer

5.4.7 Decidability: GATE2000-2.9 [top](#)

<https://gateoverflow.in/656>

Consider the following decision problems:

(P1): Does a given finite state machine accept a given string?

(P2): Does a given context free grammar generate an infinite number of strings?

Which of the following statements is true?

- A. Both $(P1)$ and $(P2)$ are decidable
- B. Neither $(P1)$ nor $(P2)$ is decidable
- C. Only $(P1)$ is decidable
- D. Only $(P2)$ is decidable

gate2000 theory-of-computation decidability normal

Answer

5.4.8 Decidability: GATE2001-2.7 [top](#)

<https://gateoverflow.in/725>

Consider the following problem X .

Given a Turing machine M over the input alphabet Σ , any state q of M and a word $w \in \Sigma^*$, does the computation of M on w visit the state of q ?

Which of the following statements about X is correct?

- A. X is decidable
- B. X is undecidable but partially decidable
- C. X is undecidable and not even partially decidable
- D. X is not a decision problem

gate2001 theory-of-computation decidability normal

Answer

5.4.9 Decidability: GATE2003-52 [top](#)

<https://gateoverflow.in/356>

Consider two languages L_1 and L_2 each on the alphabet Σ . Let $f : \Sigma^* \rightarrow \Sigma^*$ be a polynomial time computable bijection such that $(\forall x)[x \in L_1 \text{ iff } f(x) \in L_2]$. Further, let f^{-1} be also polynomial time computable.

Which of the following **CANNOT** be true?

- A. $L_1 \in P$ and L_2 is finite
- B. $L_1 \in NP$ and $L_2 \in P$
- C. L_1 is undecidable and L_2 is decidable
- D. L_1 is recursively enumerable and L_2 is recursive

gate2003 theory-of-computation normal decidability

[Answer](#)

5.4.10 Decidability: GATE2005-45 [top](#)

<https://gateoverflow.in/1375>

- Consider three decision problems P_1 , P_2 and P_3 . It is known that P_1 is decidable and P_2 is undecidable. Which one of the following is TRUE?
- A. P_3 is decidable if P_1 is reducible to P_3
 - B. P_3 is undecidable if P_3 is reducible to P_2
 - C. P_3 is undecidable if P_2 is reducible to P_3
 - D. P_3 is decidable if P_3 is reducible to P_2 's complement

gate2005 theory-of-computation decidability normal

[Answer](#)

5.4.11 Decidability: GATE2007-6 [top](#)

<https://gateoverflow.in/1204>

Which of the following problems is undecidable?

- A. Membership problem for CFGs
- B. Ambiguity problem for CFGs
- C. Finiteness problem for FSAs
- D. Equivalence problem for FSAs

gate2007 theory-of-computation decidability normal

[Answer](#)

5.4.12 Decidability: GATE2008-10 [top](#)

<https://gateoverflow.in/408>

Which of the following are decidable?

- I. Whether the intersection of two regular languages is infinite
- II. Whether a given context-free language is regular
- III. Whether two push-down automata accept the same language
- IV. Whether a given grammar is context-free

- A. I and II
- B. I and IV
- C. II and III
- D. II and IV

gate2008 theory-of-computation decidability easy

[Answer](#)

5.4.13 Decidability: GATE2012-24 [top](#)

<https://gateoverflow.in/1608>

Which of the following problems are decidable?

1. Does a given program ever produce an output?

2. If L is a context-free language, then, is \bar{L} also context-free?
 3. If L is a regular language, then, \bar{L} is also regular?
 4. If L is a recursive language, then, is \bar{L} also recursive?
- A. 1, 2, 3, 4
 B. 1, 2
 C. 2, 3, 4
 D. 3, 4

gate2012 theory-of-computation decidability normal

[Answer](#)

5.4.14 Decidability: GATE2013-41 [top](#)

<https://gateoverflow.in/1553>

Which of the following is/are undecidable?

1. G is a CFG. Is $L(G) = \emptyset$?
 2. G is a CFG. Is $L(G) = \Sigma^*$?
 3. M is a Turing machine. Is $L(M)$ regular?
 4. A is a DFA and N is an NFA. Is $L(A) = L(N)$?
- A. 3 only
 B. 3 and 4 only
 C. 1, 2 and 3 only
 D. 2 and 3 only

gate2013 theory-of-computation decidability normal

[Answer](#)

5.4.15 Decidability: GATE2014-3-35 [top](#)

<https://gateoverflow.in/2069>

Which one of the following problems is undecidable?

- A. Deciding if a given context-free grammar is ambiguous.
 B. Deciding if a given string is generated by a given context-free grammar.
 C. Deciding if the language generated by a given context-free grammar is empty.
 D. Deciding if the language generated by a given context-free grammar is finite.

gate2014-3 theory-of-computation context-free-language decidability normal

[Answer](#)

5.4.16 Decidability: GATE2015-2-21 [top](#)

<https://gateoverflow.in/8111>

Consider the following statements.

- I. The complement of every Turing decidable language is Turing decidable
- II. There exists some language which is in NP but is not Turing decidable
- III. If L is a language in NP, L is Turing decidable

Which of the above statements is/are true?

- A. Only II
 B. Only III
 C. Only I and II
 D. Only I and III

gate2015-2 theory-of-computation decidability easy

Answer

5.4.17 Decidability: GATE2015-3-53 [top](#)

<https://gateoverflow.in/8562>

Language L_1 is polynomial time reducible to language L_2 . Language L_3 is polynomial time reducible to language L_2 , which in turn polynomial time reducible to language L_4 . Which of the following is/are true?

- I. if $L_4 \in P$, then $L_2 \in P$
 - II. if $L_1 \in P$ or $L_3 \in P$, then $L_2 \in P$
 - III. $L_1 \in P$, if and only if $L_3 \in P$
 - IV. if $L_4 \in P$, then $L_3 \in P$
- A. II only
 B. III only
 C. I and IV only
 D. I only

gate2015-3 theory-of-computation decidability normal

Answer

5.4.18 Decidability: GATE2016-1-17 [top](#)

<https://gateoverflow.in/39651>

Which of the following decision problems are undecidable?

- I. Given NFAs N_1 and N_2 , is $L(N_1) \cap L(N_2) = \Phi$
 - II. Given a CFG $G = (N, \Sigma, P, S)$ and a string $x \in \Sigma^*$, does $x \in L(G)$?
 - III. Given CFGs G_1 and G_2 , is $L(G_1) = L(G_2)$?
 - IV. Given a TM M , is $L(M) = \Phi$?
- A. I and IV only
 B. II and III only
 C. III and IV only
 D. II and IV only

gate2016-1 theory-of-computation decidability easy

Answer

5.4.19 Decidability: GATE2017-1-39 [top](#)

<https://gateoverflow.in/118322>

Let A and B be finite alphabets and let $\#$ be a symbol outside both A and B . Let f be a total function from A^* to B^* . We say f is *computable* if there exists a Turing machine M which given an input $x \in A^*$, always halts with $f(x)$ on its tape. Let L_f denote the language $\{x\#f(x) \mid x \in A^*\}$. Which of the following statements is true:

- A. f is computable if and only if L_f is recursive.
- B. f is computable if and only if L_f is recursively enumerable.
- C. If f is computable then L_f is recursive, but not conversely.
- D. If f is computable then L_f is recursively enumerable, but not conversely.

gate2017-1 theory-of-computation decidability difficult

Answer

5.4.20 Decidability: GATE2017-2-41 top

<https://gateoverflow.in/118605>

Let $L(R)$ be the language represented by regular expression R . Let $L(G)$ be the language generated by a context free grammar G . Let $L(M)$ be the language accepted by a Turing machine M . Which of the following decision problems are undecidable?

- I. Given a regular expression R and a string w , is $w \in L(R)$?
 - II. Given a context-free grammar G , is $L(G) = \emptyset$?
 - III. Given a context-free grammar G , is $L(G) = \Sigma^*$ for some alphabet Σ ?
 - IV. Given a Turing machine M and a string w , is $w \in L(M)$?
- A. I and IV only
 B. II and III only
 C. II, III and IV only
 D. III and IV only

gate2017-2 theory-of-computation decidability

Answer

5.4.21 Decidability: GATE2018-36 top

<https://gateoverflow.in/204110>

Consider the following problems. $L(G)$ denotes the language generated by a grammar G . $L(M)$ denotes the language accepted by a machine M .

- I. For an unrestricted grammar G and a string w , whether $w \in L(G)$
- II. Given a Turing machine M , whether $L(M)$ is regular
- III. Given two grammars G_1 and G_2 , whether $L(G_1) = L(G_2)$
- IV. Given an NFA N , whether there is a deterministic PDA P such that N and P accept the same language

Which one of the following statement is correct?

- A. Only I and II are undecidable
 B. Only II is undecidable
 C. Only II and IV are undecidable
 D. Only I, II and III are undecidable

gate2018 theory-of-computation decidability easy

Answer

5.4.22 Decidability: TIFR2010-B-25 top

<https://gateoverflow.in/18745>

Which of the following problems is decidable? (Here, CFG means context free grammar and CFL means context free language.)

- A. Give a CFG G , find whether $L(G) = R$, where R is regular set.
- B. Given a CFG G , find whether $L(G) = \{\}$.
- C. Find whether the intersection of two CFLs is empty.
- D. Find whether the complement of CFL is a CFL.
- E. Find whether CFG G_1 and CFG G_2 generate the same language, i.e., $L(G_1) = L(G_2)$.

tifr2010 theory-of-computation context-free-language decidability

Answer

5.4.23 Decidability: TIFR2011-B-25 top

<https://gateoverflow.in/20404>

Let A_{TM} be defined as follows:

$$A_{TM} = \{\langle M, w \rangle \mid \text{The Turning machine } M \text{ accepts the word } w\}$$

And let L be some **NP**- complete language. Which of the following statements is FALSE?

- A. $L \in \mathbf{NP}$
- B. Every problem in **NP** is polynomial time reducible to L .
- C. Every problem in **NP** is polynomial time reducible to A_{TM} .
- D. Since L is **NP**- complete, A_{TM} is polynomial time reducible to L .
- E. $A_{TM} \notin \mathbf{NP}$.

tifr2011 theory-of-computation decidability

Answer

Answers: Decidability

5.4.1 Decidability: GATE1987-2I top

<https://gateoverflow.in/80603>



Yes if A and its complement are accepted by Turing machines then A is recursive.

Suppose a language A is recursively enumerable. That means there exists a Turing machine $T1$ that, given any string of the language, halts and accepts that string.

Now let's also suppose that the complement of A , $A' = \{w : w \mid A\}$, is recursively enumerable. That means there is some other Turing machine $T2$ that, given any string of A' halts and accepts that string. So any string belongs to either A or A' . Hence, any string will cause either $T1$ or $T2$ (or both) to halt. We construct a new Turing machine that emulates both $T1$ and $T2$, alternating moves between them. When either one stops, we can tell (by whether it accepted or rejected the string) to which language the string belongs.

Thus, we have constructed a Turing machine that, for each input, halts with an answer whether or not the string belongs to A' . Therefore A and A' are recursive languages.

14 votes

-- Prashant Singh (59.8k points)

5.4.2 Decidability: GATE1989-3-iii top

<https://gateoverflow.in/87123>



1. Membership problem in context-free languages. is **Decidable**.
2. Whether a given context-free language is regular. **Undecidable** [Regularity is decidable till **DCFL**]
3. Whether a finite state automation halts on all inputs. **Decidable**
4. Membership problem for type 0 languages. **Undecidable** [undecidable for RE or semidecidable]

Reff : <http://gatecse.in/grammar-decidable-and-undecidable-problems/>

15 votes

-- Prashant Singh (59.8k points)

5.4.3 Decidability: GATE1990-3-vii top

<https://gateoverflow.in/84835>



Selected Answer

- A. An arbitrary Turing machine halts after 100 steps. **DECIDABLE**, we can run TM for 100 steps and conclude that. (assuming "after 100 steps" mean "exactly after 100 steps")
- B. A Turing machine prints a specific letter. **UNDECIDABLE**,
- C. A Turing machine computes the products of two numbers, **UNDECIDABLE**, Even though we can design a TM for calculation product of 2 numbers but here it is asking whether given TM computes product of 2 numbers, so the behaviour of TM unknown hence, Undecidable.

17 votes

-- Prajwal Bhat (11.3k points)

5.4.4 Decidability: GATE1995-11 top

<https://gateoverflow.in/2647>



Selected Answer

Since L is in NP it is decidable (recursive) and so is its complement L^c . Now, L^c may or may not be in NP. But we are given that for any string length n , exactly one string belongs to L , which means for any string length all but one string belongs to L^c .

Now, definition of NP says that all "yes" instances of the problem can be solved in polynomial time using a nondeterministic TM. So, given an instance of $\langle L^c, x \rangle$, we non-deterministically take all words of length n , where n is the length of w , and see if it is in L . As soon as we get the word (such a word is sure to exist as exactly one word of length n belongs to L), we see if this word is same as x . If it is not same (and only if it is not same), $x \in L^c$ and we get this answer in polynomial time making L^c an NP problem.

11 votes

-- Arjun Suresh (350k points)

5.4.5 Decidability: GATE1996-1.9 top

<https://gateoverflow.in/2713>



Selected Answer

Answer is D.

Equivalence of Regular languages is decidable.

1. Membership,
2. Emptiness,
3. Finiteness,
4. Equivalence,
5. Ambiguity,
6. Regularity,
7. Everything,
8. Disjointedness...

All are decidable for Regular languages.

First 3 for CFL.

Only 1st for CSL and REC.

None for RE.

16 votes

-- Gate Keeda (19.6k points)

5.4.6 Decidability: GATE1997-6.5 top

<https://gateoverflow.in/2261>

Selected Answer

Option B. Equivalence of two TMs is undecidable.

Option (A) is not halting problem. In Halting problem no. of steps can go up to infinity and that is the only reason why it becomes undecidable. In (A) the number of steps is restricted to a finite number ' k ' and simulating a TM for ' k ' steps is trivially decidable because we just go to step k and output the answer.

For **Options (C)** and **(D)** we do have well defined algorithms making them decidable.

21 votes

-- Bhagirathi Nayak (14.1k points)

5.4.7 Decidability: GATE2000-2.9 top

<https://gateoverflow.in/656>

Selected Answer

For P_1 , we just need to give a run on the machine. Finite state machines always halts unlike TM.

For P_2 , check if the CFG generates any string of length between n and $2n - 1$, where n is the pumping lemma constant. If So, $L(CFG)$ is infinite, else finite. Finding the pumping lemma constant is not trivial - but there are other procedures which can do this -
<http://cs.stackexchange.com/questions/52507/is-it-decidable-whether-a-given-context-free-grammar-generates-an-infinite-number/52520>

Hence, both P_1 and P_2 are decidable - answer is (A).

http://gatcse.in/wiki/Grammar:_Decidable_and_Undecidable_Problems

20 votes

-- Arjun Suresh (350k points)

5.4.8 Decidability: GATE2001-2.7 top

<https://gateoverflow.in/725>

Selected Answer

X is undecidable but partially decidable.

We have the TM M . Just make the state q the final state and make all other final states non-final and get a new TM M' . Give input w to M' . If w would have taken M to state q (yes case of the problem), our new TM M' would accept it. So, the given problem is partially decidable.

If M goes for an infinite loop and never reaches state q (no case for the problem), M' cannot output anything. This problem is the state entry problem, which like word accepting problem and halting problem is undecidable.

40 votes

-- Arjun Suresh (350k points)

5.4.9 Decidability: GATE2003-52 top

<https://gateoverflow.in/356>



Since, f is a polynomial time computable bijection and f^{-1} is also polynomial time computable, L_1 and L_2 should have the same complexity (isomorphic). This is because, given a problem for L_1 , we can always do a polynomial time reduction to L_2 and vice versa. Hence, the answer is 'C', as in 'A', L_1 and L_2 can be finite, in 'B', L_1 and L_2 can be in P and in 'D', L_1 and L_2 can be recursive. Only, in 'C' there is no intersection for L_1 and L_2 , and hence it can't be true.

Alternatively, we can prove 'C' to be false as follows:

Given L_2 is decidable. Now, for a problem in L_1 , we can have a TM , which takes an input x , calculates $f(x)$ in polynomial time, check $f(x)$ is in L_2 (this is decidable as L_2 is decidable), and if it is, then output yes and otherwise no. Thus L_1 must also be decidable.

20 votes

-- gatecse (18k points)

5.4.10 Decidability: GATE2005-45 top

<https://gateoverflow.in/1375>



- A. If P_1 is reducible to P_3 , then P_3 is at least as hard as P_1 . So, no guarantee if P_3 is decidable.
- B. If P_3 is reducible to P_2 , then P_3 cannot be harder than P_2 . But P_2 being undecidable, this can't say P_3 is undecidable.
- C. If P_2 is reducible to P_3 , then P_3 is at least as hard as P_2 . Since, P_2 is undecidable, this means P_3 is also undecidable -**hence the answer**.
- D. Complement of an undecidable problem is undecidable. Now, reducing to an undecidable problem can't prove P_3 is decidable.

http://gatecse.in/wiki/Some_Reduction_Inferences

33 votes

-- Arjun Suresh (350k points)

5.4.11 Decidability: GATE2007-6 top

<https://gateoverflow.in/1204>



- Membership problem is decidable as it can be solved by parsers.
- Finiteness problem is decidable for FSAs (also for CFGs), as we just need to check for a loop in the DFA.
- Equivalence problem for FSAs is decidable as we can take the complement of one FSA (complement of FSA is another FSA), and do an intersection with the other (FSAs are closed under intersection also), giving a new FSA. If this new FSA accept no string, then the given FSAs are equivalent, else not equivalent.
- Only ambiguity problem for CFGs are undecidable.

http://gatecse.in/wiki/Grammar:_Decidable_and_Undecidable_Problems

20 votes

-- Arjun Suresh (350k points)

5.4.12 Decidability: GATE2008-10 top

<https://gateoverflow.in/408>



Lets see options one by one :

- I. The language here will be regular as intersection of regular languages will lead to regular language only..And we know that given a regular language , whether it is finite or not is a decidable problem..This can be seen by observing the DFA..If DFA contains a state which contains a loop and that state is reachable from start state and that state is either a final state or leading to final state , then the language will be infinite..
- II. The regularity property is undecidable for context free languages..Hence it is undecidable..Details regarding this : <https://cs.stackexchange.com/questions/19482/why-is-deciding-regularity-of-a-context-free-language-undecidable>
- III. Now equivalence of two CFLs is also undecidable property..Hence given 2 PDAs which is nothing but characterising CFLs , whether the 2 CFLs will be same or not cannot be decided..
- IV. Given a grammar , it is context free iff its productions are of the type $V \rightarrow (V \cup \Sigma)^*$ Hence it is a decidable property..

Hence, B) should be correct answer..

15 votes

-- HABIB MOHAMMAD KHAN (98.7k points)

5.4.13 Decidability: GATE2012-24 top

<https://gateoverflow.in/1608>



CFL's are not closed under complementation and a program can loop forever. So, it may not produce any output.

Regular and recursive languages are closed under complementation.

Hence, only **3,4** are decidable.

24 votes

-- Bhagirathi Nayak (14.1k points)

5.4.14 Decidability: GATE2013-41 top

<https://gateoverflow.in/1553>



It will be **D**.

1. First is Emptiness for CFG.
2. Second is everything for CFG.
3. Third is Regularity for REC
4. Fourth is equivalence for regular.

26 votes

-- Gate Keeda (19.6k points)

5.4.15 Decidability: GATE2014-3-35 top

<https://gateoverflow.in/2069>



(A) is the answer. Proving (A) is undecidable is not so easy. But we can easily prove the other

three options given here are decidable.

<https://gatecse.in/grammar-decidable-and-undecidable-problems/>

20 votes

-- Arjun Suresh (350k points)

5.4.16 Decidability: GATE2015-2-21 top

<https://gateoverflow.in/8111>



Selected Answer

- I. is true. The solution to a decision problem is either "yes" or "no", and hence if we can decide a problem, we have also decided its complement- just reverse "yes" and "no". (This is applicable for decidability and not for acceptance)
- II. is false. Because NP class is defined as the class of languages that can be solved in polynomial time by a non-deterministic Turing machine. So, none of the NP class problems is undecidable.
- III. is true for same reason as II.

So, answer is **D**.

37 votes

-- Arjun Suresh (350k points)

5.4.17 Decidability: GATE2015-3-53 top

<https://gateoverflow.in/8562>



Selected Answer

1. L_1 is polynomial time reducible to L_2 . So, L_2 is at least as hard as L_1 .
2. L_3 is polynomial time reducible to L_2 . So, L_2 is at least as hard as L_3 .
3. L_2 is polynomial time reducible to L_4 . So, L_4 is at least as hard as L_2 .

If L_4 is in P , L_3 , L_2 and L_1 must also be in P . So, I and IV are true.

We can have L_1 in P and L_2 not in P , and none of the given conditions are violated. So, II is false.

Assume L_3 not in P . Now, Since L_2 must be at least as hard as L_3 , it must also be not in P . But L_1 is less harder than L_1 as per condition 1, and it can be in P without violating any given conditions. So, III is false.

Hence, **C** is choice.

More Info: http://gatecse.in/wiki/Some_Reduction_Inferences

24 votes

-- Arjun Suresh (350k points)

5.4.18 Decidability: GATE2016-1-17 top

<https://gateoverflow.in/39651>



Selected Answer

- I. is Decidable, we may use cross product of NFA (or by converting them into DFA) , if We didn't get final states of both together at any state in it. then $L(N_1) \cap L(N_2) = \emptyset$, Disjoint languages.
- II. Membership in CFG is Decidable (CYK algorithm)

III. Equivalence of Two context free grammars is Undecidable.

IV. For TM M , $L(M) = \phi$ is Undecidable.

44 votes

-- Praveen Saini (54.6k points)

5.4.19 Decidability: GATE2017-1-39 top

<https://gateoverflow.in/118322>



Selected Answer

f being computable, given x , we get $f(x)$.

L_f is recursive - given any string we get to know if the string is in L or not.

Now lets see the two cases:

- Does f being computable implies L_f is recursive?

We are given x and we need to know if $x\#f(x)$ belongs to L_f . Since f being computable we can calculate $f(x)$ and our problem reduces to 3 string comparisons - first x , followed by # and then $f(x)$ which can be done by a TM. So, L_f must be recursive.

- Does L_f being recursive implies f be computable?

If L_f is recursive, given any string we can say whether it belongs to L_f or not. Now, to compute $f(x)$, we can do a dovetailing approach starting with strings in lexicographic (or any other order) as follows s_1, s_2, s_3, \dots and forming inputs to the TM as

$x\#s_1, x\#s_2, x\#s_3, \dots$

By dovetailing it means to simulation TM for one step in first iteration, 2 steps in second iteration, 3 steps in third iteration etc., and in each iteration adding a new string to the simulation (TM is parallelly simulating all strings).

https://courses.engr.illinois.edu/cs373/sp2009/lectures/lect_24.pdf

Now, for any x , we should have an $f(x)$ since f is a total function. So, in our simulation we are sure that sometime we will add the string $x\#f(x)$ to the simulation and then the TM will halt for it. And whenever it does halt, we can just take the part of the input string after #, and that is $f(x)$. We have a way to compute $f(x)$ for any x . So, f is computable. The same method would work even if L_f is recursively enumerable and not necessarily recursive.

So, the correct statements are

1. If L_f is recursively enumerable f is computable
2. If f is computable L_f is recursive

So, options A and B are TRUE but official key was only A.

<https://cs.stackexchange.com/questions/70626/recursive-language-and-computable-function>

28 votes

-- Arjun Suresh (350k points)

5.4.20 Decidability: GATE2017-2-41 top

<https://gateoverflow.in/118605>



Selected Answer

1st statement is Membership problem of regular language = **decidable**

2nd statement is Emptiness problem of *CFL* = **decidable**

3rd statement is accept everthing problem of *CFL*= **undecidable**

4th statement is Membership problem of *RE* language = **undecidable**

D is answer.

16 votes

-- Prashant Singh (59.8k points)

5.4.21 Decidability: GATE2018-36 top

<https://gateoverflow.in/204110>



Selected Answer

Membership of Unrestricted grammar is undecidable.

Turing machine may not halt so undecidable.

Two grammars are equal is undecidable

PDA (accepting CFLs) having more power than NFA (accepting regular) so we can decide whether both will accept the same language or not.

13 votes

-- Digvijay (54.9k points)

5.4.22 Decidability: TIFR2010-B-25 top

<https://gateoverflow.in/18745>



Selected Answer

- A. We don't have any standard algorithm to change *CFG* into *CFL* from a given *CFG* deciding a language is finite is decidable but regular its undecidable (Ref: http://gatecse.in/wiki/Grammar:_Decidable_and_Undecidable_Problems)
- B. From a given given *CFG* we can determine the *CFL* and *CFL* emptiness is Decidable.
- C. Intersection of two *CFL* is undecidable coz it is not closed under intersection.
- D. *CFL* is not closed under Complement so its undecidable.
- E. *CFL* is not closed under equivalence so it is undecidable to compare 2 language.

Therefore, **B** is decidable and **A,C,D** and **E** are undecidable.

10 votes

-- Umang Raman (15.7k points)

5.4.23 Decidability: TIFR2011-B-25 top

<https://gateoverflow.in/20404>



Selected Answer

A_{TM} is the language of the Halting Problem. It is undecidable, but Recursively Enumerable.

L is **NPC**

- a. True. Any language in **NPC** is also in **NP** by definition.
- b. True. By definition, any problem in **NP** is polynomial time reducible to any **NPC** problem.
- c. True. A_{TM} is undecidable. Any language that is decidable is polynomial time reducible to A_{TM} !
- d. **False.**

A_{TM} is undecidable. No Turing Machine can guarantee an answer in a finite time, let alone a polynomial time.

- e. True. A_{TM} is undecidable. It is certainly not in **NP**.

Hence, the correct answer is option d.

14 votes

-- Pragy Agarwal (20.6k points)

5.5

Finite Automata(65) top

5.5.1 Finite Automata: CMI2010-B-04a top

<https://gateoverflow.in/46127>

Indicate whether the following statements are true or false, providing a short explanation to substantiate your answers.

- i. A DFA with n states must accept at least one string of length greater than n .

[cmi2010](#) [theory-of-computation](#) [finite-automata](#) [descriptive](#)

Answer

5.5.2 Finite Automata: CMI2010-B-04b top

<https://gateoverflow.in/47073>

Indicate whether the following statement is true or false, providing a short explanation to substantiate your answers.

A DFA that has n states and accepts an infinite language must accept at least one string x such that $2n < |x| < 3n$, where $|x|$ denotes the length of x .

[descriptive](#) [cmi2010](#) [finite-automata](#)

Answer

5.5.3 Finite Automata: CMI2010-B-04c top

<https://gateoverflow.in/47074>

Indicate whether the following statement is true or false, providing a short explanation to substantiate your answers.

If a language L is accepted by an NFA with n states then there is a DFA with no more than 2^n states accepting L .

[descriptive](#) [cmi2010](#) [finite-automata](#)

Answer

5.5.4 Finite Automata: CMI2012-B-02a top

<https://gateoverflow.in/46562>

For a binary string $x = a_0a_1\dots a_{n-1}$ define $val(x)$ to be $\sum_{0 \leq i < n} 2^{n-1-i} \cdot a_i$. Let $\Sigma = \{(0,0), (0,1), (1,0), (1,1)\}$.

- a. Construct a finite automaton that accepts the set of all strings $(a_0, b_0)(a_1, b_1)\dots(a_{n-1}, b_{n-1}) \in \Sigma^*$ such that $val(b_0b_1\dots b_{n-1}) = 2 \cdot val(a_0a_1\dots a_{n-1})$.

[cmi2012](#) [descriptive](#) [theory-of-computation](#) [finite-automata](#)

Answer

5.5.5 Finite Automata: CMI2012-B-02b top

<https://gateoverflow.in/47091>

For a binary string $x = a_0a_1\dots a_{n-1}$ define $\text{val}(x)$ to be $\sum_{0 \leq i < n} 2^{n-1-i} \cdot a_i$. Let $\Sigma = \{(0,0), (0,1), (1,0), (1,1)\}$.

- A. Construct a finite automaton that accepts exactly those strings $(a_0, b_0)(a_1, b_1)\dots(a_{n-1}, b_{n-1}) \in \Sigma^*$ such that $\text{val}(b_0b_1\dots b_{n-1}) = 4 \cdot \text{val}(a_0a_1\dots a_{n-1})$.

[descriptive](#) [cmi2012](#) [theory-of-computation](#) [finite-automata](#)

[Answer](#)

5.5.6 Finite Automata: GATE1987-2j [top](#)

<https://gateoverflow.in/80594>

State whether the following statements are TRUE or FALSE:

A minimal DFA that is equivalent to an NDFA with n nodes has always 2^n states.

[gate1987](#) [theory-of-computation](#) [finite-automata](#)

[Answer](#)

5.5.7 Finite Automata: GATE1991-17,b [top](#)

<https://gateoverflow.in/544>

Let L be the language of all binary strings in which the third symbol from the right is a 1. Give a non-deterministic finite automaton that recognizes L . How many states does the minimized equivalent deterministic finite automaton have? Justify your answer briefly?

[gate1991](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

[Answer](#)

5.5.8 Finite Automata: GATE1993-27 [top](#)

<https://gateoverflow.in/2323>

Draw the state transition of a deterministic finite state automaton which accepts all strings from the alphabet $\{a, b\}$, such that no string has 3 consecutive occurrences of the letter b .

[gate1993](#) [theory-of-computation](#) [finite-automata](#) [easy](#)

[Answer](#)

5.5.9 Finite Automata: GATE1994-3.3 [top](#)

<https://gateoverflow.in/2480>

State True or False with one line explanation

A FSM (Finite State Machine) can be designed to add two integers of any arbitrary length (arbitrary number of digits).

[gate1994](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

[Answer](#)

5.5.10 Finite Automata: GATE1995-2.23 [top](#)

<https://gateoverflow.in/2636>

A finite state machine with the following state table has a single input x and a single output z .

present state	next state, z	
	$x = 1$	$x = 0$
A	$D, 0$	$B, 0$
B	$B, 1$	$C, 1$
C	$B, 0$	$D, 1$
D	$B, 1$	$C, 0$

If the initial state is unknown, then the shortest input sequence to reach the final state C is:

- A. 01
- B. 10
- C. 101
- D. 110

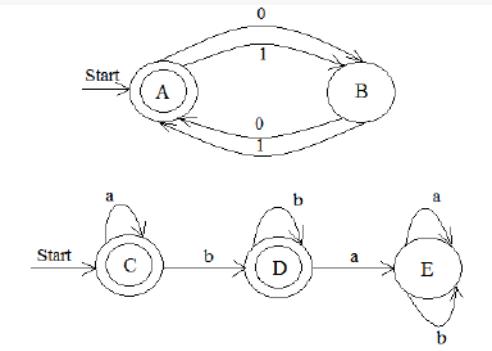
gate1995 theory-of-computation finite-automata normal

Answer

5.5.11 Finite Automata: GATE1996-12 top

<https://gateoverflow.in/2764>

Given below are the transition diagrams for two finite state machines M_1 and M_2 recognizing languages L_1 and L_2 respectively.



- A. Display the transition diagram for a machine that recognizes $L_1 \cdot L_2$, obtained from transition diagrams for M_1 and M_2 by adding only ϵ transitions and no new states.
 - B. Modify the transition diagram obtained in part (a) obtain a transition diagram for a machine that recognizes $(L_1 \cdot L_2)^*$ by adding only ϵ transitions and no new states.
- (Final states are enclosed in double circles).

gate1996 theory-of-computation finite-automata normal

Answer

5.5.12 Finite Automata: GATE1996-2.23 top

<https://gateoverflow.in/2752>

Consider the following state table for a sequential machine. The number of states in the minimized machine will be

		Input	
		0	1
Present State	A	D, 0	B, 1
	B	A, 0	C, 1
	C	A, 0	B, 1
	D	A, 1	C, 1
		Next state, Output	

- A. 4
- B. 3
- C. 2
- D. 1

gate1996 theory-of-computation normal finite-automata

Answer

5.5.13 Finite Automata: GATE1997-20 [top](#)

<https://gateoverflow.in/2280>

Construct a finite state machine with minimum number of states, accepting all strings over (a, b) such that the number of a 's is divisible by two and the number of b 's is divisible by three.

gate1997 theory-of-computation finite-automata normal

Answer

5.5.14 Finite Automata: GATE1997-21 [top](#)

<https://gateoverflow.in/2281>

Given that L is a language accepted by a finite state machine, show that L^P and L^R are also accepted by some finite state machines, where

$$L^P = \{s \mid ss' \in L \text{ some string } s'\}$$

$$L^R = \{s \mid s \text{ obtained by reversing some string in } L\}$$

gate1997 theory-of-computation finite-automata proof

Answer

5.5.15 Finite Automata: GATE1998-1.10 [top](#)

<https://gateoverflow.in/1647>

Which of the following set can be recognized by a Deterministic Finite state Automaton?

- A. The numbers $1, 2, 4, 8, \dots, 2^n, \dots$ written in binary
- B. The numbers $1, 2, 4, 8, \dots, 2^n, \dots$ written in unary
- C. The set of binary string in which the number of zeros is the same as the number of ones.
- D. The set $\{1, 101, 11011, 1110111, \dots\}$

gate1998 theory-of-computation finite-automata normal

Answer

5.5.16 Finite Automata: GATE1998-2.5 [top](#)

<https://gateoverflow.in/1677>

Let L be the set of all binary strings whose last two symbols are the same. The number of states in the minimal state deterministic finite state automaton accepting L is

- A. 2
- B. 5
- C. 8
- D. 3

gate1998 theory-of-computation finite-automata normal

Answer

5.5.17 Finite Automata: GATE1998-4 [top](#)

<https://gateoverflow.in/1695>

Design a deterministic finite state automaton (using minimum number of states) that recognizes the following language:

$L = \{w \in \{0,1\}^* \mid w \text{ interpreted as binary number (ignoring the leading zeros) is divisible by five}\}.$

gate1998 theory-of-computation finite-automata normal

Answer

5.5.18 Finite Automata: GATE1999-1.4 top

<https://gateoverflow.in/1458>

Consider the regular expression $(0+1)(0+1)\dots N$ times. The minimum state finite automaton that recognizes the language represented by this regular expression contains

- A. n states
- B. $n+1$ states
- C. $n+2$ states
- D. None of the above

gate1998 theory-of-computation finite-automata easy

Answer

5.5.19 Finite Automata: GATE2001-1.6 top

<https://gateoverflow.in/699>

Given an arbitrary non-deterministic finite automaton (NFA) with N states, the maximum number of states in an equivalent minimized DFA at least

- A. N^2
- B. 2^N
- C. $2N$
- D. $N!$

gate2001 Finite-automata theory-of-computation easy

Answer

5.5.20 Finite Automata: GATE2001-2.5 top

<https://gateoverflow.in/723>

Consider a DFA over $\Sigma = \{a,b\}$ accepting all strings which have number of a 's divisible by 6 and number of b 's divisible by 8. What is the minimum number of states that the DFA will have?

- A. 8
- B. 14
- C. 15
- D. 48

gate2001 theory-of-computation finite-automata

Answer

5.5.21 Finite Automata: GATE2001-5 top

<https://gateoverflow.in/746>

Construct DFA's for the following languages:

- A. $L = \{w \mid w \in \{a,b\}^*, w \text{ has baab as a substring}\}$
- B. $L = \{w \mid w \in \{a,b\}^*, w \text{ has an odd number of a's and an odd number of b's}\}$

gate2001 theory-of-computation easy descriptive finite-automata normal

Answer

5.5.22 Finite Automata: GATE2002-2.13 top

<https://gateoverflow.in/843>

The smallest finite automaton which accepts the language $\{x \mid \text{length of } x \text{ is divisible by 3}\}$ has

- A. 2 states
- B. 3 states
- C. 4 states
- D. 5 states

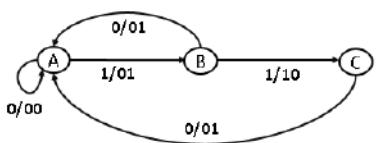
gate2002 theory-of-computation normal finite-automata

Answer

5.5.23 Finite Automata: GATE2002-2.5 top

<https://gateoverflow.in/835>

The finite state machine described by the following state diagram with A as starting state, where an arc label is and x stands for 1-bit input and y stands for 2-bit output



- A. outputs the sum of the present and the previous bits of the input
- B. outputs 01 whenever the input sequence contains 11
- C. outputs 00 whenever the input sequence contains 10
- D. none of the above

gate2002 theory-of-computation normal finite-automata

Answer

5.5.24 Finite Automata: GATE2002-21 top

<https://gateoverflow.in/874>

We require a four state automaton to recognize the regular expression $(a/b)^*abb$

- A. Give an NFA for this purpose
- B. Give a DFA for this purpose

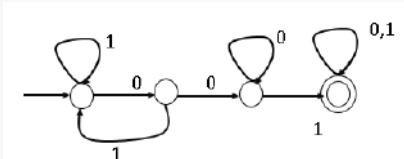
gate2002 theory-of-computation finite-automata normal descriptive

Answer

5.5.25 Finite Automata: GATE2003-50 top

<https://gateoverflow.in/939>

Consider the following deterministic finite state automaton M .



Let S denote the set of seven bit binary strings in which the first, the fourth, and the last bits are 1. The number of strings in S that are accepted by M is

- A. 1
B. 5
C. 7
D. 8

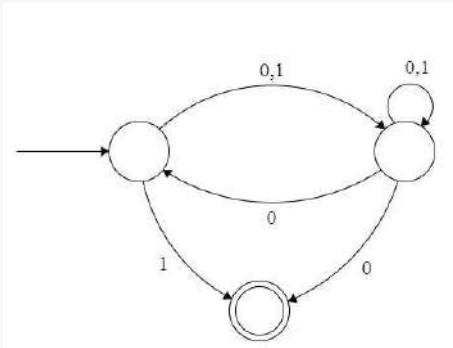
gate2003 theory-of-computation finite-automata normal

Answer

5.5.26 Finite Automata: GATE2003-55 top

<https://gateoverflow.in/943>

Consider the NFA M shown below.



Let the language accepted by M be L . Let L_1 be the language accepted by the NFA M_1 obtained by changing the accepting state of M to a non-accepting state and by changing the non-accepting states of M to accepting states. Which of the following statements is true?

- A. $L_1 = \{0, 1\}^* - L$
 B. $L_1 = \{0, 1\}^*$
 C. $L_1 \subseteq L$
 D. $L_1 = L$

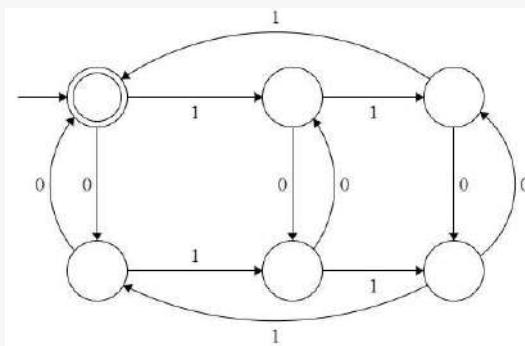
gate2003 theory-of-computation finite-automata normal

Answer

5.5.27 Finite Automata: GATE2004-86 top

<https://gateoverflow.in/1080>

The following finite state machine accepts all those binary strings in which the number of 1's and 0's are respectively:



- A. divisible by 3 and 2
 B. odd and even

- C. even and odd
D. divisible by 2 and 3

gate2004 theory-of-computation finite-automata easy

Answer

5.5.28 Finite Automata: GATE2004-IT-41 [top](#)

<https://gateoverflow.in/3684>

Let $M = (K, \Sigma, \sigma, s, F)$ be a finite state automaton, where

$$K = \{A, B\}, \Sigma = \{a, b\}, s = A, F = \{B\}, \\ \sigma(A, a) = A, \sigma(A, b) = B, \sigma(B, a) = B \text{ and } \sigma(B, b) = A$$

A grammar to generate the language accepted by M can be specified as $G = (V, \Sigma, R, S)$, where $V = K \cup \Sigma$, and $S = A$.

Which one of the following set of rules will make $L(G) = L(M)$?

- A. $\{A \rightarrow aB, A \rightarrow bA, B \rightarrow bA, B \rightarrow aA, B \rightarrow \epsilon\}$
 B. $\{A \rightarrow aA, A \rightarrow bB, B \rightarrow aB, B \rightarrow bA, B \rightarrow \epsilon\}$
 C. $\{A \rightarrow bB, A \rightarrow aB, B \rightarrow aA, B \rightarrow bA, B \rightarrow \epsilon\}$
 D. $\{A \rightarrow aA, A \rightarrow bA, B \rightarrow aB, B \rightarrow bA, A \rightarrow \epsilon\}$

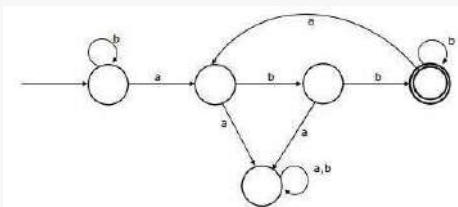
gate2004-it theory-of-computation finite-automata normal

Answer

5.5.29 Finite Automata: GATE2005-53 [top](#)

<https://gateoverflow.in/1376>

Consider the machine M :



The language recognized by M is:

- A. $\{w \in \{a, b\}^* \mid \text{every } a \text{ in } w \text{ is followed by exactly two } b's\}$
 B. $\{w \in \{a, b\}^* \mid \text{every } a \text{ in } w \text{ is followed by at least two } b's\}$
 C. $\{w \in \{a, b\}^* \mid w \text{ contains the substring 'abb'}\}$
 D. $\{w \in \{a, b\}^* \mid w \text{ does not contain 'aa' as a substring}\}$

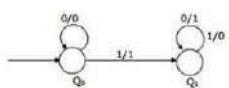
gate2005 theory-of-computation finite-automata normal

Answer

5.5.30 Finite Automata: GATE2005-63 [top](#)

<https://gateoverflow.in/1386>

The following diagram represents a finite state machine which takes as input a binary number from the least significant bit.



Which of the following is TRUE?

- A. It computes 1's complement of the input number
- B. It computes 2's complement of the input number
- C. It increments the input number
- D. it decrements the input number

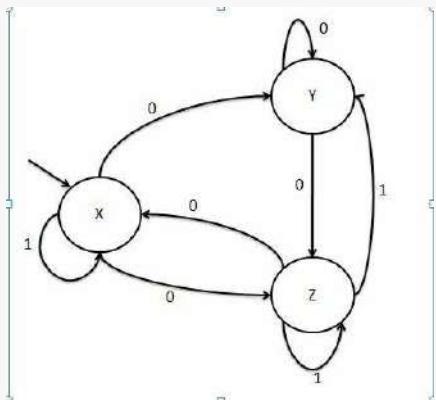
gate2005 theory-of-computation finite-automata easy

Answer

5.5.31 Finite Automata: GATE2005-IT-37 [top](#)

<https://gateoverflow.in/3784>

Consider the non-deterministic finite automaton (NFA) shown in the figure.



State X is the starting state of the automaton. Let the language accepted by the NFA with Y as the only accepting state be L_1 . Similarly, let the language accepted by the NFA with Z as the only accepting state be L_2 . Which of the following statements about L_1 and L_2 is TRUE?

- A. $L_1 = L_2$
- B. $L_1 \subset L_2$
- C. $L_2 \subset L_1$
- D. None of the above

gate2005-it theory-of-computation finite-automata normal

Answer

5.5.32 Finite Automata: GATE2005-IT-39 [top](#)

<https://gateoverflow.in/3786>

Consider the regular grammar:

$$\begin{aligned} S &\rightarrow Xa \mid Ya \\ X &\rightarrow Za \\ Z &\rightarrow Sa \mid \epsilon \\ Y &\rightarrow Wa \\ W &\rightarrow Sa \end{aligned}$$

where S is the starting symbol, the set of terminals is $\{a\}$ and the set of non-terminals is

$\{S, W, X, Y, Z\}$.

We wish to construct a deterministic finite automaton (DFA) to recognize the same language. What is the minimum number of states required for the DFA?

- A. 2
- B. 3
- C. 4
- D. 5

gate2005-it theory-of-computation finite-automata normal

Answer

5.5.33 Finite Automata: GATE2006-34 [top](#)

<https://gateoverflow.in/1291>

Consider the regular language $L = (111 + 11111)^*$. The minimum number of states in any DFA accepting this language is:

- A. 3
- B. 5
- C. 8
- D. 9

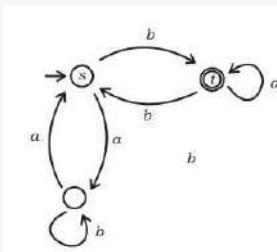
gate2006 theory-of-computation finite-automata normal

Answer

5.5.34 Finite Automata: GATE2006-IT-3 [top](#)

<https://gateoverflow.in/3542>

In the automaton below, s is the start state and t is the only final state.



Consider the strings $u = abba, v = bab$, and $w = aabb$. Which of the following statements is true?

- A. The automaton accepts u and v but not w
- B. The automaton accepts each of u, v , and w
- C. The automaton rejects each of u, v , and w
- D. The automaton accepts u but rejects v and w

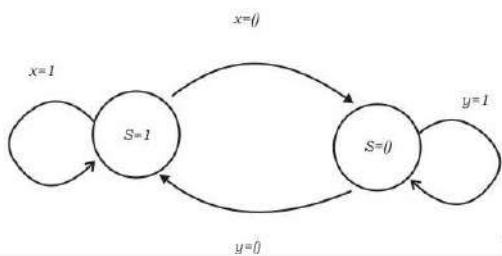
gate2006-it theory-of-computation finite-automata normal

Answer

5.5.35 Finite Automata: GATE2006-IT-37 [top](#)

<https://gateoverflow.in/3576>

For a state machine with the following state diagram the expression for the next state S^+ in terms of the current state S and the input variables x and y is



- A. $S^+ = S'.y' + S.x$
- B. $S^+ = S.x.y' + S'.y.x'$
- C. $S^+ = x.y'$
- D. $S^+ = S'.y + S.x'$

gate2006-it theory-of-computation finite-automata normal

[Answer](#)

5.5.36 Finite Automata: GATE2007-29 [top](#)

<https://gateoverflow.in/1227>

A minimum state deterministic finite automaton accepting the language

$L = \{w \mid w \in \{0,1\}^*, \text{ number of } 0\text{s and } 1\text{s in } w \text{ are divisible by } 3 \text{ and } 5, \text{ respectively}\}$ has

- A. 15 states
- B. 11 states
- C. 10 states
- D. 9 states

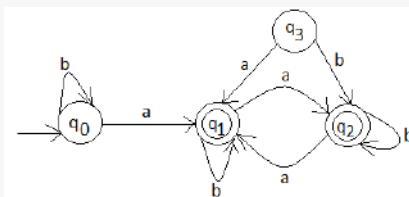
gate2007 theory-of-computation finite-automata normal

[Answer](#)

5.5.37 Finite Automata: GATE2007-74 [top](#)

<https://gateoverflow.in/1270>

Consider the following Finite State Automaton:



The language accepted by this automaton is given by the regular expression

- A. $b^*ab^*ab^*ab^*$
- B. $(a+b)^*$
- C. $b^*a(a+b)^*$
- D. $b^*ab^*ab^*$

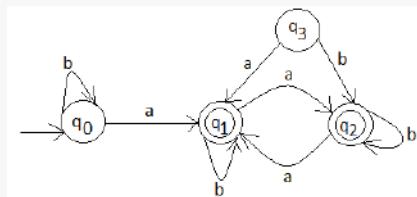
gate2007 theory-of-computation finite-automata normal

[Answer](#)

5.5.38 Finite Automata: GATE2007-75 [top](#)

<https://gateoverflow.in/43514>

Consider the following Finite State Automaton:



The minimum state automaton equivalent to the above FSA has the following number of states:

- A. 1
- B. 2
- C. 3
- D. 4

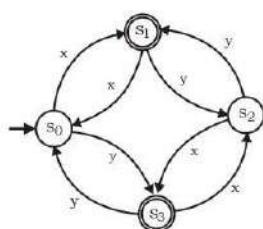
[normal](#) [gate2007](#) [theory-of-computation](#) [finite-automata](#)

Answer

5.5.39 Finite Automata: GATE2007-IT-47 [top](#)

<https://gateoverflow.in/3489>

Consider the following DFA in which S_0 is the start state and S_1, S_3 are the final states.



What language does this DFA recognize?

- A. All strings of x and y
- B. All strings of x and y which have either even number of x and even number of y or odd number of x and odd number of y
- C. All strings of x and y which have equal number of x and y
- D. All strings of x and y with either even number of x and odd number of y or odd number of x and even number of y

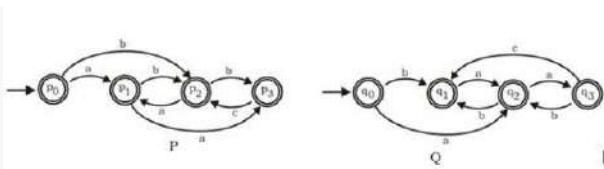
[gate2007-it](#) [theory-of-computation](#) [Finite-automata](#) [normal](#)

Answer

5.5.40 Finite Automata: GATE2007-IT-50 [top](#)

<https://gateoverflow.in/3492>

Consider the following finite automata P and Q over the alphabet $\{a, b, c\}$. The start states are indicated by a double arrow and final states are indicated by a double circle. Let the languages recognized by them be denoted by $L(P)$ and $L(Q)$ respectively.



The automation which recognizes the language $L(P) \cap L(Q)$ is :

- Detailed description of Figure 2:

 - A.** State r_0 has a self-loop labeled 'a'. Transitions from r_0 to r_1 and r_2 are both labeled 'a'. From r_1 , a transition to r_2 is labeled 'b', and a transition back to r_0 is labeled 'a'. From r_2 , a transition to r_3 is labeled 'c', and a transition back to r_1 is labeled 'a'. A self-loop on r_3 is labeled 'b'.
 - B.** State r_0 has a self-loop labeled 'a'. Transitions from r_0 to r_1 and r_2 are both labeled 'a'. From r_1 , a transition to r_2 is labeled 'b', and a transition back to r_0 is labeled 'b'. From r_2 , a transition to r_3 is labeled 'c', and a transition back to r_1 is labeled 'a'. A self-loop on r_3 is labeled 'b'.
 - C.** State r_0 has a self-loop labeled 'a'. Transitions from r_0 to r_1 and r_2 are both labeled 'a'. From r_1 , a transition to r_2 is labeled 'b', and a transition back to r_0 is labeled 'a'. From r_2 , a transition to r_3 is labeled 'b', and a transition back to r_1 is labeled 'a'. A self-loop on r_3 is labeled 'b'.
 - D.** State r_0 has a self-loop labeled 'a'. Transitions from r_0 to r_1 and r_2 are both labeled 'a'. From r_1 , a transition to r_2 is labeled 'b', and a transition back to r_0 is labeled 'a'. From r_2 , a transition to r_3 is labeled 'b', and a transition back to r_1 is labeled 'a'. A self-loop on r_3 is labeled 'b'.

gate2007-it theory-of-computation finite-automata normal

Answer

5.5.41 Finite Automata: GATE2007-IT-71 top

<https://gateoverflow.in/3523>

Consider the regular expression $R = (a + b)^*(aa + bb)(a + b)^*$

Which of the following non-deterministic finite automata recognizes the language defined by the regular expression R ? Edges labeled λ denote transitions on the empty string.

- Detailed description of Figure 3:

 - A.** The start state is s_0 . Transitions from s_0 to s_1 , s_2 , and s_3 are labeled "a,b". State s_1 has a self-loop labeled "a". State s_2 has a self-loop labeled "b". State s_3 has a self-loop labeled "a".
 - B.** The start state is s_0 . Transitions from s_0 to s_1 , s_2 , and s_3 are labeled "a,b". State s_1 has a self-loop labeled "a". State s_2 has a self-loop labeled "b". State s_3 has a self-loop labeled "a".
 - C.** The start state is s_0 . Transitions from s_0 to s_1 , s_2 , and s_3 are labeled "a,b". State s_1 has a self-loop labeled "a". State s_2 has a self-loop labeled "b". State s_3 has a self-loop labeled "a".
 - D.** The start state is s_0 . Transitions from s_0 to s_1 , s_2 , and s_3 are labeled "a,b". State s_1 has a self-loop labeled "a". State s_2 has a self-loop labeled "b". State s_3 has a self-loop labeled "a".

gate2007-it theory-of-computation finite-automata normal

Answer

5.5.42 Finite Automata: GATE2007-IT-72 top

<https://gateoverflow.in/3524>

Consider the regular expression $R = (a + b)^*(aa + bb)(a + b)^*$

Which deterministic finite automaton accepts the language represented by the regular expression R ?

- A.
-
- ```

graph LR
 S0((s0)) -- a --> S1((s1))
 S0 -- b --> S2((s2))
 S1 -- a --> S3((s3))
 S1 -- b --> S2
 S2 -- a --> S4((s4))
 S2 -- b --> S3
 S3 -- a --> S3
 S3 -- b --> S4
 S4 -- a --> S4

```
- B.
- 
- ```

graph LR
    S0((s0)) -- a --> S1((s1))
    S0 -- b --> S2((s2))
    S1 -- a --> S3((s3))
    S1 -- b --> S2
    S2 -- a --> S4((s4))
    S2 -- b --> S3
    S3 -- a --> S3
    S3 -- b --> S4
    S4 -- a --> S4
  
```
- C.
-
- ```

graph LR
 S0((s0)) -- a --> S1((s1))
 S0 -- b --> S2((s2))
 S1 -- a --> S3((s3))
 S1 -- b --> S2
 S2 -- a --> S3
 S2 -- b --> S1
 S3 -- a --> S3
 S3 -- b --> S2

```
- D.
- 
- ```

graph LR
    S0((s0)) -- a --> S1((s1))
    S0 -- b --> S2((s2))
    S1 -- a --> S3((s3))
    S1 -- b --> S2
    S2 -- a --> S3
    S2 -- b --> S1
    S3 -- a --> S3
    S3 -- b --> S1
  
```

gate2007-it theory-of-computation Finite-automata normal

Answer

5.5.43 Finite Automata: GATE2008-49 top

<https://gateoverflow.in/462>

Given below are two finite state automata (\rightarrow indicates the start state and F indicates a final state)

Y	a	b
$\rightarrow 1$	1	2
$2(F)$	2	1

Z	a	b
$\rightarrow 1$	2	2
$2(F)$	1	1

Which of the following represents the product automaton $Z \times Y$?

A.

	a	b
--	---	---

$\rightarrow P$	S	R
Q	R	S
$R(F)$	Q	P
S	Q	P

B.

	a	b
$\rightarrow P$	S	Q
Q	R	S
$R(F)$	Q	P
S	P	Q

C.

	a	b
$\rightarrow P$	Q	S
Q	R	S
$R(F)$	Q	P
S	Q	P

D.

	a	b
$\rightarrow P$	S	Q
Q	S	R
$R(F)$	Q	P
S	Q	P

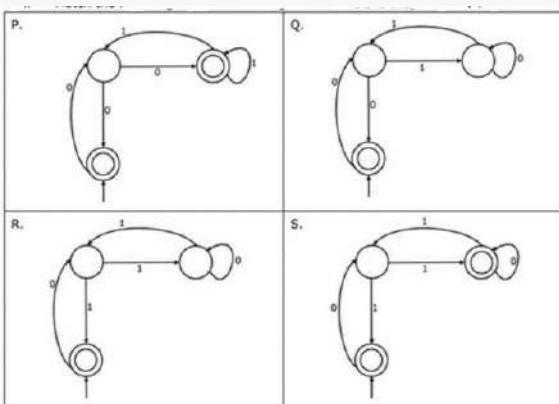
gate2008 normal theory-of-computation finite-automata

Answer

5.5.44 Finite Automata: GATE2008-52 top

<https://gateoverflow.in/464>

Match the following NFAs with the regular expressions they correspond to:



1. $\epsilon + 0(01^*1 + 00)^*01^*$

2. $\epsilon + 0(10^*1 + 00)^*0$

3. $\epsilon + 0(10^*1 + 10)^*1$

4. $\epsilon + 0(10^*1 + 10)^*10^*$

- A. $P - 2, Q - 1, R - 3, S - 4$
 B. $P - 1, Q - 3, R - 2, S - 4$

- C. $P = 1, Q = 2, R = 3, S = 4$
 D. $P = 3, Q = 2, R = 1, S = 4$

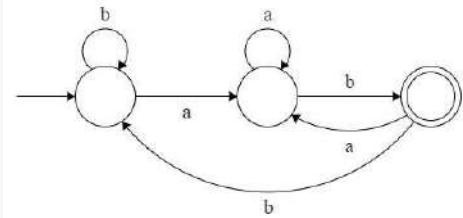
gate2008 theory-of-computation finite-automata normal

Answer

5.5.45 Finite Automata: GATE2008-IT-32 top

<https://gateoverflow.in/3342>

If the final states and non-final states in the DFA below are interchanged, then which of the following languages over the alphabet $\{a, b\}$ will be accepted by the new DFA?



- A. Set of all strings that do not end with ab
- B. Set of all strings that begin with either an a or $a b$
- C. Set of all strings that do not contain the substring ab ,
- D. The set described by the regular expression $b^*aa^*(ba)^*b^*$

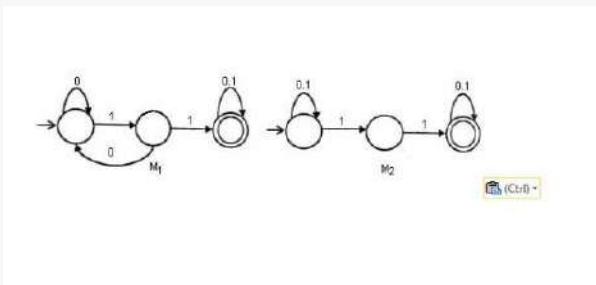
gate2008-it theory-of-computation finite-automata normal

Answer

5.5.46 Finite Automata: GATE2008-IT-36 top

<https://gateoverflow.in/3346>

Consider the following two finite automata. M_1 accepts L_1 and M_2 accepts L_2 . Which one of the following is TRUE?



- A. $L_1 = L_2$
- B. $L_1 \subset L_2$
- C. $L_1 \cap L_2' = \emptyset$
- D. $L_1 \cup L_2 \neq L_1$

gate2008-it theory-of-computation finite-automata normal

Answer

5.5.47 Finite Automata: GATE2008-IT-6 top

<https://gateoverflow.in/3266>

Let N be an NFA with n states and let M be the minimized DFA with m states recognizing the same language. Which of the following is NECESSARILY true?

- A. $m \leq 2^n$

- B. $n \leq m$
 C. M has one accept state
 D. $m = 2^n$

gate2008-it theory-of-computation finite-automata normal

Answer

5.5.48 Finite Automata: GATE2009-27 top

<https://gateoverflow.in/1313>

Given the following state table of an FSM with two states A and B , one input and one output.

PRESENT STATE A	PRESENT STATE B	Input	Next State A	Next State B	Output
0	0	0	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	1	0	0
0	0	1	0	1	0
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	0	0	1

If the initial state is $A = 0, B = 0$ what is the minimum length of an input string which will take the machine to the state $A = 0, B = 1$ with $output = 1$.

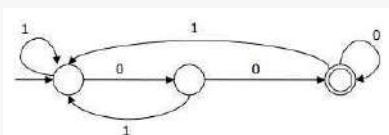
- A. 3
 B. 4
 C. 5
 D. 6

gate2009 theory-of-computation finite-automata normal

Answer

5.5.49 Finite Automata: GATE2009-41 top

<https://gateoverflow.in/1327>



The above DFA accepts the set of all strings over $\{0,1\}$ that

- A. begin either with 0 or 1.
- B. end with 0.
- C. end with 00.
- D. contain the substring 00.

gate2009 theory-of-computation finite-automata easy

Answer

5.5.50 Finite Automata: GATE2010-41 [top](#)

<https://gateoverflow.in/2342>

Let w be any string of length n in $\{0,1\}^*$. Let L be the set of all substrings of w . What is the minimum number of states in non-deterministic finite automaton that accepts L ?

- A. $n - 1$
- B. n
- C. $n + 1$
- D. 2^{n-1}

gate2010 theory-of-computation finite-automata normal

Answer

5.5.51 Finite Automata: GATE2011-42 [top](#)

<https://gateoverflow.in/2144>

Definition of a language L with alphabet $\{a\}$ is given as following.

$$L = \{a^{nk} \mid k > 0, \text{ and } n \text{ is a positive integer constant}\}$$

What is the minimum number of states needed in a DFA to recognize L ?

- A. $k + 1$
- B. $n + 1$
- C. 2^{n+1}
- D. 2^{k+1}

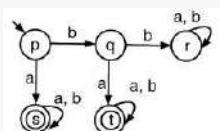
gate2011 theory-of-computation finite-automata normal

Answer

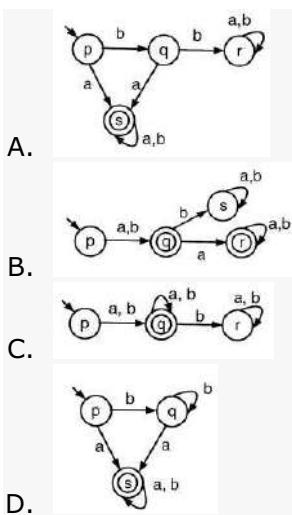
5.5.52 Finite Automata: GATE2011-45 [top](#)

<https://gateoverflow.in/2147>

A deterministic finite automaton (DFA) D with alphabet $\Sigma = \{a,b\}$ is given below.



Which of the following finite state machines is a valid minimal DFA which accepts the same language as D ?


[gate2011](#) [theory-of-computation](#) [finite-automata](#) [easy](#)
Answer

5.5.53 Finite Automata: GATE2012-12 top

<https://gateoverflow.in/44>

What is the complement of the language accepted by the NFA shown below?
Assume $\Sigma = \{a\}$ and ϵ is the empty string.

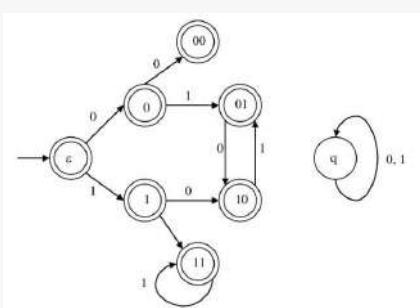
- A. \emptyset
- B. $\{\epsilon\}$
- C. a^*
- D. $\{a, \epsilon\}$

[gate2012](#) [finite-automata](#) [easy](#) [theory-of-computation](#)
Answer

5.5.54 Finite Automata: GATE2012-46 top

<https://gateoverflow.in/2159>

Consider the set of strings on $\{0, 1\}$ in which, every substring of 3 symbols has at most two zeros. For example, 001110 and 011001 are in the language, but 100010 is not. All strings of length less than 3 are also in the language. A partially completed DFA that accepts this language is shown below.



The missing arcs in the DFA are:

A.

	00	01	10	11	q
00	1	0			
01			1		
10	0				
11		0			

B.

	00	01	10	11	q
00		0			1
01		1			
10			0		
11		0			

C.

	00	01	10	11	q
00		1			0
01		1			
10			0		
11		0			

D.

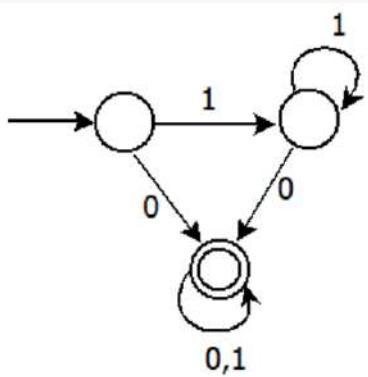
	00	01	10	11	q
00		1			0
01			1		
10	0				
11		0			

gate2012 theory-of-computation finite-automata normal

Answer

5.5.55 Finite Automata: GATE2013-33 top<https://gateoverflow.in/1544>

Consider the DFA A given below.

Which of the following are **FALSE**?

1. Complement of $L(A)$ is context-free.
 2. $L(A) = L((11^*0 + 0)(0 + 1)^*0^*1^*)$
 3. For the language accepted by A , A is the minimal DFA.
 4. A accepts all strings over $\{0, 1\}$ of length at least 2.
- A. 1 and 3 only
 B. 2 and 4 only
 C. 2 and 3 only
 D. 3 and 4 only

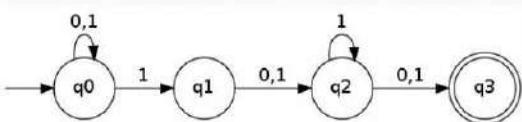
gate2013 theory-of-computation finite-automata normal

Answer

5.5.56 Finite Automata: GATE2014-1-16 [top](#)

<https://gateoverflow.in/1782>

Consider the finite automaton in the following figure:



What is the set of reachable states for the input string 0011?

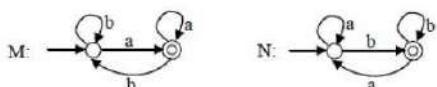
- A. $\{q_0, q_1, q_2\}$
 B. $\{q_0, q_1\}$
 C. $\{q_0, q_1, q_2, q_3\}$
 D. $\{q_3\}$

gate2014-1 theory-of-computation finite-automata easy

Answer

5.5.57 Finite Automata: GATE2015-1-52 [top](#)

<https://gateoverflow.in/8362>



Consider the DFAs M and N given above. The number of states in a minimal DFA that accept the language $L(M) \cap L(N)$ is_____.

gate2015-1 theory-of-computation finite-automata easy numerical-answers

Answer

5.5.58 Finite Automata: GATE2015-2-53 [top](#)

<https://gateoverflow.in/8256>

The number of states in the minimal deterministic finite automaton corresponding to the regular expression $(0 + 1)^*(10)$ is_____.

gate2015-2 theory-of-computation finite-automata normal numerical-answers

Answer

5.5.59 Finite Automata: GATE2015-3-18 [top](#)

<https://gateoverflow.in/8415>

Let L be the language represented by the regular expression $\Sigma^*0011\Sigma^*$ where $\Sigma = \{0,1\}$. What is the minimum number of states in a DFA that recognizes \bar{L} (complement of $\$L4$)?

- A. 4
- B. 5
- C. 6
- D. 8

gate2015-3 theory-of-computation finite-automata normal

[Answer](#)

5.5.60 Finite Automata: GATE2016-2-16 [top](#)

<https://gateoverflow.in/39562>

The number of states in the minimum sized DFA that accepts the language defined by the regular expression.

$$(0+1)^*(0+1)(0+1)^*$$

is _____.

gate2016-2 theory-of-computation finite-automata normal numerical-answers

[Answer](#)

5.5.61 Finite Automata: GATE2016-2-42 [top](#)

<https://gateoverflow.in/39591>

Consider the following two statements:

- I. If all states of an **NFA** are accepting states then the language accepted by the **NFA** is Σ^* .
- II. There exists a regular language A such that for all languages B , $A \cap B$ is regular.

Which one of the following is **CORRECT**?

- A. Only I is true
- B. Only II is true
- C. Both I and II are true
- D. Both I and II are false

gate2016-2 theory-of-computation finite-automata normal

[Answer](#)

5.5.62 Finite Automata: GATE2017-1-22 [top](#)

<https://gateoverflow.in/118302>

Consider the language L given by the regular expression $(a+b)^*b(a+b)$ over the alphabet $\{a,b\}$. The smallest number of states needed in a deterministic finite-state automaton (DFA) accepting L is _____.

gate2017-1 theory-of-computation finite-automata numerical-answers

[Answer](#)

5.5.63 Finite Automata: GATE2017-2-25 [top](#)

<https://gateoverflow.in/118160>

The minimum possible number of states of a deterministic finite automaton that accepts the regular language $L = \{w_1aw_2 \mid w_1, w_2 \in \{a,b\}^*, |w_1| = 2, |w_2| \geq 3\}$ is _____.

theory-of-computation gate2017-2 finite-automata numerical-answers

[Answer](#)

5.5.64 Finite Automata: GATE2017-2-39 [top](#)

<https://gateoverflow.in/118384>

Let δ denote the transition function and $\hat{\delta}$ denote the extended transition function of the ϵ -NFA whose transition table is given below:

δ	ϵ	a	b
$\rightarrow q_0$	$\{q_2\}$	$\{q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_3\}$
$\{q_2\}$	$\{q_0\}$	\emptyset	\emptyset
q_3	\emptyset	\emptyset	$\{q_2\}$

Then $\hat{\delta}(q_2, aba)$ is

- A. \emptyset
- B. $\{q_0, q_1, q_3\}$
- C. $\{q_0, q_1, q_2\}$
- D. $\{q_0, q_2, q_3\}$

[gate2017-2](#) [theory-of-computation](#) [finite-automata](#)

[Answer](#)

5.5.65 Finite Automata: ISI2014-CS-4a [top](#)

<https://gateoverflow.in/47443>

Construct a deterministic finite automaton accepting the following language:
 $\{w \in \{0,1\}^*: w \text{ has an equal number of } 01's \text{ and } 10's\}$. For example, 101 is in the language because it contains one instance of 10 and one instance of 01 as well.

[descriptive](#) [isi2014](#) [theory-of-computation](#) [finite-automata](#)

[Answer](#)

Answers: Finite Automata

5.5.1 Finite Automata: CMI2010-B-04a [top](#)

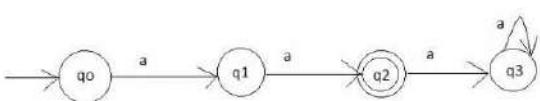
<https://gateoverflow.in/46127>

1) No. If it has limited number of string or finite string, then it is not possible

Say, the DFA accepts aa

Number of states to accept aa is 4

So, here no string with length 4

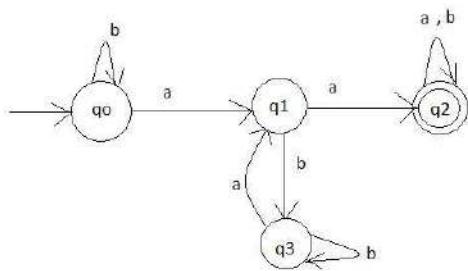


2) True. As it is infinite language it can accept any length string

$L = \{waaw \mid w \in \{a,b\}^*\}$

Say DFA accepts $(a+b)^*aa(a+b)^*$

DFA has 4 states, So string may contain of length 9, which is possible here:



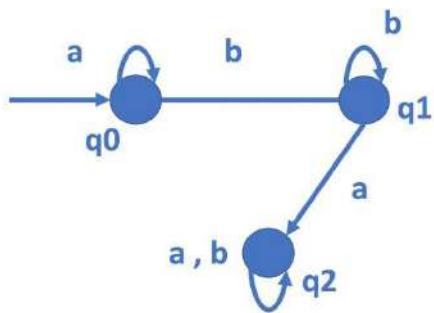
3) True

2 votes

-- srestha (87k points)

5.5.2 Finite Automata: CMI2010-B-04b top

<https://gateoverflow.in/47073>



Consider above DFA in which:

initial state = q_0

final states = $\{q_0, q_1\}$

trap state = q_2

Language generated by above DFA is a^*b^* which is an infinite language

$n = 3$ (number of states)

now consider string $X = \{aaabbbb\}$ i.e., a^3b^4

$|X| = 7$

now $2n = 2 * 3 = 6$

$3n = 3 * 3 = 9$

$2n < |x| < 3n \Rightarrow 6 < 7 < 9$

Hence, it is proved.

7 votes

-- Tauhin Gangwar (9.9k points)

5.5.3 Finite Automata: CMI2010-B-04c top

<https://gateoverflow.in/47074>



Selected Answer

If a language L is accepted by an NFA with n states then there is a DFA with no more than 2^n states accepting L .

This is correct statement. With [Proof.](#)

7 votes

-- Manoj Kumar (38,6k points)

5.5.4 Finite Automata: CMI2012-B-02a [top](#)

<https://gateoverflow.in/46562>



Selected Answer

Given that for a binary string $x = a_0 a_1 a_2 \dots a_{n-1}$

$$val(x) = \sum_{i=0}^{n-1} 2^{n-1-i} \cdot a_i$$

Here, $val(x)$ is the decimal equivalent of the binary number x .

For example,

$$x = 1010$$

$$\begin{aligned} val(x) &= 2^3 \cdot 1 + 2^2 \cdot 0 + 2^1 \cdot 1 + 2^0 \cdot 0 \\ &= 8 + 2 = 10 \end{aligned}$$

They specified that $\Sigma = \{(0,0)(0,1)(1,0)(1,1)\}$

That is every pair in the input $(a_0, b_0)(a_1, b_1) \dots (a_{n-1}, b_{n-1})$ will take any one of these 4 possibilities.

Our task is to construct a Finite Automaton for the input such that

$$val(b_0 b_1 \dots b_{n-1}) = 2 \cdot val(a_0 a_1 \dots a_{n-1})$$

Consider an input $a_0 \ b_0 \ a_1 \ b_1 \ a_2 \ b_2 \ a_3 \ b_3$
 $(0,1) \ (1,1) \ (1,1) \ (1,0) \ (0,0)$

$$val(a_0 a_1 a_2 a_3) = val(0110) = 6$$

$$val(b_0 b_1 b_2 b_3) = val(1100) = 12$$

Here, $val(b_0 b_1 b_2 b_3) = 2 \cdot val(a_0 a_1 a_2 a_3)$

So the input $(0,1)(1,1)(1,0)(0,0)$ should have to be accepted by the Finite Automaton.

Idea of constructing DFA:

Note : the DFA should reject the input starting with 1.

Ex: $(1,1)(0,1)(1,1)(0,0)$

Reason: It is not possible for the bit sequence $b_0 b_1 \dots b_{n-1}$, to represent a value equal to twice the value of the sequence $a_0 a_1 \dots a_{n-1}$ if $a_0 = 1$.

If $a_0 = 1$, the number of bits required to represent the value equal to twice the value of $a_0a_1\dots a_{n-1}$ is $n+1$.

Ex: Let $n = 4$ and $a_0a_1a_2a_3 = (1000) = 8$

Number of bits required to represent 16 is 5 but the number of bits available is $b_0 b_1 b_2 b_3$ which is 4.

So, the input should not start with 1.

Consider the example,

$$\begin{array}{cccc} (0,1) & (1,1) & (1,0) & (0,0) \\ a_0 & b_0 & a_1 & b_1 \\ & & a_2 & b_2 \\ & & a_3 & b_3 \end{array}$$

$$(a_0a_1a_2a_3) = (0110) = 6$$

$$(b_0 b_1 b_2 b_3) = (1100) = 12$$

Here we can clearly see that

$$b_0 = a_1$$

$$b_1 = a_2$$

$$b_2 = a_3$$

$$b_3 = 0$$

This condition can be generalized as,

$$b_0 = a_1$$

$$b_1 = a_2$$

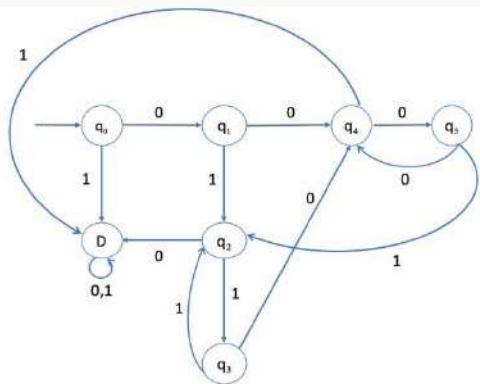
$$b_2 = a_3$$

$$b_{n-2} = a_{n-1}$$

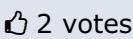
$$b_{n-1} = 0$$

It is our task to construct the DFA satisfying the above condition.

This is the corresponding DFA:



Regular expression is : $0.(00 + 11)^*0$



-- **balaeinstein** (1,3k points)

5.5.5 Finite Automata: CMI2012-B-02b top

<https://gateoverflow.in/47091>



1

Before analysing this problem , please have a look at my answer for the problem :
<https://gateoverflow.in/46562/cmi2012-b-02a> .

This problem is a slight modification to the problem that I mentioned above. But the idea of construction of FA completely differs for this problem.

Analysing the input string:

Consider the input: 01011010

$$(a_0 a_1 a_2 a_3) = 0011$$

$$(b_0 b_1 b_2 b_3) = 1100$$

$$\text{val}(b_0 b_1 b_2 b_3) = 4 \cdot \text{val}(a_0 a_1 a_2 a_3)$$

So this input is one of the strings belonging to the language.

For any string belonging to this language, the least significant 2 bits of $(b_0 b_1 b_2 b_3 \dots b_{n-1})$ should be zeros because $\text{val}(b_0 b_1 b_2 b_3) = 4 \cdot \text{val}(a_0 a_1 a_2 a_3)$. [Any bit sequence which is a multiple of 4 will end with 00]

The 2 most significant bit positions of $(a_0 a_1 a_2 a_3 \dots a_{n-1})$ should be zeros so only there exists a possibility for $(b_0 b_1 b_2 \dots b_{n-1})$ to be 4 times the value of $(a_0 a_1 a_2 a_3 \dots a_{n-1})$.

These are the 2 basic requirements for the input to be a part of the language.

Idea of construction of Finite Automaton :

Consider the input 01011010

$$(a_0 a_1 a_2 a_3) = 0011$$

$$(b_0 b_1 b_2 b_3) = 1100$$

We can clearly see from the above example that

$$a_0 = a_1 = 0$$

$$b_0 = a_2$$

$$b_1 = a_3.$$

$$b_2 = b_3 = 0$$

For any input in the language,

$$a_0 = a_1 = 0$$

$$b_0 = a_2$$

$$b_1 = a_3$$

$$b_2 = a_4$$

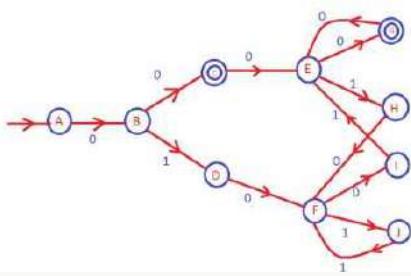
.....

$$b_{n-3} = a_{n-1}$$

$$b_{n-2} = b_{n-1} = 0$$

So, this input pattern should have to be recognized by the Finite automaton.

The corresponding Finite automaton is:



Regular Expression : $00 + (000+010(11)^*01).(00+10(11)^*01)^*0$

1 votes

-- balainstein (1.3k points)

5.5.6 Finite Automata: GATE1987-2j top

<https://gateoverflow.in/80594>

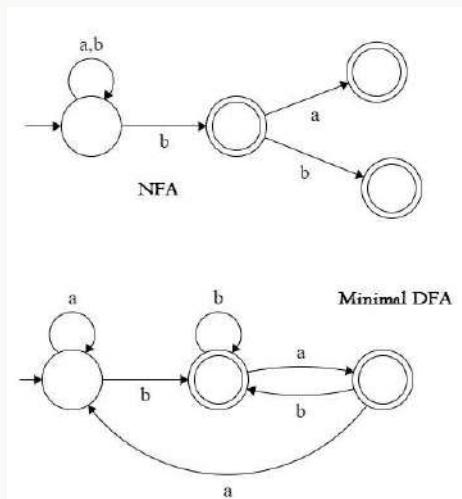


Selected Answer

No a minimal DFA that is equivalent to an NDFA with n nodes has always 2^n states.

Correct statement is A minimal DFA that is equivalent to an NDFA with n nodes has atmost 2^n states.

Example :



17 votes

-- Prashant Singh (59.8k points)

5.5.7 Finite Automata: GATE1991-17,b top

<https://gateoverflow.in/544>



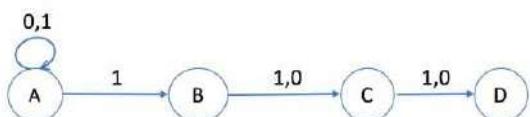
Selected Answer

Answer:

Check following NFA. I've done subset construction too. 8 States are needed even after minimization..

Every state containing D is final state.

NFA:

**NFA to DFA**

	0	1
A	A	AB
AB	AC	ABC
AC	AD	ABD
AD	A	AB
ABC	ACD	ABCD
ABD	AC	ABC
ACD	AD	ABC
ABCD	ACD	ABCD

Third symbol from the right is a '1'. So, we can also consider Myhill-Nerode theorem here. Intuitively we need to remember the last 3 bits of the string each of which forms a different equivalence class as per Myhill-Nerode theorem as shown by the following table. Here, for any set of strings (in a row), we distinguishes only the rows above it - as the relation is symmetric. Further strings in the language and not in the language are distinguished separately as ϵ distinguishes them.

Last 3 bits	Distinguishing string	In the language?
1000		N
2001	"00" - distinguishes from strings in 1.	N
3010	"0" - distinguishes from strings in 1 and 2. "00" distinguishes from strings in 4.	N
4011	"0" - distinguishes from strings in 1 and 2. "00" distinguishes from strings in 3.	N
5100		Y
6101	"00" distinguishes from strings in 5.	Y
7110	"0" distinguishes from strings in 5 and "00" distinguishes from strings in 6	Y
8111	"00" distinguishes from strings in 5 and 7. "0" distinguishes from strings in 6.	Y

15 votes

-- Akash Kanase (42.5k points)

5.5.8 Finite Automata: GATE1993-27 top

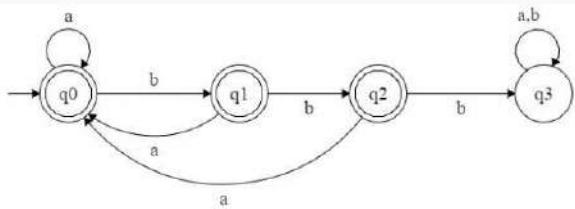
<https://gateoverflow.in/2323>

Selected Answer

Design a DFA that accepts all strings contain bbb

regular expression $(a + b)^*bbb(a + b)^*$

then take complement of DFA such that no string has 3 consecutive occurrences of the letter b .



having regular expression $(a + ba + bba)^*(\epsilon + b + bb)$

 27 votes

-- Praveen Saini (54.6k points)

5.5.9 Finite Automata: GATE1994-3.3 top

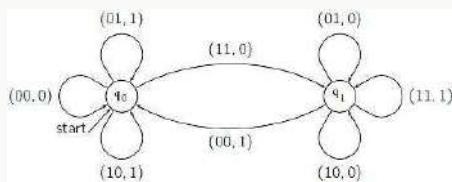
<https://gateoverflow.in/2480>

Selected Answer

Finite Automata (FA) or Finite State Machine to add two integers can be constructed using two states:

- q_0 : Start state to represent carry bit is 0
- q_1 : State to represent carry bit is 1

The inputs to FA will be pair of bits i.e. 00, 01, 10, and 11



The FA starts in state 1 (since carry is 0) and inputs a pair of bits. If the pair is 11, the FA outputs a 0 and switches to state 2 (since the carry is 1), where the next pair of bits is input and is added to a carry bit of 1.

Example: Consider the addition of 52 and 21

110100 - (binary representation of 52)

010101 - (binary representation of 21)

Since adding numbers is done from right to left, the first input symbol is 01, representing a 0 in the rightmost (binary) digit of 52 and a 1 in the rightmost digit of 21. The machine enters state q_0 (since there is no carry) and outputs a 1. The next input is 00 because both numbers have zero as the second rightmost digit. The machine enters state q_0 and outputs 0. The next input is 11. The machine enters state q_1 (since the carry is 1) and outputs 0. Being in state q_1 means that there is

a carry from this position into the next. And the remaining bits can be worked out to get 1001001 (i.e. 73).

38 votes

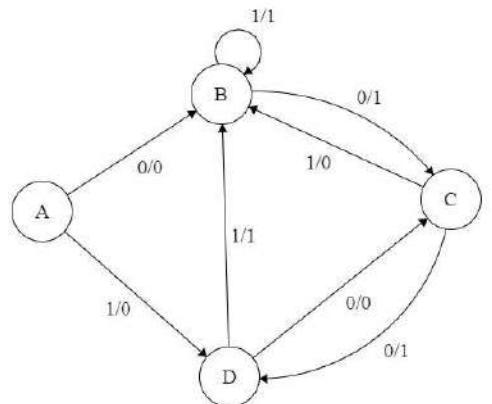
-- Saurav Kumar Gupta (2.2k points)

5.5.10 Finite Automata: GATE1995-2.23 top

<https://gateoverflow.in/2636>



Selected Answer



if A is start state , shortest sequence is 10 or 00 to reach C

if B is start state , shortest sequence is 0 to reach C

if C is start state , shortest sequence is 10 or 00 to reach C

if D is start state , shortest sequence is 0 to reach C

B) is correct.

30 votes

-- Praveen Saini (54.6k points)

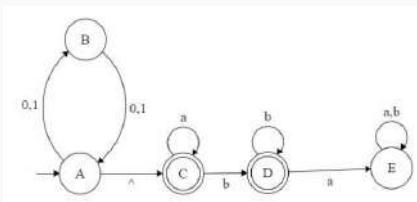
5.5.11 Finite Automata: GATE1996-12 top

<https://gateoverflow.in/2764>



Selected Answer

We can combine the final state of M_1 with the start state of M_2 as follows recognizing $L_1 L_2$. But before we combine M_1 and M_2 remove the final state of M_1 as the new machine can accept L_1 also while it should accept only $L_1 \cdot L_2$.



~Pic by Praveen Saini

17 votes

-- Arjun Suresh (350k points)

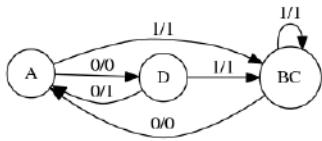
5.5.12 Finite Automata: GATE1996-2.23 top

<https://gateoverflow.in/2752>



Selected Answer

3 states are required in the minimized machine. States B and C can be combined as follows:



👍 22 votes

-- Arjun Suresh (350k points)

5.5.13 Finite Automata: GATE1997-20 top

<https://gateoverflow.in/2280>



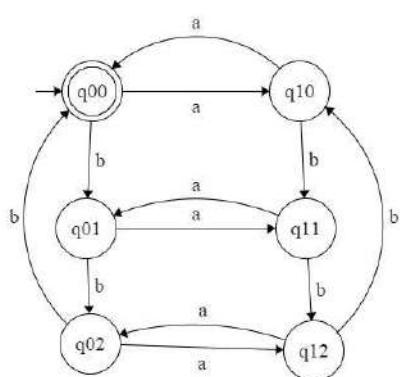
Selected Answer

A state q_{xy} means $n_a \bmod 2 = x$, $n_b \bmod 3 = y$

q_{00} means $n_a \bmod 2 = 0$, $n_b \bmod 3 = 0$ [no of a 's is divisible of 2 and no of b 's are divisible of 3]

$q_{00} \ x \ a \rightarrow q_{10}$

$q_{00} \ x \ b \rightarrow q_{01}$ and so on



👍 18 votes

-- Praveen Saini (54.6k points)

5.5.14 Finite Automata: GATE1997-21 top

<https://gateoverflow.in/2281>



Selected Answer

Suppose we have a finite automation for L , then we can build a finite automation for L^P by marking all the states from which final state is reachable as the final states for new automaton, the reasoning is that suppose we can reach final state f from some state q , then that means there exists some string s' that takes automation from q to f , so if there is some string s that takes automation to state q from start state this string should belong to the new language L^P . (L^P is the set of all prefix strings for the string in L)

Also, we can obtain an automation for L^R by swapping the start and final states of original automation L and by reversing all the edges in the DFA.

28 votes

-- Omesh Pandita (2.5k points)

5.5.15 Finite Automata: GATE1998-1.10 top

<https://gateoverflow.in/1647>



Selected Answer

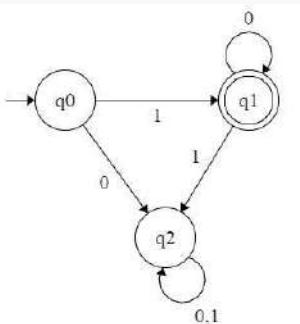
Option B is correct .

- A. A. is regular

$$L = \{1, 10, 100, 1000, 10000, \dots\}$$

regular expression 10^*

DFA :



- B. $L = \{1, 11, 1111, 11111111, \dots\} = \{1^i \mid i = 2^n, n \geq 0\}$ is non regular language

- C. Equal- Equal, is CFL, and non regular

- D. $L = \{1^i 0 1^i \mid i > 0\} \cup \{1\}$ is also CFL, and non regular

38 votes

-- Praveen Saini (54.6k points)

5.5.16 Finite Automata: GATE1998-2.5 top

<https://gateoverflow.in/1677>

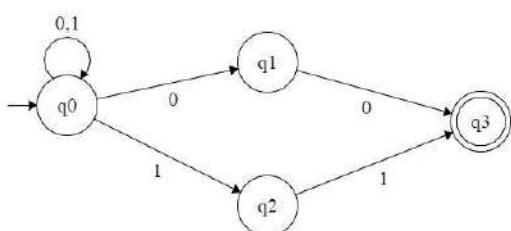


Selected Answer

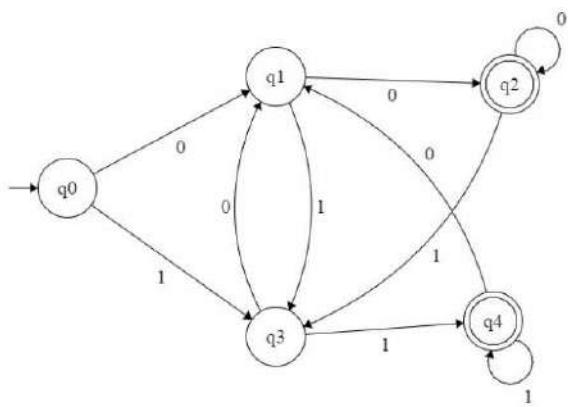
Binary strings whose last two symbols are same .

Regular expression = $(0 + 1)^* (00 + 11)$

Having NFA



Equivalent DFA



Total no of states = 5

29 votes

-- Praveen Saini (54.6k points)

5.5.17 Finite Automata: GATE1998-4 top

<https://gateoverflow.in/1695>



Selected Answer

Suppose we have decimal no 3 after that we get a symbol 2 . it becomes 32 as $3 \times 10 + 2 = 32$

In binary if we have 10 (i.e 2 in decimal say old no) and after that we get symbol 1 it become 101(i.e 5 in decimal say new no)

$$2(\text{old no.}) \times 2(\text{base}) + 1(\text{input symbol}) = 5(\text{new no.})$$

Now in the given problem, binary no is divisible by 5 , i.e 0,101,1010,1111.....

We need 5 states in DFA , 0,1,2,3 and 4 .Each state represent remainder that comes when we divide no by 5.

Input symbol = {0,1}

We get the transition:

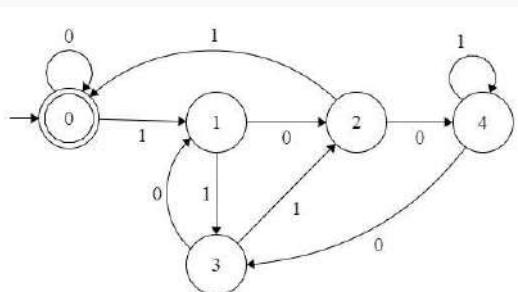
$$[\text{Old state} \times \text{base} + \text{input symbol}] \bmod 5 = \text{New state}, \text{ where base is 2}$$

$$[0 \times 2 + 0] \bmod 5 = 0 \quad [1 \times 2 + 0] \bmod 5 = 2 \quad [2 \times 2 + 0] \bmod 5 = 4$$

$$[0 \times 2 + 1] \bmod 5 = 1 \quad [1 \times 2 + 1] \bmod 5 = 3 \quad [2 \times 2 + 1] \bmod 5 = 0$$

$$[3 \times 2 + 0] \bmod 5 = 1 \quad [4 \times 2 + 0] \bmod 5 = 3$$

$$[3 \times 2 + 1] \bmod 5 = 2 \quad [4 \times 2 + 1] \bmod 5 = 4$$



30 votes

-- Praveen Saini (54.6k points)

5.5.18 Finite Automata: GATE1999-1.4 top

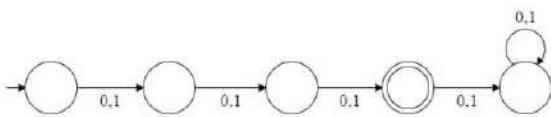
<https://gateoverflow.in/1458>



As far as for above problem say regular expression for $(0 + 1)(0 + 1)\dots 3$ times

$$= (0 + 1)(0 + 1)(0 + 1)$$

Having **DFA**:



So, for regular expression $(0 + 1)(0 + 1)\dots N$ times have **N + 2** states in DFA

$N + 1$ states in NFA (we can remove dead state)

When question is about minimum state finite automata (and nothing is mentioned NFA/DFA) then which ever having minimum no.

$N + 1$ states.

Upvote 35 votes

-- Praveen Saini (54.6k points)

5.5.19 Finite Automata: GATE2001-1.6 top

<https://gateoverflow.in/699>



Answer is (B) 2^N .

In **DFA** any subset of the N states (for N element set 2^N subsets possible) can become a new state and they can remain even when the DFA is minimized. So, maximum we can get 2^N states for the minimized **DFA**. (at least in question must be a typo for at most).

Upvote 23 votes

-- jayendra (8.2k points)

5.5.20 Finite Automata: GATE2001-2.5 top

<https://gateoverflow.in/723>



Answer is (D). It can be proved using Myhill Nerode theorem. We need a separate state for $\#a \text{ mod } 6 = 0..5$ and $\#b \text{ mod } 8 = 0..7$. Each combination of them must also be a new state giving $6 * 8 = 48$ minimum states required in the DFA.

Reading Myhill-Nerode theorem might be confusing though it is actually very simple.

<http://courses.cs.washington.edu/courses/cse322/05wi/handouts/MyhillNerode.pdf>

Upvote 27 votes

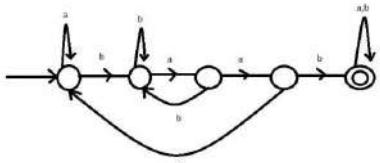
-- Arjun Suresh (350k points)

5.5.21 Finite Automata: GATE2001-5 top

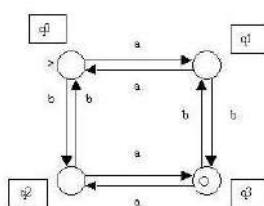
<https://gateoverflow.in/746>



DFA for A:



Part (B):



19 votes

-- jayendra (8.2k points)

5.5.22 Finite Automata: GATE2002-2.13 top

<https://gateoverflow.in/843>



It is 3 states as we need a state each for length mod 3 = 0, 1 and 2.

20 votes

-- priya023 (191 points)

5.5.23 Finite Automata: GATE2002-2.5 top

<https://gateoverflow.in/835>



Answer should be option (A).

Option (B) and (C) are clearly wrong . it says input 11 then o/p 01 and i/p 10 then o/p 00 but here at single bit o/p is 2 bit sequence

Now, for option (A) we can trace out. Suppose string is 0111.

at A---0---> A---1---> B--1-->C---1-->C

O/P 00 01 10 10

We can see here at (A,0)---> (A,00) which sum of $0 + 0 = 00$, (previous i/p bit + present i/p bit)

(A,1)--->(B,01) which is sum of $0 + 1 = 1 = 01$,

(B,1)--->(C,10) which is sum of $1 + 1 = ($ previous i/p bit + present i/p bit)= 10,

$(C, 1) \rightarrow (C, 10)$ which is sum of $1 + 1 = 10$

So, answer should be (A).

28 votes

-- sonam vyas (14.8k points)

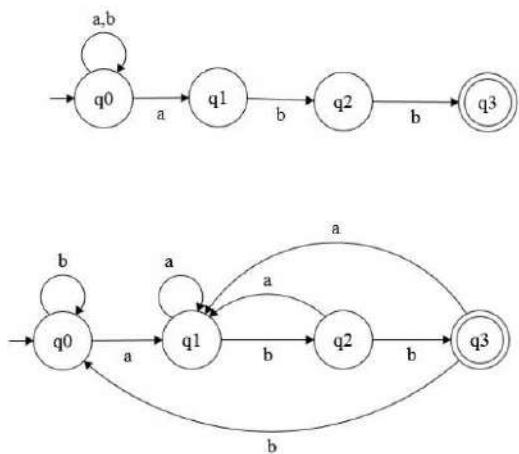
5.5.24 Finite Automata: GATE2002-21 top

<https://gateoverflow.in/874>



Selected Answer

NFA for regular expression $(a + b)^* abb$ and its equivalent DFA will be as follow:



21 votes

-- Praveen Saini (54.6k points)

5.5.25 Finite Automata: GATE2003-50 top

<https://gateoverflow.in/939>



Selected Answer

Language of above **DFA** is all strings over $\{0, 1\}$ that contain substring 001.

Regular expression of above DFA is $(0 + 1)^* 001(0 + 1)^*$

1 that is underlined can not be first bit of 7-bit binary no, but can be fourth bit or last bit.

Case 1: if it is 4th bit ,then possible set of strings can be

First 001 **two bits** Last = $1001(00 + 01 + 10 + 11)1 = 4$ strings

Case 2 : if it is last bit, then possible set of strings can be

First **two bits** fourth 001 = $1(00 + 01 + 10 + 11)1001 = 4$ strings

String common in both cases 1001001

Total strings = $4 + 4 - 1 = 7$ strings

40 votes

-- Praveen Saini (54.6k points)

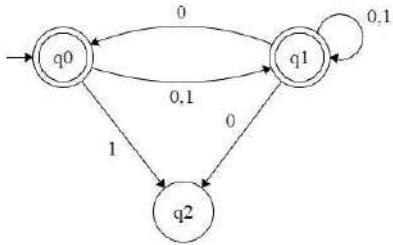
5.5.26 Finite Automata: GATE2003-55 top

<https://gateoverflow.in/943>



Answer is **B**.

As the problem said:



As in above **NFA** language L_1 is $\{0,1\}^*$. [we don't know L , we need not to find out]

Option A is wrong as L is accepting 1 and L_1 is also accepting 1

Option C is wrong as L_1 accepting $^\wedge$,null, but L is not.

Option D is wrong for same reason as option C is wrong.

14 votes

-- Praveen Saini (54.6k points)

5.5.27 Finite Automata: GATE2004-86 top

<https://gateoverflow.in/1080>



Option **(C)** and option **(D)** are cancelled out clearly because with 3 1s we can reach the final state. There is an string where we can reach the final state by 6 1's now 6 is nt odd but it is divisible by 3. Hence, option **(A)** is correct.

17 votes

-- Bhagirathi Nayak (14.1k points)

5.5.28 Finite Automata: GATE2004-IT-41 top

<https://gateoverflow.in/3684>



$$\sigma(A, a) = A, \quad A \rightarrow aA$$

$$\sigma(A, b) = B, \quad A \rightarrow bB$$

$$\sigma(B, a) = B, \quad B \rightarrow aB$$

$$\sigma(B, b) = A, \quad B \rightarrow bA$$

B is final state so $B \rightarrow \epsilon$

31 votes

-- Praveen Saini (54.6k points)

5.5.29 Finite Automata: GATE2005-53 top

<https://gateoverflow.in/1376>



A is Wrong, since $abbb$ is accepted. (1 a is followed by more than 2 b's)

C is Wrong, since $abba$ contains abb as substring, but is still not accepted.

D is Wrong, since ab does not contain aa as substring, but is still not accepted.

Hence, correct answer is **B**.

1 31 votes

-- saurabhrk (1.3k points)

5.5.30 Finite Automata: GATE2005-63 top

<https://gateoverflow.in/1386>



Answer is **option B**.

For any binary no, FSM read input from LSB and remain unchanged till first 1, and it complement after that

$100 \rightarrow 100$ [1's complement of $100 + 1 = 011 + 1 = 100 = 2$'s complement of 100]

$010 \rightarrow 110$ [1's complement of $010 + 1 = 101 + 1 = 110 = 2$'s complement of 010]

$1010\underline{1}00 \rightarrow 0101\underline{1}00$ [1's complement of $1010100 + 1 = 0101011 + 1 = 0101100$]

Note : Underline part is unchanged (till first 1 from lsb) then 0's changed to 1's and 1's changed to 0's

1 18 votes

-- Praveen Saini (54.6k points)

5.5.31 Finite Automata: GATE2005-IT-37 top

<https://gateoverflow.in/3784>



Misprints : Edge $Y \rightarrow Z$ (0 edge)
Edge $Z \rightarrow Y$ (1 edge)

Answer: **A**.

Explanation:

Writing Y and Z in terms of incoming arrows (Arden's method) :

$$Y = X0 + Y0 + Z1$$

$$Z = X0 + Z1 + Y0$$

Hence $Y = Z$. So, option (**A**).

1 29 votes

-- Dipak Majhi (979 points)

5.5.32 Finite Automata: GATE2005-IT-39 top

<https://gateoverflow.in/3786>



Selected Answer

- $S \rightarrow Xa \mid Ya$
- $X \rightarrow Za$
- $Z \rightarrow Sa \mid \epsilon$
- $Y \rightarrow Wa$
- $W \rightarrow Sa$

This is left linear grammar having language L . Convert it into right linear using following rule:

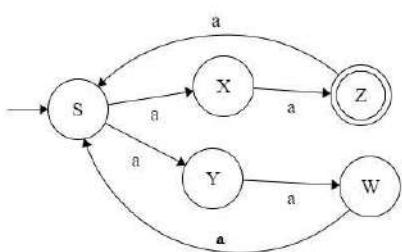
- $V_i \rightarrow V_j w$ Reverses to $V_i \rightarrow w^R V_j$
- $V_i \rightarrow w$ Reverses to $V_i \rightarrow w^R$

If the left linear grammar produced language L then the resulting right linear grammar produces L^R .

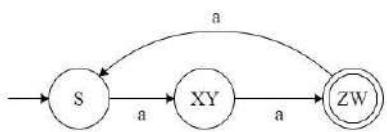
- $S \rightarrow aX \mid aY$
- $X \rightarrow aZ$
- $Z \rightarrow aS \mid \epsilon$
- $Y \rightarrow aW$
- $W \rightarrow aS$

is right linear grammar having language L^R .

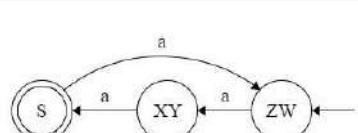
Having **NFA**



Having **DFA** for language L^R



DFA for language L (reversal)



$L = \{w : n_a(w) \bmod 3 = 2, w \text{ belong to } \{a, b\}^*\}$ same as Omesh Pandita answered.

Having

3 states.

36 votes

-- Praveen Saini (54.6k points)

5.5.33 Finite Automata: GATE2006-34 top<https://gateoverflow.in/1291>

Selected Answer

Given language $L = (111 + 11111)^*$

Strings , that belongs in the language

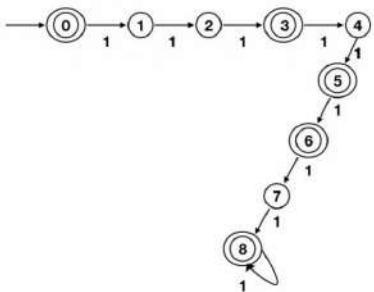
 $L = \{ \text{null}, 111, 11111, 111111, 11111111, 111111111, 1111111111, \dots \}$ form string length 8 , (number of 1's) , you can generate any length of string from length 3 and 5 (i.e. length 8 ,length 9, length 10 , length 11 ,.....etc)

 $L = \{ \text{null}, 111, 11111, 111111, 11111111, 111111111^* \}$

Strings in length , that belongs in the language

 $L = \{0, 3, 5, 6, 8, 9, 10, 11, \dots\}$

So, required DFA will be:



So, there are 5 states are final states and 4 states are non-final states ,total number of states are 9 states .

Hence, option D is true.

37 votes

-- Mithlesh Upadhyay (5.8k points)

5.5.34 Finite Automata: GATE2006-IT-3 top<https://gateoverflow.in/3542>

Selected Answer

for u for v for w
 $\delta(s, \text{abbaba}) \quad \delta(s, \text{bab}) \quad \delta(s, \text{aabb})$
 $\vdash \delta(x, \text{bbaba}) \quad \vdash \delta(t, \text{ab}) \quad \vdash \delta(x, \text{abb})$
 $\vdash \delta(x, \text{baba}) \quad \vdash \delta(t, \text{b}) \quad \vdash \delta(s, \text{bb})$
 $\vdash s -$
 $\vdash \delta(x, \text{aba}) \quad \text{rejected} \quad \vdash \delta(t, \text{b})$
 $\vdash \delta(s, \text{ba})$
 $\vdash s -$
rejected
 $\vdash \delta(t, \text{a})$
 $\vdash t -$
accepted

20 votes

-- Praveen Saini (54.6k points)

5.5.35 Finite Automata: GATE2006-IT-37 top

<https://gateoverflow.in/3576>

Selected Answer

Present State	Inputs		Next state
S	x	y	S ⁺
0	X	0	1
0	X	1	0
1	0	X	0
1	1	X	1

From above table:

$$S^+ = S'y' + Sx$$

39 votes

-- Praveen Saini (54.6k points)

5.5.36 Finite Automata: GATE2007-29 top

<https://gateoverflow.in/1227>

Selected Answer

Answer will be (A) 15 states.

We need a separate state for #0 = 0,1,2 and #1 = 0,1,2,3,4 giving total minimum number of states = $3 * 5 = 15$.

This is a direct consequence of Myhill-Nerode theorem.

<http://courses.cs.washington.edu/courses/cse322/05wi/handouts/MyhillNerode.pdf>

18 votes

-- Arjun Suresh (350k points)

5.5.37 Finite Automata: GATE2007-74 top

<https://gateoverflow.in/1270>

Selected Answer

The answer is C.

You can see that both the states, q_1 and q_2 are final and are accepting $(a+b)^*$.

[Edit]

1. q_3 is unreachable state, hence it can be removed.
2. States q_1 and q_2 are indistinguishable, so, they can be merged.

24 votes

-- Gate Keeda (19.6k points)

5.5.38 Finite Automata: GATE2007-75 top

<https://gateoverflow.in/43514>

Selected Answer

Answer will be **2** state.



19 votes

-- Manoj Kumar (38.6k points)

5.5.39 Finite Automata: GATE2007-IT-47 top

<https://gateoverflow.in/3489>



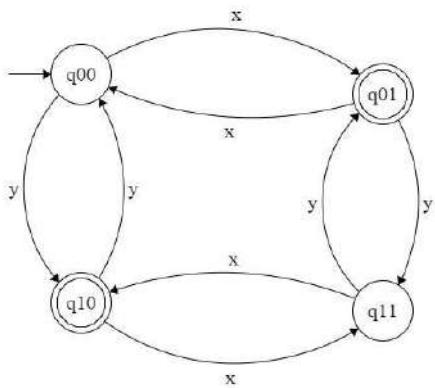
Above **DFA** can be redesigned as: [S_0 as q_{00} , S_1 as q_{10} , S_2 as q_{11} , S_3 as q_{01}]

Where, each state as $q_{ab} [a = n_a \bmod 2, b = n_b \bmod 2]$

q_{00} as $n_a \bmod 2 = 0, n_b \bmod 2 = 0$ [no of x is even no of y is even]

and $\delta(q_{00}, x) \rightarrow q_{10}$ [$(0+1) \bmod 2 = 1$ as x increase from 0 to 1] $\delta(q_{00}, y) \rightarrow q_{01}$

and $\delta(q_{10}, x) \rightarrow q_{00}$ [$(1+1) \bmod 2 = 0$] $\delta(q_{00}, y) \rightarrow q_{01}$ and soon



q_{01} is final state mean where no of x is even and no of y is odd

q_{10} is final state mean where no of x is odd and no of y is even.

So, **D** is correct answer.

19 votes

-- Praveen Saini (54.6k points)

5.5.40 Finite Automata: GATE2007-IT-50 top

<https://gateoverflow.in/3492>



Design a **DFA** using P and Q having p_0q_0 as start state

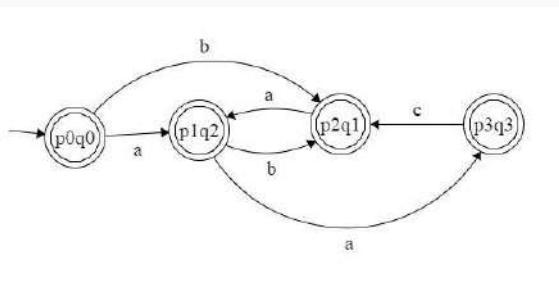
$\delta(p_0q_0, a) \rightarrow \delta(p_0, a) \cup \delta(q_0, a)$

$Q \setminus \Sigma$	a	b	c
$\rightarrow p0q0^*$	$p1q2$	$p2q1$	
$p1q2^*$	$p3q3$	$p2q1$	
$p2q1^*$	$p1q2$	$p3$ (No Need)	
$p3q3^*$		$q2$ (No Need)	$p2q1$

In case of intersection final state are those where final state of P and final state of Q comes together.

No need mean when we reach to $p3$ (or $q2$) then we cannot reach to any final state because we cannot have state of P and Q together (intersection) so no need to show it in diagram [may draw a dead state for it if required]

DFA results in:



That is option **A**.

Note : i) DFA must have transition for each symbol $Q \times \Sigma \rightarrow Q$

34 votes

-- Praveen Saini (54.6k points)

See the languages being accepted by P and Q . In P before a 'c' there must be either 'b' or 'aa'. In Q , before 'c' there must be 'aa'. So, obviously, in their intersection before 'c' there must be 'aa' which is satisfied only by option A.

20 votes

-- Arjun Suresh (350k points)

5.5.41 Finite Automata: GATE2007-IT-71 top

<https://gateoverflow.in/3523>



Selected Answer

When we say non-deterministic finite automata recognizes the language defined by the regular expression R then it means that it won't accept any other string that does not fall under the R and it will only accept all strings that fall under R .

So, we see **B**. accepts **aba**, **C**. accepts **aba**, **D** accepts **a** just find some examples that do not follow the rule. So, option **A** is correct..

17 votes

-- Shreya Roy (4.3k points)

5.5.42 Finite Automata: GATE2007-IT-72 top

<https://gateoverflow.in/3524>**DFA** given in option **A**Having S_3 and S_4 are equivalent state. That can minimized.and result in **DFA** given in:https://gateoverflow.in/3523/gate2007-it_71

20 votes

-- Praveen Saini (54.6k points)

5.5.43 Finite Automata: GATE2008-49 top

<https://gateoverflow.in/462>

Selected Answer

	States	a	b
→	11 (P)	12 22	
	12 (S)	11 21	
	21 (Q)	22 12	
(F)	22 (R)	21 11	

So, 11 is P and 22 is R in choice. So, the answer should be (A) but in the row for S, it should be P and Q and not Q and P.

28 votes

-- Arjun Suresh (350k points)

5.5.44 Finite Automata: GATE2008-52 top

<https://gateoverflow.in/464>

Selected Answer

 $S - 4$ is confirmed $R - 3$ is true coz everything it accepts ends with 1; this is made mandatory only by 3
this rules out option B and option Duse string 01010 and compare P vs Q; this makes $Q - 2$ as confirmed.Hence, **option C** is correct.

21 votes

-- Amar Vashishth (30.5k points)

5.5.45 Finite Automata: GATE2008-IT-32 top

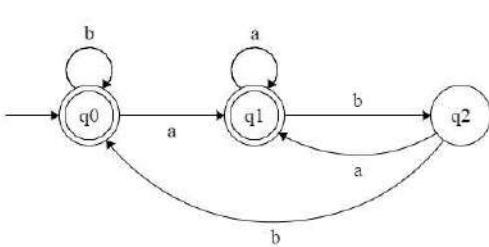
<https://gateoverflow.in/3342>

Selected Answer

Above **DFA** is for regular expression $(a + b)^*ab$. All strings end with ab .

Complement of **DFA** accepts all strings does not end with ab .

DFA(L') is:



B. String begin with either a or b .

ab (string start with a) doesn't accept in it reach to nonfinal state $q2$.

bab (string start with b) doesn't accept in it reach to nonfinal state $q2$.

C. Set of strings that do not contain the substring ab

aba (have substring ab) does accept in it reach to final state $q1$.

D. The set described by the regular expression $b^*aa^*(ba)^*b^*$

b is string accepted by $\text{DFA}(L')$ but above regular expression cannot derive it.

Option **A** is correct.

DFA (L') accepts all strings that doesn't end with ab .

拇指图标 28 votes

-- Praveen Saini (54.6k points)

5.5.46 Finite Automata: GATE2008-IT-36 top

<https://gateoverflow.in/3346>



Selected Answer

$L_1 : (0 + 10)^*11(0 + 1)^*$ $L_2 : (0 + 1)^*11(0 + 1)^*$ it is quite clear that $L_1 = L_2$..

As both languages L_1 and L_2 are equal So Complement of Language L_2 will be the complement of Language L_1 also. For a given language L , $L \cap L^C = \Phi$

Hence, both options **(A)** and **(C)** are correct.

拇指图标 32 votes

-- Vicky Bajoria (5k points)

5.5.47 Finite Automata: GATE2008-IT-6 top

<https://gateoverflow.in/3266>



Selected Answer

A state in a DFA will be a subset of the set of states of the equivalent NFA. So, the maximum number of states in the equivalent DFA of an NFA, will be 2^n , where n is the number of states in NFA, as a set with n items has maximum 2^n subsets.

So, answer here is (A).

24 votes

-- Arjun Suresh (350k points)

5.5.48 Finite Automata: GATE2009-27 [top](#)

<https://gateoverflow.in/1313>



Selected Answer

From above table, we look at next state part

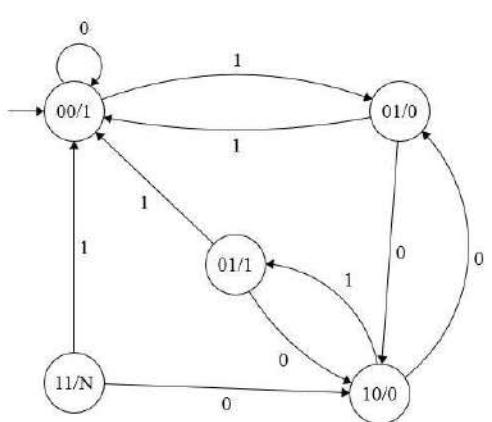
Whenever we reach state 00 we get output 1 [at row 1, row 6, row 8], **so we have state 00 with output 1**

When we reach at state 01, we get output 0 [at row 3, row 5] and output 1 [row 7], **so we have two state 01 with output 0, 01 with output 1.**

When we reach at state 10, we get output we get output 0 [at row 2, row 4], **so we have state 10 with output 0.**

We don't reach at state 11 [11 is not there in next state part], but **we have state 11 with don't know (N) output.**

If we draw the Moore Machine for above **FSM** [from the table: present state x input symbol -> next state]



It is clear from **FSM** from state 00 to reach state 01 with output 1 i.e, 01/1 with need **minimum length input 101**

Minimum length of input = length of 101. **That is 3.**

26 votes

-- Praveen Saini (54.6k points)

5.5.49 Finite Automata: GATE2009-41 [top](#)

<https://gateoverflow.in/1327>

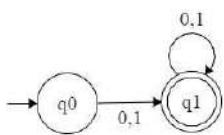


Selected Answer

- A. Begin either with 0 or 1

Regular expression $(0+1)(0+1)^*$

[begin with 0 or 1 Anything]



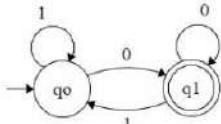
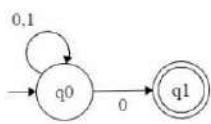
B. End with 0

regular expression = $(0 + 1)^*0$

NFA

[Anything end with 0]

Equivalent DFA



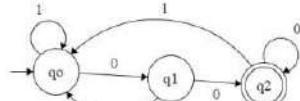
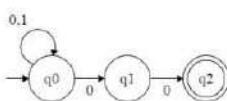
C. End with 00

Regular expression $(0 + 1)^*00$

NFA

[Anything end with 00]

Equivalent DFA



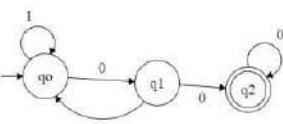
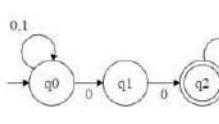
D. Containing the substring

Regular expression = $(0 + 1)^*00(0 + 1)^*$

NFA

[Anything substring 00 Anything]

Equivalent DFA



So, C is the correct answer.

31 votes

-- Praveen Saini (54.6k points)

5.5.50 Finite Automata: GATE2010-41 top

<https://gateoverflow.in/2342>



Selected Answer

We need a state for counting the length. So, for length n we need $n + 1$ states (one for length zero). We don't need a reject state for larger strings as we have NFA and not DFA. So, totally $n + 1$ states are required. (For DFA it would be $n + 2$).

42 votes

-- Arjun Suresh (350k points)

5.5.51 Finite Automata: GATE2011-42 top

<https://gateoverflow.in/2144>



Selected Answer

(B) $n + 1$

We need a state for strings of length $0, 1, 2, \dots, n$ (and their respective multiples with k). Each of these set of strings form an equivalence class as per Myhill-Nerode relation and hence needs a separate state in min-DFA.

Myhill-Nerode Class 1	Myhill-Nerode Class 2	Myhill-Nerode Class n	Myhill-Nerode Class n+1
-----------------------	-----------------------	-----------------------	-------------------------

Myhill-Nerode Class 1	Myhill-Nerode Class 2	Myhill-Nerode Class n	Myhill-Nerode Class n+1
ϵ	$a, \#a$ $= n$ $+ 1,$ $\#a$ $= 2n$ $+ 1, .$ \dots	$\#a$ $= n$ $- 1,$ $\#a$ $= 2n$ $- 1,$ $\#a$ $= 3n$ \dots	$\#a$ $= n,$ $\#a$ $= 2n,$ $\#a$ $= 3n,$ \dots

One thing to notice here is $k > 0$. Because of this we are not able to combine Class 1 and Class $n+1$. Had it been $k \geq 0$, we would have had only n equivalent classes and equivalently n states in the minimal DFA.

Upvotes: 23 votes

-- Arjun Suresh (350k points)

5.5.52 Finite Automata: GATE2011-45 top

<https://gateoverflow.in/2147>



Selected Answer

(A) is the answer.

In (B) and (C) when the first letter of input is ' b ' we reach final state, while in the given DFA first letter ' b ' is not a final state. So, (B) and (C) are not accepting same language as the given DFA.

In (D) we can reach final state when the last letter is ' a ', whatever be the previous transitions. But in the given DFA, when the first 2 letters are ' b ' we can never reach the final state. So, (D) is also accepting a different language than the given DFA.

Upvotes: 27 votes

-- Arjun Suresh (350k points)

5.5.53 Finite Automata: GATE2012-12 top

<https://gateoverflow.in/44>



Selected Answer

The language being accepted is a^+ . So, complement of the language is $\{\epsilon\}$.

Upvotes: 44 votes

-- Arjun Suresh (350k points)

5.5.54 Finite Automata: GATE2012-46 top

<https://gateoverflow.in/2159>



Selected Answer

(D) is the answer. From 00 state, a '0' should take the DFA to the dead state- q . From 11, a '0' should go to 10 representing the 10 at the end of string so far. Similarly, from 00 a 1 should go to 01, from 01 a '1' should go to 11 and from 10 a '0' should go to '00'.

Upvotes: 21 votes

-- Arjun Suresh (350k points)

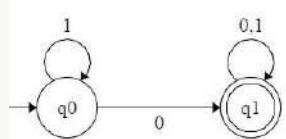
5.5.55 Finite Automata: GATE2013-33 top

<https://gateoverflow.in/1544>



Selected Answer

1. Complement of $L(A)$ (regular language) is regular , i.e also Context Free. **True**
2. Regular expression is $(11^*0 + 0)(0 + 1)^*$ **True**
3. Minimized DFA is:



[both non-final states are equivalents can be minimized]

So, 3 is **False**

4. From 3, shortest length string reached from q_0 to q_1 (final) is 0, so 4 is **False**

Note :

- a. $(0 + 1)^*0^*1^* = (0 + 1)^* + \text{something} = (0 + 1)^*$
- b. $(11^*0 + 0)(0 + 1)^* = (11^* +)0(0 + 1)^* = 1^*0(0 + 1)^*$ look at Minimized DFA at 3.

44 votes

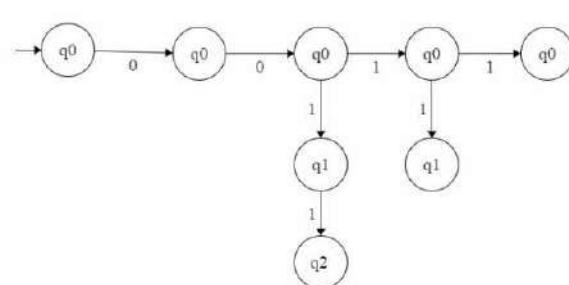
-- Praveen Saini (54.6k points)

5.5.56 Finite Automata: GATE2014-1-16 top

<https://gateoverflow.in/1782>



Selected Answer



q_0, q_1 and q_2 are reachable from q_0 on input 0011

24 votes

-- Praveen Saini (54.6k points)

5.5.57 Finite Automata: GATE2015-1-52 top

<https://gateoverflow.in/8362>

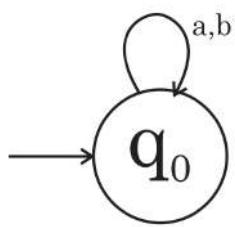


Selected Answer

$$L(M) = (a + b)^* \quad a = \{a, aa, ba, aaa, aba, bba, \dots\}$$

$$L(N) = (a + b)^* \quad b = \{b, ab, bb, aab, abb, bbb, \dots\}$$

So, $L(M) \cap L(N) = \{\}$. So, in the minimal DFA, we just have 1 start state with all transitions going to it self and no final state.



69 votes

-- Arjun Suresh (350k points)

5.5.58 Finite Automata: GATE2015-2-53 top

<https://gateoverflow.in/8256>

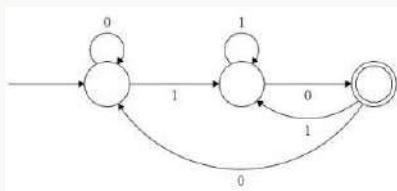


All strings ending with 10. So, we need
3 states.

1. From first state on 1, we go to second state.
2. From second state on 0 we go to third state.
3. From third state on 0 we go to first state and on 1 we go to second state.

Only third state is final.

$L = (0+1)^*10$ And following minimal DFA will be



30 votes

-- Arjun Suresh (350k points)

5.5.59 Finite Automata: GATE2015-3-18 top

<https://gateoverflow.in/8415>



First we can draw dfa for L which has 5 states after that for L compliment we will convert all final to non final and all non final to final so, total states is 5.

Answer is option B.

36 votes

-- Anoop Sonkar (5k points)

5.5.60 Finite Automata: GATE2016-2-16 top

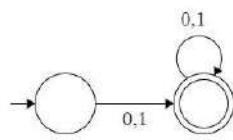
<https://gateoverflow.in/39562>



All strings over $\{0,1\}$ having length ≥ 1

$$(0+1)^*(0+1)(0+1)^* = (0+1)(0+1)^* = (0+1)^*(0+1) = (0+1)^+$$

Having DFA:



No of states in minimal DFA = 2.

Upvote 56 votes

-- Praveen Saini (54.6k points)

5.5.61 Finite Automata: GATE2016-2-42 top

<https://gateoverflow.in/39591>



Selected Answer

- False, as in NFA, it is not necessary that all states have transitions for all symbols.
- True, there exists a regular language $A = \{\}$, such that for all languages B , $A \cap B = \{\}$ is regular

So, answer is option B.

Upvote 79 votes

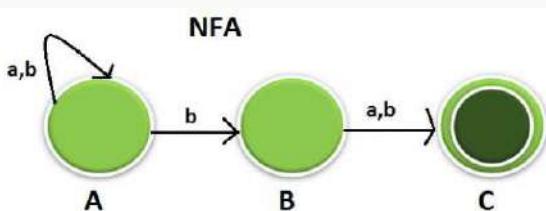
-- Praveen Saini (54.6k points)

5.5.62 Finite Automata: GATE2017-1-22 top

<https://gateoverflow.in/118302>



Selected Answer



NFA to DFA Conversion:

In the STT below you can see **4 states** are there but before confirming the ans **lets try to minimize**, It cant be minimized further because A goes to Nonfinal Nonfinal & AB goes to Final Final hence can not be merged. Similarly $*AC$ goes NF NF & $*ABC$ goes to F F hence can not be merged. **Hence, 4 is the final answer.**

Delta	a	b
A	A	AB
AB	*AC	*ABC
*AC	A	AB
*ABC	*AC	*ABC

Upvote 29 votes

-- Ahwan Mishra (10.5k points)

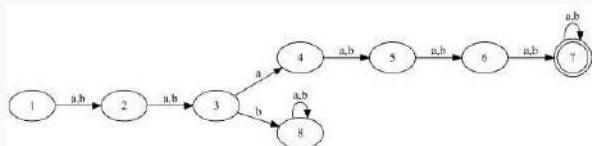
5.5.63 Finite Automata: GATE2017-2-25 top

<https://gateoverflow.in/118160>



Selected Answer

Answer is **8** states included one trap state (**8**) and final state (**7**).



3rd symbol from start is an 'a'

33 votes

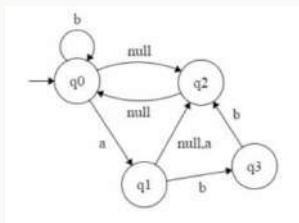
-- Prashant Singh (59.8k points)

5.5.64 Finite Automata: GATE2017-2-39 top

<https://gateoverflow.in/118384>



Selected Answer



C is the answer.

20 votes

-- Prashant Singh (59.8k points)

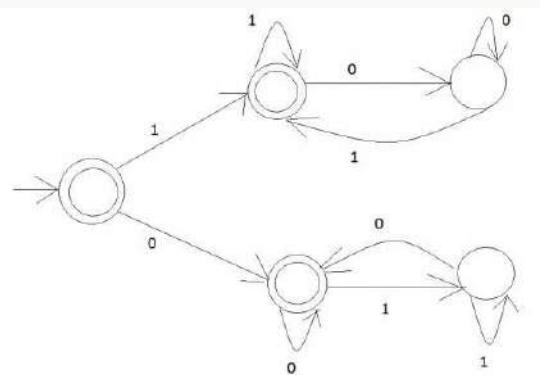
5.5.65 Finite Automata: ISI2014-CS-4a top

<https://gateoverflow.in/47443>



Selected Answer

The **DFA** for that language $w\$$ is:



13 votes

-- srestha (87k points)

5.6

Grammar(2) top

5.6.1 Grammar: GATE2008-51 top

<https://gateoverflow.in/463>

Match the following:

E.	Checking that identifiers are declared before their use	P.	$L = \{a^n b^m c^n d^m \mid n \geq 1, m \geq 1\}$
F.	Number of formal parameters in the declaration of a function agrees with the number of actual parameters in a use of that function	Q.	$X \rightarrow XbX \mid XcX \mid dXf \mid g$
G.	Arithmetic expressions with matched pairs of parentheses	R.	$L = \{wcw \mid w \in (a \mid b)^*\}$
H.	Palindromes	S.	$X \rightarrow bXb \mid cXc \mid \epsilon$

- A. E-P, F-R, G-Q, H-S
 B. E-R, F-P, G-S, H-Q
 C. E-R, F-P, G-Q, H-S
 D. E-P, F-R, G-S, H-Q

gate2008 normal theory-of-computation grammar

Answer

5.6.2 Grammar: ISI2013-CS-4a top

<https://gateoverflow.in/47637>

Give a context-free grammar G that generates $L = \{0^i 1^j 0^k \mid i + k = j\}$.
 Prove that $L = L(G)$.

descriptive isi2013 context-free-language grammar theory-of-computation

Answer

Answers: Grammar

5.6.1 Grammar: GATE2008-51 top

<https://gateoverflow.in/463>



H-S is true coz strings generated by this grammar satisfies the definition of an even length palindrome string. this rules out B and D options.

G-Q is confirmed as both options A and C has it as true.

E-R is true coz R is the only grammar that checks: what(w) has occurred earlier is present afterwards This equals the definition of E

Hence, **option C** is true.

30 votes

-- Amar Vashishth (30.5k points)

5.6.2 Grammar: ISI2013-CS-4a top

<https://gateoverflow.in/47637>



$$L = \{0^i 1^j 0^k \mid j = i + k, i, j, k \geq 0\} = \{\epsilon, 0110, 011100, 001110, \dots\}$$

$$\text{Now, } L = \{0^i 1^{i+k} 0^k \mid i, k \geq 0\}$$

Here, **CFG** will be:

- $S \rightarrow S_1 S_2$
- $S_1 \rightarrow 0S_11 \mid \epsilon$
- $S_2 \rightarrow 1S_20 \mid \epsilon$

12 votes

-- srestha (87k points)

5.7

Identify Class Language(33) top

5.7.1 Identify Class Language: CMI2010-A-01 top

<https://gateoverflow.in/46131>

Over the alphabet $\{0, 1\}$, consider the language

$$L = \{w \mid w \text{ does not contain the substring } 0011\}$$

Which of the following is true about L .

- L is not context free
- L is regular
- L is not regular but it is context free
- L is context free but not recursively enumerable

[cmi2010](#) [theory-of-computation](#) [identify-class-language](#)

[Answer](#)

5.7.2 Identify Class Language: GATE1987-1-xiii top

<https://gateoverflow.in/80293>

FORTRAN is a:

- Regular language.
- Context-free language.
- Context-sensitive language.
- None of the above.

[gate1987](#) [theory-of-computation](#) [identify-class-language](#)

[Answer](#)

5.7.3 Identify Class Language: GATE1988-2ix top

<https://gateoverflow.in/93949>

What is the type of the language L , where $L = \{a^n b^n \mid 0 < n < 327\text{-th prime number}\}$

[gate1988](#) [normal](#) [descriptive](#) [algorithms](#) [theory-of-computation](#) [identify-class-language](#)

[Answer](#)

5.7.4 Identify Class Language: GATE1991-17,a top

<https://gateoverflow.in/26653>

Show that the Turing machines, which have a read only input tape and constant size work tape, recognize precisely the class of regular languages.

[gate1991](#) [theory-of-computation](#) [descriptive](#) [identify-class-language](#)

[Answer](#)

5.7.5 Identify Class Language: GATE1994-19 top

<https://gateoverflow.in/2515>

(a) Given a set:

$$S = \{x \mid \text{there is an } x\text{-block of 5's in the decimal expansion of } \pi\}$$

(Note: *x-block* is a maximal block of *x* successive 5's)

Which of the following statements is true with respect to S ? No reason to be given for the answer.

- i. S is regular
- ii. S is recursively enumerable
- iii. S is not recursively enumerable
- iv. S is recursive

(b) Given that a language L_1 is regular and that the language $L_1 \cup L_2$ is regular, is the language L_2 always regular? Prove your answer.

gate1994 theory-of-computation identify-class-language normal

[Answer](#)

5.7.6 Identify Class Language: GATE1999-2.4 [top](#)

<https://gateoverflow.in/1482>

Multiple choices may be correct:

If L_1 is context free language and L_2 is a regular language which of the following is/are false?

- A. $L_1 - L_2$ is not context free
- B. $L_1 \cap L_2$ is context free
- C. $\sim L_1$ is context free
- D. $\sim L_2$ is regular

gate1999 theory-of-computation identify-class-language normal

[Answer](#)

5.7.7 Identify Class Language: GATE2000-1.5 [top](#)

<https://gateoverflow.in/628>

Let L denote the languages generated by the grammar $S \rightarrow 0S0 \mid 00$.

Which of the following is TRUE?

- A. $L = 0^+$
- B. L is regular but not 0^+
- C. L is context free but not regular
- D. L is not context free

gate2000 theory-of-computation easy identify-class-language

[Answer](#)

5.7.8 Identify Class Language: GATE2002-1.7 [top](#)

<https://gateoverflow.in/811>

The language accepted by a Pushdown Automaton in which the stack is limited to 10 items is best described as

- A. Context free
- B. Regular

- C. Deterministic Context free
D. Recursive

gate2002 theory-of-computation easy identify-class-language

[Answer](#)

5.7.9 Identify Class Language: GATE2004-87 [top](#)

<https://gateoverflow.in/1081>

The language $\{a^m b^n c^{m+n} \mid m, n \geq 1\}$ is

- A. regular
B. context-free but not regular
C. context-sensitive but not context free
D. type-0 but not context sensitive

gate2004 theory-of-computation normal identify-class-language

[Answer](#)

5.7.10 Identify Class Language: GATE2005-55 [top](#)

<https://gateoverflow.in/1378>

Consider the languages:

$$L_1 = \{a^n b^n c^m \mid n, m > 0\} \text{ and } L_2 = \{a^n b^m c^m \mid n, m > 0\}$$

Which one of the following statements is FALSE?

- A. $L_1 \cap L_2$ is a context-free language
B. $L_1 \cup L_2$ is a context-free language
C. L_1 and L_2 are context-free languages
D. $L_1 \cap L_2$ is a context sensitive language

gate2005 theory-of-computation identify-class-language normal

[Answer](#)

5.7.11 Identify Class Language: GATE2005-IT-4 [top](#)

<https://gateoverflow.in/3748>

Let L be a regular language and M be a context-free language, both over the alphabet Σ . Let L^c and M^c denote the complements of L and M respectively. Which of the following statements about the language $L^c \cup M^c$ is TRUE?

- A. It is necessarily regular but not necessarily context-free.
B. It is necessarily context-free.
C. It is necessarily non-regular.
D. None of the above

gate2005-it theory-of-computation normal identify-class-language

[Answer](#)

5.7.12 Identify Class Language: GATE2005-IT-6 [top](#)

<https://gateoverflow.in/3751>

The language $\{0^n 1^n 2^n \mid 1 \leq n \leq 10^6\}$ is

- A. regular
- B. context-free but not regular
- C. context-free but its complement is not context-free
- D. not context-free

gate2005-it theory-of-computation easy identify-class-language

[Answer](#)

5.7.13 Identify Class Language: GATE2006-30 [top](#)

<https://gateoverflow.in/993>

For $s \in (0+1)^*$ let $d(s)$ denote the decimal value of s (e.g. $d(101) = 5$). Let

$$L = \{s \in (0+1)^* \mid d(s) \bmod 5 = 2 \text{ and } d(s) \bmod 7 \neq 4\}$$

Which one of the following statements is true?

- A. L is recursively enumerable, but not recursive
- B. L is recursive, but not context-free
- C. L is context-free, but not regular
- D. L is regular

gate2006 theory-of-computation normal identify-class-language

[Answer](#)

5.7.14 Identify Class Language: GATE2006-33 [top](#)

<https://gateoverflow.in/996>

Let L_1 be a regular language, L_2 be a deterministic context-free language and L_3 a recursively enumerable, but not recursive, language. Which one of the following statements is false?

- A. $L_1 \cap L_2$ is a deterministic CFL
- B. $L_3 \cap L_1$ is recursive
- C. $L_1 \cup L_2$ is context free
- D. $L_1 \cap L_2 \cap L_3$ is recursively enumerable

gate2006 theory-of-computation normal identify-class-language

[Answer](#)

5.7.15 Identify Class Language: GATE2007-30 [top](#)

<https://gateoverflow.in/1228>

The language $L = \{0^i 2 1^i \mid i \geq 0\}$ over the alphabet $\{0, 1, 2\}$ is:

- A. not recursive
- B. is recursive and is a deterministic CFL
- C. is a regular language
- D. is not a deterministic CFL but a CFL

gate2007 theory-of-computation normal identify-class-language

[Answer](#)

5.7.16 Identify Class Language: GATE2008-9 [top](#)

<https://gateoverflow.in/407>

Which of the following is true for the language

$$\{a^p \mid p \text{ is a prime}\}?$$

- A. It is not accepted by a Turing Machine
- B. It is regular but not context-free
- C. It is context-free but not regular
- D. It is neither regular nor context-free, but accepted by a Turing machine

[gate2008](#) [theory-of-computation](#) [easy](#) [identify-class-language](#)

Answer

5.7.17 Identify Class Language: GATE2008-IT-33 [top](#)

<https://gateoverflow.in/3343>

Consider the following languages.

- $L_1 = \{a^i b^j c^k \mid i = j, k \geq 1\}$
- $L_2 = \{a^i b^j \mid j = 2i, i \geq 0\}$

Which of the following is true?

- A. L_1 is not a CFL but L_2 is
- B. $L_1 \cap L_2 = \emptyset$ and L_1 is non-regular
- C. $L_1 \cup L_2$ is not a CFL but L_2 is
- D. There is a 4-state PDA that accepts L_1 , but there is no DPDA that accepts L_2 .

[gate2008-it](#) [theory-of-computation](#) [normal](#) [identify-class-language](#)

Answer

5.7.18 Identify Class Language: GATE2009-40 [top](#)

<https://gateoverflow.in/1326>

Let $L = L_1 \cap L_2$, where L_1 and L_2 are languages as defined below:

$$L_1 = \{a^m b^m c a^n b^n \mid m, n \geq 0\}$$

$$L_2 = \{a^i b^j c^k \mid i, j, k \geq 0\}$$

Then L is

- A. Not recursive
- B. Regular
- C. Context free but not regular
- D. Recursively enumerable but not context free.

[gate2009](#) [theory-of-computation](#) [easy](#) [identify-class-language](#)

Answer

5.7.19 Identify Class Language: GATE2010-40 [top](#)

<https://gateoverflow.in/2341>

Consider the languages

$$L_1 = \{0^i 1^j \mid i \neq j\},$$

$$L2 = \{0^i 1^j \mid i = j\},$$

$$L3 = \{0^i 1^j \mid i = 2j + 1\},$$

$$L4 = \{0^i 1^j \mid i \neq 2j\}$$

- A. Only $L2$ is context free.
- B. Only $L2$ and $L3$ are context free.
- C. Only $L1$ and $L2$ are context free.
- D. All are context free

[gate2010](#) [theory-of-computation](#) [context-free-language](#) [identify-class-language](#) [normal](#)

[Answer](#)

5.7.20 Identify Class Language: GATE2011-26 [top](#)

<https://gateoverflow.in/2128>

Consider the languages $L1$, $L2$ and $L3$ as given below.

$$L1 = \{0^p 1^q \mid p, q \in N\}, L2 = \{0^p 1^q \mid p, q \in N \text{ and } p = q\} \text{ and } L3 = \{0^p 1^q 0^r \mid p, q, r \in N \text{ and } p = q = r\}$$

Which of the following statements is **NOT TRUE**?

- A. Push Down Automata (PDA) can be used to recognize $L1$ and $L2$
- B. $L1$ is a regular language
- C. All the three languages are context free
- D. Turing machines can be used to recognize all the languages

[gate2011](#) [theory-of-computation](#) [identify-class-language](#) [normal](#)

[Answer](#)

5.7.21 Identify Class Language: GATE2013-32 [top](#)

<https://gateoverflow.in/1543>

Consider the following languages.

$$L1 = \{0^p 1^q 0^r \mid p, q, r \geq 0\}$$

$$L2 = \{0^p 1^q 0^r \mid p, q, r \geq 0, p \neq r\}$$

Which one of the following statements is **FALSE**?

- A. $L2$ is context-free.
- B. $L1 \cap L2$ is context-free.
- C. Complement of $L2$ is recursive.
- D. Complement of $L1$ is context-free but not regular.

[gate2013](#) [theory-of-computation](#) [identify-class-language](#) [normal](#)

[Answer](#)

5.7.22 Identify Class Language: GATE2014-3-36 [top](#)

<https://gateoverflow.in/2070>

Consider the following languages over the alphabet $\Sigma = \{0, 1, c\}$

$$L1 = \{0^n 1^n \mid n \geq 0\}$$

$$L2 = \{wcw^r \mid w \in \{0, 1\}^*\}$$

$$L_3 = \{ww^r \mid w \in \{0,1\}^*\}$$

Here, w^r is the reverse of the string w . Which of these languages are deterministic Context-free languages?

- A. None of the languages
- B. Only L_1
- C. Only L_1 and L_2
- D. All the three languages

gate2014-3 theory-of-computation identify-class-language context-free-language normal

[Answer](#)

5.7.23 Identify Class Language: GATE2017-1-37 [top](#)

<https://gateoverflow.in/118320>

Consider the context-free grammars over the alphabet $\{a,b,c\}$ given below. S and T are non-terminals.

$$G_1 : S \rightarrow aSb \mid T, T \rightarrow cT \mid \epsilon$$

$$G_2 : S \rightarrow bSa \mid T, T \rightarrow cT \mid \epsilon$$

The language $L(G_1) \cap L(G_2)$ is

- A. Finite
- B. Not finite but regular
- C. Context-Free but not regular
- D. Recursive but not context-free

gate2017-1 theory-of-computation context-free-language identify-class-language normal

[Answer](#)

5.7.24 Identify Class Language: GATE2017-2-40 [top](#)

<https://gateoverflow.in/118615>

Consider the following languages.

- $L_1 = \{a^p \mid p \text{ is a prime number}\}$
- $L_2 = \{a^n b^m c^{2m} \mid n \geq 0, m \geq 0\}$
- $L_3 = \{a^n b^n c^{2n} \mid n \geq 0\}$
- $L_4 = \{a^n b^n \mid n \geq 1\}$

Which of the following are CORRECT?

- I. L_1 is context free but not regular
 - II. L_2 is not context free
 - III. L_3 is not context free but recursive
 - IV. L_4 is deterministic context free
-
- A. I, II and IV only
 - B. II and III only
 - C. I and IV only
 - D. III and IV only

gate2017-2 theory-of-computation identify-class-language

[Answer](#)

5.7.25 Identify Class Language: GATE2018-35 [top](#)

<https://gateoverflow.in/204109>

Consider the following languages:

- I. $\{a^m b^n c^p d^q \mid m + p = n + q, \text{ where } m, n, p, q \geq 0\}$
- II. $\{a^m b^n c^p d^q \mid m = n \text{ and } p = q, \text{ where } m, n, p, q \geq 0\}$
- III. $\{a^m b^n c^p d^q \mid m = n = p \text{ and } p \neq q, \text{ where } m, n, p, q \geq 0\}$
- IV. $\{a^m b^n c^p d^q \mid mn = p + q, \text{ where } m, n, p, q \geq 0\}$

Which of the above languages are context-free?

- A. I and IV only
- B. I and II only
- C. II and III only
- D. II and IV only

[gate2018](#) [theory-of-computation](#) [identify-class-language](#) [context-free-language](#) [normal](#)

Answer

5.7.26 Identify Class Language: TIFR2010-B-22 [top](#)

<https://gateoverflow.in/18622>

Let L consist of all binary strings beginning with a 1 such that its value when converted to decimal is divisible by 5. Which of the following is true?

- A. L can be recognized by a deterministic finite state automaton.
- B. L can be recognized by a non-deterministic finite state automaton but not by a deterministic finite state automaton.
- C. L can be recognized by a deterministic push-down automaton but not by a non-deterministic finite state automaton.
- D. L can be recognized by a non-deterministic push-down automaton but not by a deterministic push-down automaton.
- E. L cannot be recognized by any push-down automaton.

[tifr2010](#) [theory-of-computation](#) [identify-class-language](#)

Answer

5.7.27 Identify Class Language: TIFR2010-B-35 [top](#)

<https://gateoverflow.in/19247>

Consider the following languages over the alphabet $\{0, 1\}$.

$$L1 = \{x \cdot x^R \mid x \in \{0, 1\}^*\}$$

$$L2 = \{x \cdot x \mid x \in \{0, 1\}^*\}$$

Where x^R is the reverse of string x ; e.g. $011^R = 110$. Which of the following is true?

- A. Both $L1$ and $L2$ are regular.
- B. $L1$ is context-free but not regular whereas $L2$ is regular.
- C. Both $L1$ and $L2$ are context free and neither is regular.
- D. $L1$ is context free but $L2$ is not context free.
- E. Both $L1$ and $L2$ are not context free.

[tifr2010](#) [theory-of-computation](#) [identify-class-language](#)

Answer

5.7.28 Identify Class Language: TIFR2012-B-18 [top](#)

<https://gateoverflow.in/25216>

Let a^i denote a sequence $a.a\dots a$ with i letters and let \mathbb{N} be the set of natural numbers $1, 2, \dots$. Let $L_1 = \{a^i b^{2i} \mid i \in \mathbb{N}\}$ and $L_2 = \{a^i b^{i^2} \mid i \in \mathbb{N}\}$ be two languages. Which of the following is correct?

- A. Both L_1 and L_2 are context-free languages.
- B. L_1 is context-free and L_2 is recursive but not context-free.
- C. Both L_1 and L_2 are recursive but not context-free.
- D. L_1 is regular and L_2 is context-free.
- E. Complement of L_2 is context-free.

[tifr2012](#) [theory-of-computation](#) [identify-class-language](#)

Answer

5.7.29 Identify Class Language: TIFR2014-B-13 [top](#)

<https://gateoverflow.in/27320>

Let L be a given context-free language over the alphabet $\{a, b\}$. Construct L_1, L_2 as follows. Let $L_1 = L - \{xyx \mid x, y \in \{a, b\}^*\}$, and $L_2 = L \cdot L$. Then,

- A. Both L_1 and L_2 are regular.
- B. Both L_1 and L_2 are context free but not necessarily regular.
- C. L_1 is regular and L_2 is context free.
- D. L_1 and L_2 both may not be context free.
- E. L_1 is regular but L_2 may not be context free.

[tifr2014](#) [theory-of-computation](#) [identify-class-language](#)

Answer

5.7.30 Identify Class Language: TIFR2015-B-8 [top](#)

<https://gateoverflow.in/29865>

Let $\Sigma_1 = \{a\}$ be a one letter alphabet and $\Sigma_2 = \{a, b\}$ be a two letter alphabet. A language over an alphabet is a set of finite length words comprising letters of the alphabet. Let L_1 and L_2 be the set of languages over Σ_1 and Σ_2 respectively. Which of the following is true about L_1 and L_2 :

- A. Both are finite.
- B. Both are countably infinite.
- C. L_1 is countable but L_2 is not.
- D. L_2 is countable but L_1 is not.
- E. Neither of them is countable.

[tifr2015](#) [identify-class-language](#)

Answer

5.7.31 Identify Class Language: TIFR2017-B-14 [top](#)

<https://gateoverflow.in/95825>

Consider the following grammar G with terminals $\{[,]\}$, start symbol S , and non-terminals $\{A, B, C\}$:

$$S \rightarrow AC \mid SS \mid AB$$

$$C \rightarrow SB$$

$$A \rightarrow [$$

$$B \rightarrow]$$

A language L is called prefix-closed if for every $x \in L$, every prefix of x is also in L . Which of the following is FALSE?

- A. $L(G)$ is context free
- B. $L(G)$ is infinite
- C. $L(G)$ can be recognized by a deterministic push down automaton
- D. $L(G)$ is prefix-closed
- E. $L(G)$ is recursive

tifr2017 theory-of-computation identify-class-language

Answer

5.7.32 Identify Class Language: TIFR2017-B-4 [top](#)

<https://gateoverflow.in/95680>

Let L be the language over the alphabet $\{1, 2, 3, (,)\}$ generated by the following grammar (with start symbol S , and non-terminals $\{A, B, C\}$):

$$S \rightarrow ABCA \rightarrow (B \rightarrow 1B \mid 2B \mid 3BB \rightarrow 1 \mid 2 \mid 3C \rightarrow)$$

Then, which of the following is TRUE?

- A. L is finite
- B. L is not recursively enumerable
- C. L is regular
- D. L contains only strings of even length
- E. L is context-free but not regular

tifr2017 theory-of-computation identify-class-language

Answer

5.7.33 Identify Class Language: TIFR2018-B-11 [top](#)

<https://gateoverflow.in/179295>

Consider the language $L \subseteq \{a, b, c\}^*$ defined as

$$L = \{a^p b^q c^r : p = q \text{ or } q = r \text{ or } r = p\}.$$

Which of the following answer is TRUE about complexity of this language?

- A. L is regular but not context-free
- B. L is context-free but not regular
- C. L is decidable but not context free
- D. The complement of L , defined as $\overline{L} = \{a, b, c\}^*/L$, is regular.
- E. L is regular, context-free and decidable

tifr2018 identify-class-language theory-of-computation

Answer

Answers: Identify Class Language

5.7.1 Identify Class Language: CMI2010-A-01 [top](#)

<https://gateoverflow.in/46131>



(B) L is regular.

1 upvote

-- Praveen Saini (54.6k points)

5.7.2 Identify Class Language: GATE1987-1-xiii [top](#)

<https://gateoverflow.in/80293>



C. Context sensitive language.

Due to presence of some features which cannot be handled by **PDA**.

some of the features are:

1. Variable declared before use
2. Matching formal and actual parameters of functions

1 upvote

-- Ravit Anand (181 points)

5.7.3 Identify Class Language: GATE1988-2ix [top](#)

<https://gateoverflow.in/93949>



Here n is finite and finite language must be regular.

If n is not restricted, then it would be **DCFL**.

1 upvote

-- kirti singh (3.8k points)

5.7.4 Identify Class Language: GATE1991-17,a [top](#)

<https://gateoverflow.in/26653>

https://en.wikipedia.org/wiki/Read-only_Turing_machine

1 upvote

-- srestha (87k points)

5.7.5 Identify Class Language: GATE1994-19 [top](#)

<https://gateoverflow.in/2515>



(b) No. Need not be. Take $L_2 = \{a^n b^n \mid n > 0\}$ and $L_1 = \text{all strings over } \{a, b\}$. Now, $L_1 \cup L_2$ is L_1 only and is regular but L_2 is not regular.

1 upvote

-- Arjun Suresh (350k points)

5.7.6 Identify Class Language: GATE1999-2.4 [top](#)

<https://gateoverflow.in/1482>



L_2 is regular, so complement of L_2 , ($\sim L_2$), is also regular .

Regular languages under complement. So, **D is True.**

$L_1 \cap L_2$ is context free.

Intersection of Context free language with Regular language is Context free. So, **B is True.**

$L_1 - L_2 = L_1 \cap (\sim L_2)$ is context free

Intersection of Context free language with Regular language is Context free. So, **A is False .**

$\sim L_1$ is not context free

Context free languages are not closed under complement. So **C is False (May/not be).**

14 votes

-- Praveen Saini (54.6k points)

5.7.7 Identify Class Language: GATE2000-1.5 top

<https://gateoverflow.in/628>



Selected Answer

B. is the answer for this question

Language generated by this grammar is $L = (00)^+$ that is regular but not 0^+

14 votes

-- Manu Thakur (39.6k points)

5.7.8 Identify Class Language: GATE2002-1.7 top

<https://gateoverflow.in/811>



Selected Answer

B. Regular.

With only finite positions in stack, we can have only finite configurations and these can also be modeled as states in a finite automata.

14 votes

-- Arjun Suresh (350k points)

5.7.9 Identify Class Language: GATE2004-87 top

<https://gateoverflow.in/1081>



Selected Answer

Language is not regular bcoz we need to match count of c 's is equal to count of b 's + count of a 's and that can implement by **PDA**.

$\delta(q_0, a, \epsilon) = (q_0, a)$ [push a in stack, as per language a comes first]

$\delta(q_0, a, a) = (q_0, aa)$ [push all a's into stack]

$\delta(q_0, b, a) = (q_1, ba)$ [push b in stack, state change to q_1 that sure b comes after a]

$\delta(q_1, b, b) = (q_1, bb)$ [push all b's in stack]

$\delta(q_1, c, b) = (q_2, \epsilon)$ [pop one b for one c]

$\delta(q_2, c, b) = (q_2, \epsilon)$ [pop one b 's for each c and continue same]

$\delta(q_2, c, a) = (q_3, \epsilon)$ [pop one a for one c , when there is no more b in stack]

$\delta(q_3, c, a) = (q_3, \epsilon)$ [pop one a for each c and continue same]

$\delta(q_3, \epsilon, \epsilon) = (q_f, \epsilon)$ [if sum of c 's is sum of a 's and b 's then stack will be empty when there is no c in input]

Answer is **option B** : language is context-free but not regular.

Note : 1 state changes make sure b 's comes after a and c 's comes after b 's]

15 votes

-- Praveen Saini (54.6k points)

5.7.10 Identify Class Language: GATE2005-55 top

<https://gateoverflow.in/1378>



Selected Answer

L_1 is **CFL** [put a 's in stack , and pop a with each b]]

L_2 is **CFL** [put b 's in stack and pop b with each c]

C) is True.

B) is True. **CFL** is closed under Union [$S \rightarrow S_1 \mid S_2$ where S_1 is grammar for L_1 and S_2 for L_2]

CFL is not closed under Intersection, so intersection of two CFLs may or may not be CFL. Lets examine:

$L_1 \cap L_2 = \{a^i b^i c^i, i > 0\}$ which is Context sensitive but not context free [can't match a 's, b 's and c 's with one stack]

So, **A**) is False.

D) is True.

24 votes

-- Praveen Saini (54.6k points)

5.7.11 Identify Class Language: GATE2005-IT-4 top

<https://gateoverflow.in/3748>



Selected Answer

Take $L = \Sigma^*$ then $L^c = \emptyset$ and $M^c \cup \emptyset = M^c$

We know that complement of CFL need not be a CFL as CFL is not closed under complement.

So, (A) and (B) are false.

If we take $L = \emptyset$ then $L^c = \Sigma^*$ and $M^c \cup \Sigma^* = \Sigma^*$ which is regular - (C) is also false.

So, answer (**D**)

45 votes

-- Arjun Suresh (350k points)

5.7.12 Identify Class Language: GATE2005-IT-6 top

<https://gateoverflow.in/3751>

Selected Answer

Regular (in fact finite). Since n is finite, we have a finite set of strings satisfying the given conditions. So, we can easily make a finite automata for those strings.

18 votes

-- Arjun Suresh (350k points)

5.7.13 Identify Class Language: GATE2006-30 top

<https://gateoverflow.in/993>

Selected Answer

Refer this:

https://gateoverflow.in/1695/gate1998_4

$L_1 = \{s \in (0+1)^* \mid d(s) \text{ mod } 5 = 2\}$ is regular

Having 2 as final state out of $\{0,1,2,3,4\}$ states

As given in example in posted link [in same DFA , final state will be 2 instead of 0]

Similarly, $L_2 = \{s \in (0+1)^* \mid d(s) \text{ mod } 7 \neq 4\}$ is also regular

Having states $\{0,1,2,3,4,5,6\}$ and all are final state except 4

$L_1 \cap L_2$ is L (given problem) is also regular

As regular languages are closed under intersection. D is correct option.

35 votes

-- Praveen Saini (54.6k points)

5.7.14 Identify Class Language: GATE2006-33 top

<https://gateoverflow.in/996>

Selected Answer

- A. True : DCFL are closed under Intersection with Regular Languages
- B. False : L_1 is recursive hence also decidable, L_3 is RE but not Recursive hence it is undecidable. Intersection of Recursive language and Recursive Enumerable language is Recursive Enumerable language.
- C. True : L_1 is regular hence also CFL and every DCFL is also CFL and All CFL are closed under Union.
- D. True : L_1 is regular hence also RE; L_2 is DCFL hence also RE; RE languages are closed under Intersection

35 votes

-- Danish (3.8k points)

5.7.15 Identify Class Language: GATE2007-30 top

<https://gateoverflow.in/1228>

Selected Answer

$L = \{0^i 21^i \mid i \geq 0\}$ has only one comparison that can be done using a DPDA. Hence, its DCFL.

Context free languages are a proper subset of Recursive Languages. \therefore it is recursive too.

Answer is **option B.**

Upvote 20 votes

-- Amar Vashishth (30.5k points)

5.7.16 Identify Class Language: GATE2008-9 top

<https://gateoverflow.in/407>



Selected Answer

We have [algorithms to generate prime numbers](#) \implies we can generate sequence of p for the given language, hence strings as defined by the language definition.

So, by Church Turing Thesis we can say that there exists a Turing Machine which can accept the given language.

Answer is **option D.**

Upvote 18 votes

-- Amar Vashishth (30.5k points)

5.7.17 Identify Class Language: GATE2008-IT-33 top

<https://gateoverflow.in/3343>



Selected Answer

Both languages can be accepted by a **DPDA**:

L_1 = start pushing element X into the stack on input 'a' ... start popping X on input 'b' ... move to final state when stack empty and input = 'c'

L_2 = start pushing elements XX into the stack on input 'a' ... start popping X on input 'b' ... move to final state when stack empty and input = 'epsilon'

So, **(A)** and **(D)** are False.

$L_1 \cup L_2$ is a CFL ... we can build it by having L_1, L_2 and an extra state ... and an 'epsilon' transition to both L_1 and L_2 from that extra state.

So, **(C)** is false.

$L_1 \cap L_2 = \Phi$ because we have no string $a^i b^j$ where $i = j$ and $i = 2j$ for $i, j \geq 1$

and clearly L_1 is not a regular language

So, **(B)** is true.

Upvote 28 votes

-- Danish (3.8k points)

5.7.18 Identify Class Language: GATE2009-40 top

<https://gateoverflow.in/1326>



Selected Answer

Answer is C.

$$L_1 = \{a^m b^m c a^n b^n \mid m, n \geq 0\}$$

It is a CFL. It will generate strings which start with a's followed by equal number of b's and single c followed by a's followed by equal number of b's

i.e., $abcab, aabbcab, aaabbcaabaabb, \dots$

$$L_2 = \{a^i b^j c^k \mid i, j, k \geq 0\}$$

It is a Regular (and of course CFL). It will generate strings which begin with a's followed by any number of b's followed by any number of c's

So, $L_1 \cap L_2$ will be **all strings which are common in both languages**

So, those will be the strings which begin with a's followed by equal number of b's and ending with c

They would be of the form $a^m b^m c$

Such a language is not regular as it requires equal number of a's and b's. But it is Context Free language as we can make a PDA for it. Also, CFLs are closed under intersection with regular set. So, any CFL intersection a regular set gives a CFL (may or may not be a regular set).

13 votes

-- Arunav Khare (5.1k points)

$L_1 \cap L_2 = \{a^m b^m c \mid m \geq 0\}$, which is context free but not regular.

Option C.

36 votes

-- Arjun Suresh (350k points)

5.7.19 Identify Class Language: GATE2010-40 top

<https://gateoverflow.in/2341>



Selected Answer

All are context free.

$L_1 \rightarrow$ Push 0 on stack and when 1 comes, start popping. If stack becomes empty and 1's are remaining start pushing 1. At end of string accept if stack is non- empty.

$L_2 \rightarrow$ Do the same as for L_1 , but accept if stack is empty at end of string.

$L_3 \rightarrow$ Do the same as for L_2 , but for each 0, push two 0's on stack and start the stack with a 0.

$L_4 \rightarrow$ Do the same as for L_1 , but for each 0, push two 0's on stack

All are in fact DCFL. Pushing two 0's on stack might sound non-trivial but we can do this by pushing one symbol and going to a new state. Then on this new state on empty symbol, push one more symbol on stack and come back.

29 votes

-- Arjun Suresh (350k points)

5.7.20 Identify Class Language: GATE2011-26 top

<https://gateoverflow.in/2128>



Selected Answer

Answer is **C.**

L_1 is **RL**

L_2 is **CFL**

L_3 is **CSL**

Turning Machine is powerful Machine it can be used to accept all the languages (RL,CFL,CSL,RE)

Upvote 24 votes

-- Sona Praneeth Akula (4.2k points)

5.7.21 Identify Class Language: GATE2013-32 top

<https://gateoverflow.in/1543>



Selected Answer

L_1 is regular and hence context-free also. Regular expression for L_1 is $0^*1^*0^*$. So, (D) is the false choice.

L_2 is context-free but not regular. We need a stack to count if the number of 0's before and after the 1 (1 may be absent also) are not same. So, $L_1 \cap L_2$ is context-free as regular \cap context-free is context-free. → (B) is true.

Complement of L_2 is recursive as context-free complement is always recursive (actually even context-sensitive).

Upvote 30 votes

-- Arjun Suresh (350k points)

5.7.22 Identify Class Language: GATE2014-3-36 top

<https://gateoverflow.in/2070>



Selected Answer

C.

L_3 is CFL and not DCFL as in no way we can deterministically determine the MIDDLE point of the input string.

Upvote 26 votes

-- Gate Keeda (19.6k points)

5.7.23 Identify Class Language: GATE2017-1-37 top

<https://gateoverflow.in/118320>



Selected Answer

Since while intersection all strings produced by production aSb in G_1 and bSa in G_2 will be 0

So, only common production will be:

$S \rightarrow T$

$T \rightarrow cT \mid \epsilon$

Which is nothing but c^* hence it is REGULAR and INFINITE

So, option is (B).

1 30 votes

-- sriv_shubham (3.3k points)

5.7.24 Identify Class Language: GATE2017-2-40 top <https://gateoverflow.in/118615>



L_1 is CSL, L_2 is context free

L_3 is not Context free and L_4 is DCFL

So, option is D.

1 20 votes

-- Swapnil (1.8k points)

5.7.25 Identify Class Language: GATE2018-35 top <https://gateoverflow.in/204109>



I and II is CFL since can be done via single stack.

III and IV is CSL.

B is answer.

1 22 votes

-- Prashant Singh (59.8k points)

5.7.26 Identify Class Language: TIFR2010-B-22 top <https://gateoverflow.in/18622>



I can be recognized by a DFA. we have a DFA to accept all such strings which when interpreted as decimal number are divisible by n . Where n can be anything the DFA of such can be made by a trick.

States are equal to possible remainders

0	1
---	---

q_0	q_0	q_1
-------	-------	-------

q_1	q_2	q_3
-------	-------	-------

q_2	q_4	q_0
-------	-------	-------

q_3	q_1	q_2
-------	-------	-------

q_4	q_3	q_4
-------	-------	-------

If you can see the symmetry in it. write states and make fill like $q_0 \ q_1 \ q_2 \ q_3 \dots$

Now, it is saying that it has to always start with 1 which the above DFA will not satisfy so make it a

nfa by making a transition from q_0 on zero to a new dead state. now you have a nfa reduce it which will result in a deterministic **DFA**.

So, option is **A**.

Upvote 9 votes

-- Ravi Singh (15.7k points)

5.7.27 Identify Class Language: TIFR2010-B-35 top <https://gateoverflow.in/19247>



L1 = $\{x \cdot x^R \mid x \in \{0,1\}^*\}$ Its even palindrome so it is CFL
L2 = $\{x \cdot x \mid x \in \{0,1\}^*\}$ Its string matching so it is CSL but not CFL

Option (D) **L1** is context free but **L2** is not context free.

Upvote 11 votes

-- Umang Raman (15.7k points)

5.7.28 Identify Class Language: TIFR2012-B-18 top <https://gateoverflow.in/25216>



L_1 - CFL, $S \rightarrow aSbb \mid abb$

L_2 - Not CFL ,we can't count i^2 using **CFL**.

- False because L_2 is not **CFL**.
- True.** L_2 is recursive
- False because L_1 is **CFL**.
- False, L_1 not regular.
- False, as complement of L_2 is also not context free. It still need to computer i^2 for checking for inequality.

Answer is **B**.

Upvote 7 votes

-- Akash Kanase (42.5k points)

5.7.29 Identify Class Language: TIFR2014-B-13 top <https://gateoverflow.in/27320>



L is a context free language over $\{a,b\}$
 $L_1 = L - \{xyx \mid x, y \in \{a,b\}^*\}$

$= L - \{\text{all strings over } \{a,b\}\}$ [Note: all strings can be generated from y by putting $x = \epsilon$]

$= L - (a+b)^* = \{\}$, empty set. [Note : $L_1 - L_2 = \{\text{string in } L_1 \text{ but not in } L_2\}$]

So , L_1 is a Regular Language.

L is a context free language over $\{a,b\}$

$L_2 = L \cdot L$

Context free languages are closed under Concatenation.

So, L_2 is Context Free Language.

Option C is correct.

30 votes

-- Praveen Saini (54.6k points)

5.7.30 Identify Class Language: TIFR2015-B-8 [top](#)



Selected Answer

Languages over alphabet set are uncountable so, answer should be (E).

7 votes

-- Pooja Palod (31.3k points)

5.7.31 Identify Class Language: TIFR2017-B-14 [top](#)



Selected Answer

The given grammar generates balanced parenthesis.

Lets take a smallest string : [[]] (say x)

Prefixes of x are : [, [, [[]

BUT they don't belong to the language generated by the given grammar.

So, the answer will be **Option D**.

Correct me if am wrong.

12 votes

-- sarveswara rao vangala (2.1k points)

5.7.32 Identify Class Language: TIFR2017-B-4 [top](#)



Selected Answer

Language accepted by that grammar is:

$((1+2+3)^+)$

Hence, regular!

10 votes

-- Motamarri Anusha (12.3k points)

5.7.33 Identify Class Language: TIFR2018-B-11 [top](#)

$$L1 = a^p b^q c^r : p = q$$

$$L2 = a^p b^q c^r : q = r$$

$$L3 = a^p b^q c^r : r = p$$

$$L = L1 \cup L2 \cup L3$$

L1, L2, L3 are CFL

We know that union of CFL is CFL
Hence, L is Context free language

Answer is option B.

4 votes

-- Rameez Raza (2.3k points)

5.8

Minimal State Automata(2) top

5.8.1 Minimal State Automata: GATE1997-70 top

<https://gateoverflow.in/19700>

Following is a state table for time finite state machine.

Present State	Next State Output	
	Input - 0	Input - 1
A	B.1	H.1
B	F.1	D.1
C	D.0	E.1
D	C.0	F.1
E	D.1	C.1
F	C.1	C.1
G	C.1	D.1
H	C.0	A.1

- A. Find the equivalence partition on the states of the machine.
- B. Give the state table for the minimal machine. (Use appropriate names for the equivalent states. For example if states X and Y are equivalent then use XY as the name for the equivalent state in the minimal machine).

gate1997 theory-of-computation minimal-state-automata

Answer

5.8.2 Minimal State Automata: GATE2018-6 top

<https://gateoverflow.in/204080>

Let N be an NFA with n states. Let k be the number of states of a minimal DFA which is equivalent to N . Which one of the following is necessarily true?

- A. $k \geq 2^n$
- B. $k \geq n$
- C. $k \leq n^2$
- D. $k \leq 2^n$

gate2018 theory-of-computation minimal-state-automata normal

Answer

Answers: Minimal State Automata

5.8.1 Minimal State Automata: GATE1997-70 top

<https://gateoverflow.in/19700>



Selected Answer

As nothing is mentioned in question, assuming that the first state itself is the start state i.e. State A so, State G is not reachable from the start and hence is removed before applying the minimization algorithm.

$$P_0 \rightarrow [ABEFG][CDH] \rightarrow [ABEF][CDH]$$

$$P_1 \rightarrow [AB][EF][CDH]$$

$$P_2 \rightarrow [A][B][EF][CD][H]$$

$$P_3 \rightarrow [A][B][EF][CD][H] // P_2 == P_3 \text{ So, Stop}$$

State table for the minimal machine ::

Present State	input= 0	input= 1
A	$B.1$	$H.1$
B	$EF.1$	$CD.1$
CD	$CD.0$	$EF.1$
EF	$CD.1$	$CD.1$
H	$CD.0$	$A.1$

👍 9 votes

-- Vidhi Sethi (8.7k points)

5.8.2 Minimal State Automata: GATE2018-6 top

<https://gateoverflow.in/204080>



Selected Answer

Number of states in minimal DFA - k must be $\leq 2^n$ as each state corresponds to a subset of states of the corresponding NFA.

D is answer.

Option B is not always TRUE as the NFA N can have epsilon moves and hence can have more number of states than the minimal DFA.

👍 10 votes

-- Prashant Singh (59.8k points)

5.9

Non Determinism(7) top

5.9.1 Non Determinism: GATE1992-02,xx top

<https://gateoverflow.in/577>

02. Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

In which of the cases stated below is the following statement true?

"For every non-deterministic machine M_1 there exists an equivalent deterministic machine M_2 recognizing the same language".

- A. M_1 is non-deterministic finite automaton.
- B. M_1 is non-deterministic PDA.
- C. M_1 is a non-deterministic Turing machine.
- D. For no machines M_1 and M_2 , the above statement true.

gate1992 theory-of-computation easy non-determinism

[Answer](#)

5.9.2 Non Determinism: GATE1994-1.16 [top](#)

<https://gateoverflow.in/2459>

Which of the following conversions is not possible (algorithmically)?

- A. Regular grammar to context free grammar
- B. Non-deterministic FSA to deterministic FSA
- C. Non-deterministic PDA to deterministic PDA
- D. Non-deterministic Turing machine to deterministic Turing machine

gate1994 theory-of-computation easy non-determinism

[Answer](#)

5.9.3 Non Determinism: GATE1998-1.11 [top](#)

<https://gateoverflow.in/1648>

Regarding the power of recognition of languages, which of the following statements is false?

- A. The non-deterministic finite-state automata are equivalent to deterministic finite-state automata.
- B. Non-deterministic Push-down automata are equivalent to deterministic Push-down automata.
- C. Non-deterministic Turing machines are equivalent to deterministic Push-down automata.
- D. Non-deterministic Turing machines are equivalent to deterministic Turing machines.
- E. Multi-tape Turing machines are available are equivalent to Single-tape Turing machines.

gate1998 theory-of-computation easy non-determinism

[Answer](#)

5.9.4 Non Determinism: GATE2004-IT-9 [top](#)

<https://gateoverflow.in/3650>

Which one of the following statements is FALSE?

- A. There exist context-free languages such that all the context-free grammars generating them are ambiguous
- B. An unambiguous context-free grammar always has a unique parse tree for each string of the language generated by it
- C. Both deterministic and non-deterministic pushdown automata always accept the same set of languages
- D. A finite set of strings from some alphabet is always a regular language

gate2004-it theory-of-computation easy non-determinism

[Answer](#)

5.9.5 Non Determinism: GATE2005-54 top

<https://gateoverflow.in/1377>

Let N_f and N_p denote the classes of languages accepted by non-deterministic finite automata and non-deterministic push-down automata, respectively. Let D_f and D_p denote the classes of languages accepted by deterministic finite automata and deterministic push-down automata respectively. Which one of the following is TRUE?

- A. $D_f \subset N_f$ and $D_p \subset N_p$
- B. $D_f \subset N_f$ and $D_p = N_p$
- C. $D_f = N_f$ and $D_p = N_p$
- D. $D_f = N_f$ and $D_p \subset N_p$

gate2005 theory-of-computation easy non-determinism

Answer

5.9.6 Non Determinism: GATE2009-16, ISRO2017-12 top

<https://gateoverflow.in/1308>

Which one of the following is FALSE?

- A. There is a unique minimal DFA for every regular language
- B. Every NFA can be converted to an equivalent PDA.
- C. Complement of every context-free language is recursive.
- D. Every nondeterministic PDA can be converted to an equivalent deterministic PDA.

gate2009 theory-of-computation easy isro2017 non-determinism

Answer

5.9.7 Non Determinism: GATE2011-8 top

<https://gateoverflow.in/2110>

Which of the following pairs have **DIFFERENT** expressive power?

- A. Deterministic finite automata (DFA) and Non-deterministic finite automata (NFA)
- B. Deterministic push down automata (DPDA) and Non-deterministic push down automata (NPDA)
- C. Deterministic single tape Turing machine and Non-deterministic single tape Turing machine
- D. Single tape Turing machine and multi-tape Turing machine

gate2011 theory-of-computation easy non-determinism

Answer

Answers: Non Determinism

5.9.1 Non Determinism: GATE1992-02,xx top

<https://gateoverflow.in/577>



Selected Answer

Answer: **A** and **C**.

- For every NFA there exists a DFA.
- For every NPDA there does not exist a deterministic PDA.
- Every nondeterministic Turing machine has an equivalent deterministic Turing Machine.

14 votes

-- Rajarshi Sarkar (34.1k points)

5.9.2 Non Determinism: GATE1994-1.16 top

<https://gateoverflow.in/2459>



Selected Answer

This will be **C**. Because if that would have been possible then NPDA and DPDA must had same powers, which is not the case. You can take example of NFA and DFA. Both are convertible to each other and hence share the same power.

18 votes

-- Gate Keeda (19.6k points)

5.9.3 Non Determinism: GATE1998-1.11 top

<https://gateoverflow.in/1648>



Selected Answer

- True. Proof - Subset Construction Procedure
- False. No conversion from **NPDA** To **DPDA**
- False. **Power(TM) > NPDA > DPDA.**
- True
- True

Answer: This question has multiple answers, **B** and **C**.

17 votes

-- Akash Kanase (42.5k points)

5.9.4 Non Determinism: GATE2004-IT-9 top

<https://gateoverflow.in/3650>



Selected Answer

- This is true for inherently ambiguous language
- Always correct, that's why called un ambiguous
- NPDA is a superset of DPDA, hence it's **FALSE**
- Finite language is always regular

25 votes

-- Manu Thakur (39.6k points)

5.9.5 Non Determinism: GATE2005-54 top

<https://gateoverflow.in/1377>



Selected Answer

Option **D**.

NFA and DFA both have equivalent power.(every nfa can be converted into equivalent DFA) but NPDA can accept more languages than DPDA.

17 votes

-- shreya ghosh (3.5k points)

5.9.6 Non Determinism: GATE2009-16, ISRO2017-12 [top](#)



Selected Answer

<https://gateoverflow.in/1308>

Option **D. NDPA** is more powerful than **DPDA**, so they are not equivalent. Actually, **DPDA** is a proper subset of **NDPA**.

C is TRUE as **CFL** is a proper subset of recursive languages and recursive languages are closed under complement.

Upvote: 28 votes

-- Bhagirathi Nayak (14.1k points)

5.9.7 Non Determinism: GATE2011-8 [top](#)

<https://gateoverflow.in/2110>


Selected Answer

Expressing power of any machine can be defined as the maximum number of languages it can accept..if machine M_1 can accept more languages then M_2 then we can say that expressing power of M_1 is greater then M_2 .

- Languages accepted by NFA,will also be accepted by DFA because we can make DFA corresponding to NFA. So their expressing power is same.
- In this case languages accepted by NPDA is more then DPDA, so expressing power of NPDA is more then DPDA
- Both deterministic and non deterministic turing can accept same language.so there expressing power is same.
- In turing machine if we increase the number of tape then also language accepted by that machine is same as single tape turing machine.so there expressing power is same.

Answer is **B**.

Upvote: 24 votes

-- neha pawar (4.2k points)

5.10

P Np Npc Nph(5) [top](#)

5.10.1 P Np Npc Nph: GATE2005-58 [top](#)

<https://gateoverflow.in/1381>

Consider the following two problems on undirected graphs:

- α : Given $G(V, E)$, does G have an independent set of size $|V| - 4$?
- β : Given $G(V, E)$, does G have an independent set of size 5?

Which one of the following is TRUE?

- α is in P and β is NP-complete
- α is NP-complete and β is in P
- Both α and β are NP-complete
- Both α and β are in P

gate2005 theory-of-computation p-np-npc-nph normal

Answer**5.10.2 P Np Npc Nph: GATE2006-31** [top](#)<https://gateoverflow.in/994>

Let SHAM_3 be the problem of finding a Hamiltonian cycle in a graph $G = (V, E)$ with $|V|$ divisible by 3 and DHAM_3 be the problem of determining if a Hamiltonian cycle exists in such graphs. Which one of the following is true?

- A. Both DHAM_3 and SHAM_3 are NP-hard
- B. SHAM_3 is NP-hard, but DHAM_3 is not
- C. DHAM_3 is NP-hard, but SHAM_3 is not
- D. Neither DHAM_3 nor SHAM_3 is NP-hard

gate2006 theory-of-computation p-np-npc-nph normal

Answer**5.10.3 P Np Npc Nph: GATE2009-14** [top](#)<https://gateoverflow.in/1306>

Let π_A be a problem that belongs to the class NP. Then which one of the following is TRUE?

- A. There is no polynomial time algorithm for π_A .
- B. If π_A can be solved deterministically in polynomial time, then $P = NP$.
- C. If π_A is NP-hard, then it is NP-complete.
- D. π_A may be undecidable.

gate2009 theory-of-computation p-np-npc-nph

Answer**5.10.4 P Np Npc Nph: GATE2012-4** [top](#)<https://gateoverflow.in/36>

Assuming $P \neq NP$, which of the following is **TRUE**?

- A. $NP - \text{complete} = NP$
- B. $NP - \text{complete} \cap P = \emptyset$
- C. $NP - \text{hard} = NP$
- D. $P = NP - \text{complete}$

gate2012 theory-of-computation p-np-npc-nph

Answer**5.10.5 P Np Npc Nph: GATE2013-18** [top](#)<https://gateoverflow.in/1440>

Which of the following statements are **TRUE**?

1. The problem of determining whether there exists a cycle in an undirected graph is in P .
 2. The problem of determining whether there exists a cycle in an undirected graph is in NP .
 3. If a problem A is $NP - \text{Complete}$, there exists a non-deterministic polynomial time algorithm to solve A
- A. 1, 2 and 3
 - B. 1 and 2 only
 - C. 2 and 3 only

D. 1 and 3 only

gate2013 theory-of-computation p-np-npc-nph normal

Answer

Answers: P Np Npc Nph

5.10.1 P Np Npc Nph: GATE2005-58 top

<https://gateoverflow.in/1381>



Selected Answer

Independent Set- a set of vertices in a graph no two of which are adjacent.

Maximal Independent set problem - Given a graph G , find the size of the maximal independent set in it. This problem is NP-hard.

Independent set decision problem - Given a graph G and a number k , does G have an independent set of size k . This problem is NP-complete (NP-hard but in NP).

Now, in the given problem β corresponds to the Independent set decision problem. But there is a difference there. We have 5 instead of k . And this drastically changes the problem statement. We can now give a polynomial time **deterministic** algorithm for β .

- Find all vertex sets of size 5. We get $|V|C_5$ such vertex sets
- For each of them check if there is any adjacent vertices. This check can be done in constant time if we use an Adjacency matrix representation

Thus the whole time complexity reduces to $|V|C_5$ which is $O(|V|^5)$ and hence polynomial. ($|V|C_k$ is not polynomial but $|V|C_5$ is).

Problem α is asking for an independent set of size $|V| - 4$. This is equivalent to asking if G has a vertex cover* of size 4. Following a similar approach as done for β this problem also can be done in polynomial time.

So, both α and β are in P .

D choice.

Vertex cover of a graph G is the set of vertices such that each edge of the graph is incident on atleast one of those vertices.

Independent Set and Vertex cover Reduction: <https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/npcomplete.pdf>

10 votes

-- Arjun Suresh (350k points)

5.10.2 P Np Npc Nph: GATE2006-31 top

<https://gateoverflow.in/994>



Selected Answer

The only difference between **SHAM** and **DHAM**, in **SHAM** $|V|$ is divisible by 3 which can be checked in constant amount of time.

S,o the hardness of the two problem will the same. Next, finding hamiltonian cycle comes under **NPC** problem and **NPC** problem is a subset of **NPH**, so both are **NPH**.

So, option (A).

11 votes

-- Vicky Bajoria (5k points)

5.10.3 P Np Npc Nph: GATE2009-14 top

<https://gateoverflow.in/1306>



A problem which is in **P**, is also in **NP**- so, **A** is false. If problem can be solved deterministically in Polynomial time, then also we can't comment anything about $P=NP$, we just put this problem in **P**. So, **B** also false. **C** is TRUE because that is the definition of **NP**-complete.

D is false because all **NP** problems are not only decidable but decidable in polynomial time using a non-deterministic Turing machine.

12 votes

-- shreya ghosh (3.5k points)

5.10.4 P Np Npc Nph: GATE2012-4 top

<https://gateoverflow.in/36>



Answer is (**B**) $NP - \text{complete} \cap P = \emptyset$

Since, $P \neq NP$, there is at least one problem in NP , which is harder than all P problems. Lets take the hardest such problem, say X . Since, $P \neq NP$, $X \notin P$.

Now, by definition, $NP - \text{complete}$ problems are the hardest problems in NP and so X problem is in $NP - \text{complete}$. And being in NP , X can be reduced to all problems in $NP - \text{complete}$, making any other $NP - \text{complete}$ problem as *hard* as X . So, since $X \notin P$, none of the other $NP - \text{complete}$ problems also cannot be in P .

24 votes

-- Arjun Suresh (350k points)

5.10.5 P Np Npc Nph: GATE2013-18 top

<https://gateoverflow.in/1440>



Cycle detection in graph is in P as it can be done using a graph traversal ($O(V + E)$)

Ref: <http://www.geeksforgeeks.org/detect-cycle-undirected-graph/>

If a problem is in P then it is also in NP as P is a subset of NP . So, both **1** and **2** are TRUE.

Statement **3** is also true as $NP - \text{Complete}$ requires a problem to be in NP and for any problem in NP , we have a non-deterministic polynomial time algorithm.

So, answer is **A**) - 1, 2 and 3 are TRUE.

20 votes

-- Arjun Suresh (350k points)

5.11**Pumping Lemma(1)** top**5.11.1 Pumping Lemma: GATE2005-IT-40** top<https://gateoverflow.in/3787>

A language L satisfies the Pumping Lemma for regular languages, and also the Pumping Lemma for context-free languages. Which of the following statements about L is TRUE?

- A. L is necessarily a regular language.
- B. L is necessarily a context-free language, but not necessarily a regular language.
- C. L is necessarily a non-regular language.
- D. None of the above

[gate2005-it](#) [theory-of-computation](#) [pumping-lemma](#) [easy](#)
Answer**Answers: Pumping Lemma****5.11.1 Pumping Lemma: GATE2005-IT-40** top<https://gateoverflow.in/3787>

Selected Answer

Answer is (D). If a language is regular, it definitely satisfies pumping lemma. But converse need not be true. If a language satisfies pumping lemma then it may or may not be regular.

20 votes

-- Rajat Sharma (583 points)

5.12**Pushdown Automata(13)** top**5.12.1 Pushdown Automata: GATE1996-13** top<https://gateoverflow.in/2765>

Let $Q = (\{q_1, q_2\}, \{a, b\}, \{a, b, \perp\}, \delta, \perp, \phi)$ be a pushdown automaton accepting by empty stack for the language which is the set of all nonempty even palindromes over the set $\{a, b\}$. Below is an incomplete specification of the transitions δ . Complete the specification. The top of the stack is assumed to be at the right end of the string representing stack contents.

1. $\delta(q_1, a, \perp) = \{(q_1, \perp a)\}$
2. $\delta(q_1, b, \perp) = \{(q_1, \perp b)\}$
3. $\delta(q_1, a, a) = \{(q_1, aa)\}$
4. $\delta(q_1, b, a) = \{(q_1, ab)\}$
5. $\delta(q_1, a, b) = \{(q_1, ba)\}$
6. $\delta(q_1, b, b) = \{(q_1, bb)\}$
7. $\delta(q_1, a, a) = \{(\dots, \dots)\}$
8. $\delta(q_1, b, b) = \{(\dots, \dots)\}$
9. $\delta(q_2, a, a) = \{(q_2, \epsilon)\}$
10. $\delta(q_2, b, b) = \{(q_2, \epsilon)\}$
11. $\delta(q_2, \epsilon, \perp) = \{(q_2, \epsilon)\}$

[gate1996](#) [theory-of-computation](#) [pushdown-automata](#) [normal](#)
Answer

5.12.2 Pushdown Automata: GATE1997-6.6 [top](#)

<https://gateoverflow.in/2262>

Which of the following languages over $\{a, b, c\}$ is accepted by a deterministic pushdown automata?

- A. $\{wcw^R \mid w \in \{a, b\}^*\}$
- B. $\{ww^R \mid w \in \{a, b, c\}^*\}$
- C. $\{a^n b^n c^n \mid n \geq 0\}$
- D. $\{w \mid w \text{ is a palindrome over } \{a, b, c\}\}$

Note: w^R is the string obtained by reversing ' w '.

[gate1997](#) [theory-of-computation](#) [pushdown-automata](#) [easy](#)

Answer

5.12.3 Pushdown Automata: GATE1998-13 [top](#)

<https://gateoverflow.in/1727>

Let $M = (\{q_0, q_1\}, \{0, 1\}, \{z_0, X\}, \delta, q_0, z_0, \phi)$ be a Pushdown automation where δ is given by

$$\begin{aligned}\delta(q_0, 1, z_0) &= \{(q_0, xz_0)\} \\ \delta(q_0, \epsilon, z_0) &= \{(q_0, \epsilon)\} \\ \delta(q_0, 1, X) &= \{(q_0, XX)\} \\ \delta(q_1, 1, X) &= \{(q_1, \epsilon)\} \\ \delta(q_0, 0, X) &= \{(q_1, X)\} \\ \delta(q_0, 0, z_0) &= \{(q_0, z_0)\} z\end{aligned}$$

- a. What is the language accepted by this PDA by empty stack?
- b. Describe informally the working of the PDA

[gate1998](#) [theory-of-computation](#) [pushdown-automata](#) [descriptive](#)

Answer

5.12.4 Pushdown Automata: GATE1999-1.6 [top](#)

<https://gateoverflow.in/377>

Let L_1 be the set of all languages accepted by a PDA by final state and L_2 the set of all languages accepted by empty stack. Which of the following is true?

- A. $L_1 = L_2$
- B. $L_1 \supset L_2$
- C. $L_1 \subset L_2$
- D. None

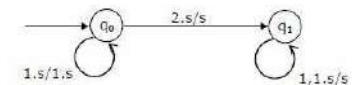
[normal](#) [theory-of-computation](#) [gate1999](#) [pushdown-automata](#)

Answer

5.12.5 Pushdown Automata: GATE2000-8 [top](#)

<https://gateoverflow.in/679>

A push down automation (pda) is given in the following extended notation of finite state diagram:



The nodes denote the states while the edges denote the moves of the pda. The edge labels are of the form $d, s/s'$ where d is the input symbol read and s, s' are the stack contents before and after the move. For example the edge labeled $1, s/1.s$ denotes the move from state q_0 to q_0 in which the input symbol 1 is read and pushed to the stack.

- A. Introduce two edges with appropriate labels in the above diagram so that the resulting pda accepts the language $\{x2x^R \mid x \in \{0,1\}^*\}$, x^R denotes reverse of x , by empty stack.
- B. Describe a non-deterministic pda with three states in the above notation that accept the language $\{0^n 1^m \mid n \leq m \leq 2n\}$ by empty stack

gate2000 theory-of-computation descriptive pushdown-automata

Answer

5.12.6 Pushdown Automata: GATE2001-6 [top](#)

<https://gateoverflow.in/747>

Give a deterministic PDA for the language $L = \{a^n cb^{2n} \mid n \geq 1\}$ over the alphabet $\Sigma = \{a, b, c\}$. Specify the acceptance state.

gate2001 theory-of-computation normal pushdown-automata

Answer

5.12.7 Pushdown Automata: GATE2004-IT-40 [top](#)

<https://gateoverflow.in/3683>

Let $M = (K, \Sigma, \Gamma, \Delta, s, F)$ be a pushdown automaton, where

$K = (s, f), F = \{f\}, \Sigma = \{a, b\}, \Gamma = \{a\}$ and
 $\Delta = \{((s, a, \epsilon), (s, a)), ((s, b, \epsilon), (s, a)), ((s, a, a), (f, \epsilon)), ((f, a, a), (f, \epsilon)), ((f, b, a), (f, \epsilon))\}$.

Which one of the following strings is not a member of $L(M)$?

- A. aaa
- B. $aabab$
- C. $baaba$
- D. bab

gate2004-it theory-of-computation pushdown-automata normal

Answer

5.12.8 Pushdown Automata: GATE2005-IT-38 [top](#)

<https://gateoverflow.in/3785>

Let P be a non-deterministic push-down automaton (NPDA) with exactly one state, q , and exactly one symbol, Z , in its stack alphabet. State q is both the starting as well as the accepting state of the PDA. The stack is initialized with one Z before the start of the operation of the PDA. Let the input alphabet of the PDA be Σ . Let $L(P)$ be the language accepted by the PDA by reading a string and reaching its accepting state. Let $N(P)$ be the language accepted by the PDA by reading a string and emptying its stack.

Which of the following statements is TRUE?

- A. $L(P)$ is necessarily Σ^* but $N(P)$ is not necessarily Σ^* .
- B. $N(P)$ is necessarily Σ^* but $L(P)$ is not necessarily Σ^* .
- C. Both $L(P)$ and $N(P)$ are necessarily Σ^* .

- D. Neither $L(P)$ nor $N(P)$ are necessarily Σ^*

gate2005-it theory-of-computation pushdown-automata normal

[Answer](#)

5.12.9 Pushdown Automata: GATE2006-IT-31 [top](#)

<https://gateoverflow.in/3570>

Which of the following languages is accepted by a non-deterministic pushdown automaton (PDA) but NOT by a deterministic PDA?

- A. $\{a^n b^n c^n \mid n \geq 0\}$
- B. $\{a^l b^m c^n \mid l \neq m \text{ or } m \neq n\}$
- C. $\{a^n b^n \mid n \geq 0\}$
- D. $\{a^m b^n \mid m, n \geq 0\}$

gate2006-it theory-of-computation pushdown-automata normal

[Answer](#)

5.12.10 Pushdown Automata: GATE2006-IT-33 [top](#)

<https://gateoverflow.in/3572>

Consider the pushdown automaton (PDA) below which runs over the input alphabet (a, b, c) . It has the stack alphabet $\{Z_0, X\}$ where Z_0 is the bottom-of-stack marker. The set of states of the PDA is (s, t, u, f) where s is the start state and f is the final state. The PDA accepts by final state. The transitions of the PDA given below are depicted in a standard manner. For example, the transition $(s, b, X) \rightarrow (t, XZ_0)$ means that if the PDA is in state s and the symbol on the top of the stack is X , then it can read b from the input and move to state t after popping the top of stack and pushing the symbols Z_0 and X (in that order) on the stack.

- $$\begin{aligned} (s, a, Z_0) &\rightarrow (s, XXZ_0) \\ (s, \epsilon, Z_0) &\rightarrow (f, \epsilon) \\ (s, a, X) &\rightarrow (s, XXX) \\ (s, b, X) &\rightarrow (t, \epsilon) \\ (t, b, X) &\rightarrow (t, .\epsilon) \\ (t, c, X) &\rightarrow (u, \epsilon) \\ (u, c, X) &\rightarrow (u, \epsilon) \\ (u, \epsilon, Z_0) &\rightarrow (f, \epsilon) \end{aligned}$$

The language accepted by the PDA is

- A. $\{a^l b^m c^n \mid l = m = n\}$
- B. $\{a^l b^m c^n \mid l = m\}$
- C. $\{a^l b^m c^n \mid 2l = m + n\}$
- D. $\{a^l b^m c^n \mid m = n\}$

gate2006-it theory-of-computation pushdown-automata normal

[Answer](#)

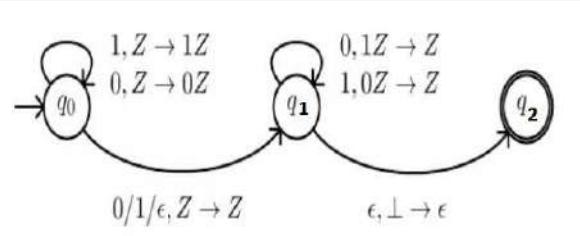
5.12.11 Pushdown Automata: GATE2015-1-51 [top](#)

<https://gateoverflow.in/8357>

Consider the NPDA

$$\langle Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}, \Gamma = \{0, 1, \perp\}, \delta, q_0, \perp, F = \{q_2\} \rangle$$

, where (as per usual convention) Q is the set of states, Σ is the input alphabet, Γ is the stack alphabet, δ is the state transition function q_0 is the initial state, \perp is the initial stack symbol, and F is the set of accepting states. The state transition is as follows:



Which one of the following sequences must follow the string 101100 so that the overall string is accepted by the automaton?

- A. 10110
- B. 10010
- C. 01010
- D. 01001

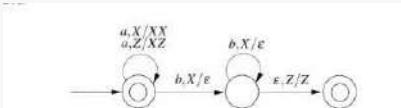
gate2015-1 theory-of-computation pushdown-automata normal

Answer

5.12.12 Pushdown Automata: GATE2016-1-43 top

<https://gateoverflow.in/39732>

Consider the transition diagram of a PDA given below with input alphabet $\Sigma = \{a, b\}$ and stack alphabet $\Gamma = \{X, Z\}$. Z is the initial stack symbol. Let L denote the language accepted by the PDA



Which one of the following is **TRUE**?

- A. $L = \{a^n b^n \mid n \geq 0\}$ and is not accepted by any finite automata
- B. $L = \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$ and is not accepted by any deterministic PDA
- C. L is not accepted by any Turing machine that halts on every input
- D. $L = \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$ and is deterministic context-free

gate2016-1 theory-of-computation pushdown-automata normal

Answer

5.12.13 Pushdown Automata: ISI 2015 PCB C4 A top

<https://gateoverflow.in/121031>

Design a context free grammar for the language consisting of all strings over $\{a, b\}$ that are not the form WW for any string W

pushdown-automata theory-of-computation isi2015

Answer

Answers: Pushdown Automata

5.12.1 Pushdown Automata: GATE1996-13 top

<https://gateoverflow.in/2765>



Selected Answer

$\delta(q_1, a, b) = \{(q_2, ba)\}$ means from state q_1 on input a with stack top being b , the PDA moves to state q_2 and pushes a on top of stack.

So, here the missing transitions are at the middle of the input string:

$$\begin{aligned}\delta(q_1, a, a) &= \{(q_2, \epsilon)\} \\ \delta(q_1, b, b) &= \{(q_2, \epsilon)\}\end{aligned}$$

Once middle is reached, now we should start popping. And so, we must go to state q_2 as well as pop the previous character on the stack. (The character before and after the middle must be same as the string is even length palindrome)

(This is a non-deterministic PDA)

15 votes

-- Arjun Suresh (350k points)

5.12.2 Pushdown Automata: GATE1997-6.6 top

<https://gateoverflow.in/2262>



Selected Answer

- A. $\{wcw^R \mid w \in \{a,b\}^*\}$ // Can be realized using DPDA cz we know the center of the string that is c here //Hence Option A is ANS
- B. $\{ww^R \mid w \in \{a,b,c\}^*\}$ // (set of even palindrom)NPDA bcz we can't find deterministically the center of palindrom string
cfg will $\rightarrow S \rightarrow aSa \mid bSb \mid cSc \mid \epsilon$
- C. $\{a^n b^n c^n \mid n \geq 0\}$ // CSL
- D. $\{w \mid w \text{ is a palindrome over } \{a,b,c\}\}$ // it is a cfl similar as option B
cfg will $\rightarrow S \rightarrow aSa \mid bSb \mid cSc \mid a \mid b \mid c \mid \epsilon$

15 votes

-- Rajesh Pradhan (22.6k points)

5.12.3 Pushdown Automata: GATE1998-13 top

<https://gateoverflow.in/1727>



Selected Answer

q_0 is start state

$$\delta(q_0, 0, Z_0) = (q_0, Z_0)$$

[Do Nothing operation, just read any no of 0's but do not keep in stack (any no of 0's because on reading 0's it remains on same state q_0)]

$$\delta(q_0, 1, Z_0) = (q_0, XZ_0) \quad [\text{Read first } 1 \text{ and keep one } X \text{ in stack}]$$

$$\delta(q_0, 1, X) = (q_0, XX) \quad [\text{Read any no of } 1's \text{ and keep one } X \text{ for each } 1 \text{ in stack}]$$

$$\delta(q_0, 0, X) = (q_1, X)$$

[Read single 0 and do nothing in stack, state changed from q_0 to q_1]

$$\delta(q_1, 1, X) = (q_1, \epsilon)$$

[Pop out one X from stack on reading each 1 on state q_1 (matching each 1 with the 1 read before single 0)]

$$\delta(q_0, \epsilon, Z_0) = (q_0, \epsilon)$$

[stack is empty , inputs are accepted here ,that is , ϵ or any of 0 's (we read earlier with Do Nothing operation)]

$$L = \{0^m, m \geq 0\}$$

No input accept after reaching on q_1 because stack will remain with Z_0 , stack initial symbol

Note : if we add one more transition

$$\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon) , \text{ then } L \text{ will be } \{0^m \cup 0^i 1^j 0 1^j, m, i, j \geq 0\}$$

Upvote 20 votes

-- Praveen Saini (54.6k points)

5.12.4 Pushdown Automata: GATE1999-1.6 top

<https://gateoverflow.in/377>



Selected Answer

Answer to the question is (A) $L_1 = L_2$.

Reason is for any PDA which accepts by final state there is an equivalent PDA (equivalent means that accepts the same language) which accepts by empty stack and vice-verse.

Now, this is not the case for DPDA's.

The set of languages accepted by a DPDA by empty stack is a strict subset of the set of languages accepted by a DPDA by final state.

It can also be said that set of languages accepted by a DPDA by empty stack is the set of languages accepted by a DPDA by final state and which has the prefix property.

A language has prefix property means if $w \in L$, then no proper prefix of $w \in L$.

From the above definition of prefix property it must be clear why DPDA by empty stack has this property. If any prefix of a word w (w in L) is in L means the stack should have been empty even before completely processing w . But, being a deterministic PDA, once the stack becomes empty, the DPDA accepts and halts. So, in no way can a DPDA accept w and its prefix.

PS: A DPDA with acceptance by empty stack cannot even accept all regular languages- example a^* .

Good read: http://www.cs.ucr.edu/~jiang/cs150/slides4week7_PDA+EquivToCFG.pdf

Upvote 49 votes

-- gatecse (1.8k points)

5.12.5 Pushdown Automata: GATE2000-8 top

<https://gateoverflow.in/679>

In a) $x2x^R$

Say for some word **0112110** we have to push every thing into the stack till **2** . then we get **1** then **1** will be at top of stack so pop it or if get **0** then **0** will at top of stack so pop it. For any word of language it is applicable. **2** is a mark that tell now we have to pop **0** for **0** and **1** for **1**.

So, on the edge q_0 to q_0 add $0, s/0.s$

and on edge q_1 to q_1 add $0, 0.s/s$

14 votes

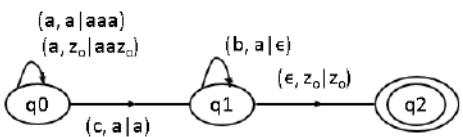
-- Praveen Saini (54.6k points)

5.12.6 Pushdown Automata: GATE2001-6 top

<https://gateoverflow.in/747>



Selected Answer



$$\begin{aligned} L &= \{a^n cb^2 n\} \\ &= \{acbb, aacbbbb, aaacbbbbbb, \dots\} \end{aligned}$$

Here, 'c' acts as center. Push 2 a's for each 'a' and after 'c' start popping an 'a' for each 'b'. If stack is empty and string is finished we move to q_2 which is the acceptance state.

13 votes

-- Saraswati Walijkar (419 points)

5.12.7 Pushdown Automata: GATE2004-IT-40 top

<https://gateoverflow.in/3683>



Selected Answer

Answer is D.

The language is like:

In start state a 's or b 's come, just push a 's on stack except for the last a which is used to shift from "start state" to "final state" (non-determinism) without consuming any stack symbol. Now in "final state", for equal no's of a 's and b 's just pop a 's from stack. This is the interpretation of transitions given for the language. So, it accepts strings of the form wab^n where $w \in (a+b)^*$ and $n < |w|$. This is violated only for option D and hence it is not in L .

- A. aaa : push apush apop a Final State, ACCEPTED
- B. $aabab$: push apush apush apop apop a Final State, ACCEPTED
- C. $baaba$: push apush apush apop apop a Final State, ACCEPTED
- D. bab : push apop adead state REJECTED

PS: This PDA is an NPDA and acceptance is by FINAL State. If no valid move is given for any state in NPDA, then the corresponding transition goes to a dead state.

18 votes

-- Rohan Mundhey (3.6k points)

5.12.8 Pushdown Automata: GATE2005-IT-38 top

<https://gateoverflow.in/3785>



Selected Answer

Answer is (D).

In **NPDA** we may have a dead configuration. This mean we may not give any transition to any alphabet from this state.

We say that a string is accepted if PDA is in final state after reading the final symbol in the string or after it has read '\$' symbol denoting end of the string and **it is in final state**.

Now coming to options:

Question never says that we have transitions defined for all the alphabet symbols in the PDA. Although it is ALREADY in the FINAL state we may not have ANY transition for any input symbol. In this case string will be rejected as it will never finish reading the string.

To sum up: A string is rejected in following two ways:

1. If no transition is defined for any configuration(this includes the final state as well because to accept the string we need the transition $(f, \$, _) \rightarrow (f, _)$ in final state or accepting state where blank ('_') denotes arbitrary stack symbol that does not matter because we are not accepting by EMPTY stack)
2. If string enters a configuration for which no transition is defined, STRING is rejected.

So, option (D) is correct. Because the same way it may not empty the stack when it finishes reading the string.

29 votes

-- Sandeep_Uniyal (7.5k points)

5.12.9 Pushdown Automata: GATE2006-IT-31 top

<https://gateoverflow.in/3570>



Selected Answer

Option **B** is correct.

$$L = \{a^l b^m c^n \mid l \neq m \text{ or } m \neq n\}$$

$$(q_0, a, Z_0) \rightarrow (q_0, aZ_0)$$

$$(q_0, a, a) \rightarrow (q_0, aa)$$

$$(q_0, b, a) \rightarrow (q_1, \epsilon), (q_2, ba)$$

[here it is NPDA where we have to check $l \neq m$ or $m \neq n$; for $l \neq m$ we need to pop a for b ; for $m \neq n$ we need to keep b in stack so that we can pop b for c]

$$(q_1, b, a) \rightarrow (q_1, \epsilon)$$

$$(q_1, c, a) \rightarrow (q_f, \epsilon)$$

$$(q_1, b, Z_0) \rightarrow (q_f, \epsilon)$$

$$(q_2, b, b) \rightarrow q_2, bb)$$

$$(q_2, c, b) \rightarrow (q_3, \epsilon)$$

$$(q_3, c, b) \rightarrow (q_3, \epsilon)$$

$$(q_3, c, a) \rightarrow (q_f, \epsilon)$$

$(q_3, \epsilon, b) \rightarrow (q_f, \epsilon)$

- (A) is wrong as it is not context free
- (D) a^*b^* is regular, so must have DFA, and so an equivalent DPDA
- (C) can be accepted using DPDA

1 21 votes

-- Praveen Saini (54.6k points)

5.12.10 Pushdown Automata: GATE2006-IT-33 top

<https://gateoverflow.in/3572>



Selected Answer

For every a we put two X in stack [at state s]

After that for every b we pop out one X [reach to state t (getting b after a)]

After that for every c we pop out one X [reach to state u (getting c after b)]

If all X are popped out then reached to final state f , mean for every b there is a , for every c there is a .

a was followed by b and b was followed by c [state s to t , t to u , u to f , final]

means sum of no of b 's and no of c 's = twice of no of a 's [one a for one b , one a for one c]

i.e. $\{a^l b^m c^n \mid 2l = m + n\}$

1 24 votes

-- Praveen Saini (54.6k points)

5.12.11 Pushdown Automata: GATE2015-1-51 top

<https://gateoverflow.in/8357>

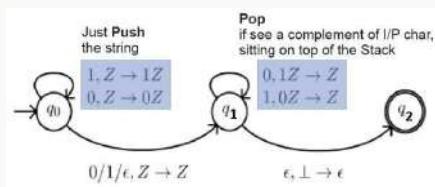


Selected Answer

Here, Z is used to represent the entire stack content except the top

Z is the string in Stack read from top to bottom. $1, Z \rightarrow 1Z$ means, on input symbol 1, the stack content changes from Z to $1Z$

In q_0 state, for '1', a '1' is pushed and for a '0' a '0' is pushed. In q_1 state, for a '0' a '1' is popped and for a '1' a '0' is popped. So, the given PDA is accepting all strings of the form $x0x'_r$ or $x1x'_r$ or xx'_r , where x'_r is the reverse of the 1's complement of x . i.e.:



The given string 101100 has 6 letters and we are given 5 letter strings. So, $x0$ is done, with $x = 10110$. So, $x'_r = (01001)_r = 10010$.

Answer is option B.

1 57 votes

-- Arjun Suresh (350k points)

5.12.12 Pushdown Automata: GATE2016-1-43 [top](#)

<https://gateoverflow.in/39732>



Strings accepted at I^{st} final state are a^n , $n \geq 0$ and strings accepted at II^{nd} final state are $a^n b^n$, $n \geq 0$ (actually $n \geq 1$ at this state, $n = 0$ already included at first state).

$$L = \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$$

Language is deterministic context-free accepted by DPDA (dpda is already given) and so by TM too, and not regular (as we need stack to implement it), and so cannot be accepted by FA

1 51 votes

-- Praveen Saini (54,6k points)

5.12.13 Pushdown Automata: ISI 2015 PCB C4 A [top](#)

<https://gateoverflow.in/121031>



Now, this is what we called a problem.

Here is the grammar,

$$S \rightarrow E \mid U \mid \epsilon$$

$$E \rightarrow AB \mid BA$$

$$A \rightarrow ZAZ \mid a$$

$$B \rightarrow ZBZ \mid b$$

$$U \rightarrow ZUZ \mid Z$$

$$Z \rightarrow a \mid b$$

For more Explore [this](#).

1 10 votes

-- Muktinath Vishwakarma (35,4k points)

5.13 Recursive And Recursively Enumerable Languages(13) [top](#)

5.13.1 Recursive And Recursively Enumerable Languages: GATE1990-3-vi [top](#)

<https://gateoverflow.in/84833>

Choose the correct alternatives (More than one may be correct).

Recursive languages are:

- A. A proper superset of context free languages.
- B. Always recognizable by pushdown automata.
- C. Also called type 0 languages.
- D. Recognizable by Turing machines.

[gate1990](#) [normal](#) [theory-of-computation](#) [turing-machine](#) [recursive-and-recursively-enumerable-languages](#)

Answer

5.13.2 Recursive And Recursively Enumerable Languages: GATE2003-13 top

<https://gateoverflow.in/904>

Nobody knows yet if $P = NP$. Consider the language L defined as follows.

$$L = \begin{cases} (0+1)^* & \text{if } P = NP \\ \emptyset & \text{otherwise} \end{cases}$$

Which of the following statements is true?

- A. L is recursive
- B. L is recursively enumerable but not recursive
- C. L is not recursively enumerable
- D. Whether L is recursively enumerable or not will be known after we find out if $P = NP$

gate2003 theory-of-computation normal recursive-and-recursively-enumerable-languages

Answer

5.13.3 Recursive And Recursively Enumerable Languages: GATE2003-15 top

<https://gateoverflow.in/120>

If the strings of a language L can be effectively enumerated in lexicographic (i.e., alphabetic) order, which of the following statements is true?

- A. L is necessarily finite
- B. L is regular but not necessarily finite
- C. L is context free but not necessarily regular
- D. L is recursive but not necessarily context-free

theory-of-computation gate2003 normal recursive-and-recursively-enumerable-languages

Answer

5.13.4 Recursive And Recursively Enumerable Languages: GATE2005-56 top

<https://gateoverflow.in/1379>

Let L_1 be a recursive language, and let L_2 be a recursively enumerable but not a recursive language. Which one of the following is TRUE?

- A. L_1' is recursive and L_2' is recursively enumerable
- B. L_1' is recursive and L_2' is not recursively enumerable
- C. L_1' and L_2' are recursively enumerable
- D. L_1' is recursively enumerable and L_2' is recursive

gate2005 theory-of-computation recursive-and-recursively-enumerable-languages easy

Answer

5.13.5 Recursive And Recursively Enumerable Languages: GATE2008-13, ISRO2016-36 top

<https://gateoverflow.in/411>

If L and \bar{L} are recursively enumerable then L is

- A. regular
- B. context-free
- C. context-sensitive
- D. recursive

[gate2008](#) [theory-of-computation](#) [easy](#) [isro2016](#) [recursive-and-recursively-enumerable-languages](#)

[Answer](#)

5.13.6 Recursive And Recursively Enumerable Languages: GATE2008-48 [top](#)

<https://gateoverflow.in/461>

Which of the following statements is false?

- A. Every NFA can be converted to an equivalent DFA
- B. Every non-deterministic Turing machine can be converted to an equivalent deterministic Turing machine
- C. Every regular language is also a context-free language
- D. Every subset of a recursively enumerable set is recursive

[gate2008](#) [theory-of-computation](#) [easy](#) [recursive-and-recursively-enumerable-languages](#)

[Answer](#)

5.13.7 Recursive And Recursively Enumerable Languages: GATE2010-17 [top](#)

<https://gateoverflow.in/2190>

Let L_1 be the recursive language. Let L_2 and L_3 be languages that are recursively enumerable but not recursive. Which of the following statements is not necessarily true?

- A. $L_2 - L_1$ is recursively enumerable.
- B. $L_1 - L_3$ is recursively enumerable.
- C. $L_2 \cap L_3$ is recursively enumerable.
- D. $L_2 \cup L_3$ is recursively enumerable.

[gate2010](#) [theory-of-computation](#) [recursive-and-recursively-enumerable-languages](#) [decidability](#) [normal](#)

[Answer](#)

5.13.8 Recursive And Recursively Enumerable Languages: GATE2014-1-35 [top](#)

<https://gateoverflow.in/1810>

Let L be a language and \bar{L} be its complement. Which one of the following is **NOT** a viable possibility?

- A. Neither L nor \bar{L} is recursively enumerable (r.e.).
- B. One of L and \bar{L} is r.e. but not recursive; the other is not r.e.
- C. Both L and \bar{L} are r.e. but not recursive.
- D. Both L and \bar{L} are recursive.

[gate2014-1](#) [theory-of-computation](#) [easy](#) [recursive-and-recursively-enumerable-languages](#)

Answer

5.13.9 Recursive And Recursively Enumerable Languages: GATE2014-2-16 [top](#)

<https://gateoverflow.in/1972>

Let $A \leq_m B$ denotes that language A is mapping reducible (also known as many-to-one reducible) to language B . Which one of the following is FALSE?

- A. If $A \leq_m B$ and B is recursive then A is recursive.
- B. If $A \leq_m B$ and A is undecidable then B is undecidable.
- C. If $A \leq_m B$ and B is recursively enumerable then A is recursively enumerable.
- D. If $A \leq_m B$ and B is not recursively enumerable then A is not recursively enumerable.

gate2014-2 theory-of-computation recursive-and-recursively-enumerable-languages normal

Answer

5.13.10 Recursive And Recursively Enumerable Languages: GATE2015-1-3 [top](#)

<https://gateoverflow.in/8019>

For any two languages L_1 and L_2 such that L_1 is context-free and L_2 is recursively enumerable but not recursive, which of the following is/are necessarily true?

- I. \bar{L}_1 (Complement of L_1) is recursive
 - II. \bar{L}_2 (Complement of L_2) is recursive
 - III. L_1 is context-free
 - IV. $L_1 \cup L_2$ is recursively enumerable
- A. I only
 - B. III only
 - C. III and IV only
 - D. I and IV only

gate2015-1 theory-of-computation recursive-and-recursively-enumerable-languages normal

Answer

5.13.11 Recursive And Recursively Enumerable Languages: GATE2016-1-44 [top](#)

<https://gateoverflow.in/39721>

Let X be a recursive language and Y be a recursively enumerable but not recursive language. Let W and Z be two languages such that \bar{Y} reduces to W , and Z reduces to \bar{X} (reduction means the standard many-one reduction). Which one of the following statements is TRUE?

- A. W can be recursively enumerable and Z is recursive.
- B. W can be recursive and Z is recursively enumerable.
- C. W is not recursively enumerable and Z is recursive.
- D. W is not recursively enumerable and Z is not recursive.

gate2016-1 theory-of-computation easy recursive-and-recursively-enumerable-languages

Answer

5.13.12 Recursive And Recursively Enumerable Languages: TIFR2010-B-40 [top](#)

<https://gateoverflow.in/19048>

Which of the following statement is FALSE?

- A. All recursive sets are recursively enumerable.
- B. The complement of every recursively enumerable sets is recursively enumerable.
- C. Every Non-empty recursively enumerable set is the range of some totally recursive function.
- D. All finite sets are recursive.
- E. The complement of every recursive set is recursive.

[tifr2010](#) | [theory-of-computation](#) | [recursive-and-recursively-enumerable-languages](#)

Answer

5.13.13 Recursive And Recursively Enumerable Languages: TIFR2012-B-19 [top](#)

<https://gateoverflow.in/25218>

Which of the following statements is TRUE?

- A. Every turning machine recognizable language is recursive.
- B. The complement of every recursively enumerable language is recursively enumerable.
- C. The complement of a recursive language is recursively enumerable.
- D. The complement of a context-free language is context-free.
- E. The set of turning machines which do not halt on empty input forms a recursively enumerable set.

[tifr2012](#) | [theory-of-computation](#) | [recursive-and-recursively-enumerable-languages](#)

Answer

Answers: Recursive And Recursively Enumerable Languages

5.13.1 Recursive And Recursively Enumerable Languages: GATE1990-3-vi [top](#)

<https://gateoverflow.in/84833>



Selected Answer

- A. A proper superset of context free languages. **TRUE** Since there are languages which are not CFL still Recursive
- B. Always recognizable by pushdown automata. **FALSE**
- C. Also called type 0 languages. **FALSE** R.E languages are actually type-0 languages.
- D. Recognizable by Turing machines **TRUE**

13 votes

-- Prajwal Bhat (11.3k points)

5.13.2 Recursive And Recursively Enumerable Languages: GATE2003-13 [top](#)

<https://gateoverflow.in/904>



Selected Answer

(A) L is recursive. If $P = NP$, L is Σ^* which is recursive (in fact regular). If not, $L = \emptyset$ which is again recursive. So, in both cases L is recursive.

26 votes

-- Arjun Suresh (350k points)

5.13.3 Recursive And Recursively Enumerable Languages: GATE2003-15 top

<https://gateoverflow.in/120>



Selected Answer

Answer is (D) L is recursive but not necessarily Regular or not even context-free.

Since, the strings of L can be enumerated it means L is recursively enumerable. That is we have a TM which accepts all strings in L . Now, to be recursive the TM should reject all strings not in L . Since, the strings of the language can be enumerated in lexicographic order, it's easy to do this. For any word w , if we see a word in the enumeration which is lexicographically higher than w but no w , it means w is not in the language. This makes L "recursive".

Now, why L need not be context free or regular? Consider

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

The strings of this language can be enumerated in lexicographic order. But we know L is not context free as no PDA can accept L .

👍 58 votes

-- Arjun Suresh (350k points)

5.13.4 Recursive And Recursively Enumerable Languages: GATE2005-56 top

<https://gateoverflow.in/1379>



Selected Answer

L_1 being recursive, we have a TM M for L_1 which accepts all words in L_1 and rejects all words in L_1' . So, this TM also works for L_1' by changing the accept and reject states. Thus L_1' is recursive.

L_2 being recursively enumerable but not recursive means TM for L_2 can accept all words in L_2 but cannot reject all words not in L_2 \Rightarrow TM for L_2' cannot exist (as otherwise TM for L_2 could simulate the moves of that TM to reject words in L_2') $\Rightarrow L_2'$ is not recursively enumerable. So, (B).

👍 22 votes

-- Arjun Suresh (350k points)

5.13.5 Recursive And Recursively Enumerable Languages: GATE2008-13, ISRO2016-36 top

<https://gateoverflow.in/411>



Selected Answer

(D) recursive

L is recursively enumerable means a TM accepts all strings in L . L is recursively enumerable means a TM accepts all strings in L . So, we can always decide if a string is in L or not, making L recursive.

<http://goo.gl/RtV8MO>

👍 26 votes

-- Keith Kr (6.1k points)

5.13.6 Recursive And Recursively Enumerable Languages: GATE2008-48 top

<https://gateoverflow.in/461>



Selected Answer

There exists a set of languages which is **RE** but not **REC** (i.e. Recursively Enumerable but not Recursive), this set is a subset of **RE** but is Not Recursive.

Option **D** tells us that every subset of **RE** is **REC** this is false.
Hence, **option D** is chosen.

18 votes

-- Amar Vashishth (30.5k points)

5.13.7 Recursive And Recursively Enumerable Languages: GATE2010-17 top

<https://gateoverflow.in/2190>



Selected Answer

Recursively enumerable languages are closed under union and intersection. So, lets consider each option

A. $L_2 - L_1 = L_2 \cap \overline{L_1}$

Recursive languages are closed under complement, and so $\overline{L_1}$ is recursive and hence recursively enumerable also. So, $L_2 \cap \overline{L_1}$ is recursively enumerable is always TRUE.

B. $L_1 - L_3 = L_1 \cap \overline{L_3}$

Recursively enumerable languages are not closed under complement. So, $\overline{L_3}$ may or may not be recursively enumerable and hence we can't say anything if $L_1 \cap \overline{L_3}$ is recursively enumerable or not.

C. Intersection of two recursively enumerable languages is always recursively enumerable(RE closed under intersection).

D. Union of two recursively enumerable languages is always recursively enumerable(RE closed under union).

For verifying closure properties:

http://gatecse.in/wiki/Closure_Property_of_Language_Families

50 votes

-- Arjun Suresh (350k points)

5.13.8 Recursive And Recursively Enumerable Languages: GATE2014-1-35 top

<https://gateoverflow.in/1810>



Selected Answer

(C) is not possible. If L is re we have a TM that accepts string in L . If L' is re, we have a TM that accepts strings in L' . So, using both these TMs we can make a new TM M which accepts strings in L and rejects strings in L' - that is M decides L , making L recursive.

21 votes

-- Arjun Suresh (350k points)

5.13.9 Recursive And Recursively Enumerable Languages: GATE2014-2-16 top

<https://gateoverflow.in/1972>



$A \leq_m B$ means A cannot be harder than B . (Since A can be reduced to B , instead of deciding A , we can now decide B)

So, the first 3 options are correct. Option (D) is false, as B is not recursively enumerable doesn't guarantee A is not recursively enumerable.

25 votes

-- Arjun Suresh (350k points)

5.13.10 Recursive And Recursively Enumerable Languages: GATE2015-1-3 top

<https://gateoverflow.in/8019>



Answer is D.

L_1 is context-free and hence recursive also. Recursive set being closed under complement, L_1' will be recursive.

L_1' being recursive it is also recursively enumerable and Recursively Enumerable set is closed under Union. So, $L_1' \cup L_2$ is recursively enumerable.

Context free languages are not closed under complement, so **III** is false

Recursive set is closed under complement. So, if L_2' is recursive, $(L_2')' = L_2$ is also recursive which is not the case here. So, **II** is also false.

24 votes

-- Arjun Suresh (350k points)

5.13.11 Recursive And Recursively Enumerable Languages: GATE2016-1-44 top

<https://gateoverflow.in/39721>



X is recursive language, so \overline{X} is also recursive.

Y is recursively enumerable, but not recursive so \overline{Y} is not recursively enumerable language.

$A \leq B$, (A is reducible to B), i. e, solving A cannot be "harder" than solving B .

1. If A is reducible to B , and B is decidable, then A is decidable.
 - i) if A is reducible to B , and B is recursive, then A is recursive.
2. If A is undecidable and reducible to B , then B is undecidable.
 - i) if B is recursively enumerable, and A is reducible to B , then A is recursively enumerable.
 - ii) if A is not recursively enumerable, and reducible to B , then B is not recursively enumerable.

Now Back to question.

\overline{Y} is not recursively enumerable, and reducible to W , then W is not recursively enumerable (using 2(ii)).

Z is reducible to \overline{X} and \overline{X} is recursive, then Z is recursive (using 1(i)).

Option C is correct.

48 votes

-- Praveen Saini (54.6k points)

5.13.12 Recursive And Recursively Enumerable Languages: TIFR2010-B-40 [top](#)

<https://gateoverflow.in/19048>



Selected Answer

(B) The complement of every recursively enumerable sets is recursively enumerable.

Because RE language are not closed under complement.

10 votes

-- Umang Raman (15.7k points)

5.13.13 Recursive And Recursively Enumerable Languages: TIFR2012-B-19 [top](#)

<https://gateoverflow.in/25218>



Selected Answer

- a. False. Turing recognizable language are recursive enumerable and recursive set is a **proper subset** of it.
- b. False, Complement of r.e. **need not** be r.e.
- c. **True**. Complement of recursive language is recursive and every recursive language is recursive enumerable.
- d. False. Complement of CFL **need not** be CFL (but is guaranteed to be a CSL).
- e. False. **NOT halt** on empty, yes ans not possible. Only no ans is there. So it will be non r.e. while its complement is r.e. but not recursive.

12 votes

-- srestha (87k points)

5.14

Regular Expressions(27) [top](#)

5.14.1 Regular Expressions: CMI2017-A-01 [top](#)

<https://gateoverflow.in/203254>

The regular expression $(a^* + b)^*$ is equivalent to which of the following regular expressions:

- a. a^*b^*
- b. $(a^*b + b)^*$
- c. $(a + b^*)^*$
- d. $(a^*b)^*$

[theory-of-computation](#) [regular-expressions](#) [cmi2017](#)

Answer

5.14.2 Regular Expressions: GATE1987-10d [top](#)

<https://gateoverflow.in/82455>

Give a regular expression over the alphabet $\{0, 1\}$ to denote the set of proper non-null substrings of the string 0110.

gate1987 theory-of-computation regular-expressions

Answer

5.14.3 Regular Expressions: GATE1991-03,xiii [top](#)

<https://gateoverflow.in/527>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only.

Let $r = 1(1 + 0)^*$, $s = 11^*0$ and $t = 1^*0$ be three regular expressions. Which one of the following is true?

- A. $L(s) \subseteq L(r)$ and $L(s) \subseteq L(t)$
- B. $L(r) \subseteq L(s)$ and $L(s) \subseteq L(t)$
- C. $L(s) \subseteq L(t)$ and $L(s) \subseteq L(r)$
- D. $L(t) \subseteq L(s)$ and $L(s) \subseteq L(r)$
- E. None of the above

gate1991 theory-of-computation regular-expressions normal

Answer

5.14.4 Regular Expressions: GATE1992-02,xvii [top](#)

<https://gateoverflow.in/575>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Which of the following regular expression identities is/are TRUE?

- A. $r^{(*)} = r^*$
- B. $(r^*s^*) = (r+s)^*$
- C. $(r+s)^* = r^* + s^*$
- D. $r^*s^* = r^* + s^*$

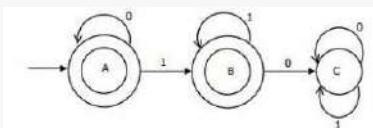
gate1992 theory-of-computation regular-expressions easy

Answer

5.14.5 Regular Expressions: GATE1994-2.10 [top](#)

<https://gateoverflow.in/2477>

The regular expression for the language recognized by the finite state automaton of figure is _____



gate1994 theory-of-computation finite-automata regular-expressions easy

Answer

5.14.6 Regular Expressions: GATE1995-1.9 , ISRO2017-13 [top](#)

<https://gateoverflow.in/2598>

In some programming language, an identifier is permitted to be a letter followed by any number of letters or digits. If L and D denote the sets of letters and digits respectively, which of the following expressions defines an identifier?

- A. $(L + D)^+$
- B. $(L \cdot D)^*$
- C. $L(L + D)^*$
- D. $L(L \cdot D)^*$

gate1995 theory-of-computation regular-expressions easy isro2017

[Answer](#)

5.14.7 Regular Expressions: GATE1996-1.8 [top](#)

<https://gateoverflow.in/2712>

Which two of the following four regular expressions are equivalent? (ϵ is the empty string).

- i. $(00)^*(\epsilon + 0)$
 - ii. $(00)^*$
 - iii. 0^*
 - iv. $0(00)^*$
-
- A. (i) and (ii)
 - B. (ii) and (iii)
 - C. (i) and (iii)
 - D. (iii) and (iv)

gate1996 theory-of-computation regular-expressions easy

[Answer](#)

5.14.8 Regular Expressions: GATE1997-6.4 [top](#)

<https://gateoverflow.in/2260>

Which one of the following regular expressions over $\{0,1\}$ denotes the set of all strings not containing 100 as substring?

- A. $0^*(1 + 0)^*$
- B. 0^*1010^*
- C. $0^*1^*01^*$
- D. $0^*(10 + 1)^*$

gate1997 theory-of-computation regular-expressions normal

[Answer](#)

5.14.9 Regular Expressions: GATE1998-1.12 [top](#)

<https://gateoverflow.in/1649>

The string 1101 does not belong to the set represented by

- A. $110^*(0 + 1)$
- B. $1(0 + 1)^*101$
- C. $(10)^*(01)^*(00 + 11)^*$
- D. $(00 + (11)^*0)^*$

gate1998 theory-of-computation regular-expressions easy

Answer**5.14.10 Regular Expressions: GATE1998-1.9** [top](#)<https://gateoverflow.in/1646>

If the regular set A is represented by $A = (01 + 1)^*$ and the regular set B is represented by $B = ((01)^*1^*)^*$, which of the following is true?

- A. $A \subset B$
- B. $B \subset A$
- C. A and B are incomparable
- D. $A = B$

[gate1998](#) [theory-of-computation](#) [regular-expressions](#) [normal](#)**Answer****5.14.11 Regular Expressions: GATE1998-3b** [top](#)<https://gateoverflow.in/2941>

Give a regular expression for the set of binary strings where every 0 is immediately followed by exactly k 1's and preceded by at least k 1's (k is a fixed integer)

[gate1998](#) [theory-of-computation](#) [regular-expressions](#) [easy](#)**Answer****5.14.12 Regular Expressions: GATE2000-1.4** [top](#)<https://gateoverflow.in/627>

Let S and T be languages over $\Sigma = \{a, b\}$ represented by the regular expressions $(a + b^*)^*$ and $(a + b)^*$, respectively. Which of the following is true?

- A. $S \subset T$
- B. $T \subset S$
- C. $S = T$
- D. $S \cap T = \emptyset$

[gate2000](#) [theory-of-computation](#) [regular-expressions](#) [easy](#)**Answer****5.14.13 Regular Expressions: GATE2003-14** [top](#)<https://gateoverflow.in/905>

The regular expression $0^*(10^*)^*$ denotes the same set as

- A. $(1^*0)^*1^*$
- B. $0 + (0 + 10)^*$
- C. $(0 + 1)^*10(0 + 1)^*$
- D. None of the above

[gate2003](#) [theory-of-computation](#) [regular-expressions](#) [easy](#)**Answer****5.14.14 Regular Expressions: GATE2004-IT-7** [top](#)<https://gateoverflow.in/3648>

Which one of the following regular expressions is NOT equivalent to the regular expression $(a + b + c)^*$?

- A. $(a^* + b^* + c^*)^*$

- B. $(a^*b^*c^*)^*$
 C. $((ab)^* + c^*)^*$
 D. $(a^*b^* + c^*)^*$

gate2004-it theory-of-computation regular-expressions normal

Answer

5.14.15 Regular Expressions: GATE2005-IT-5 [top](#)

<https://gateoverflow.in/3749>

Which of the following statements is TRUE about the regular expression $01^*0?$

- A. It represents a finite set of finite strings.
 B. It represents an infinite set of finite strings.
 C. It represents a finite set of infinite strings.
 D. It represents an infinite set of infinite strings.

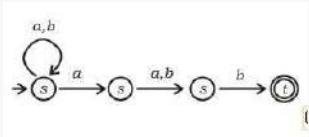
gate2005-it theory-of-computation regular-expressions easy

Answer

5.14.16 Regular Expressions: GATE2006-IT-5 [top](#)

<https://gateoverflow.in/3544>

Which regular expression best describes the language accepted by the non-deterministic automaton below?



- A. $(a+b)^* a(a+b)b$
 B. $\{abb\}^*$
 C. $(a+b)^* a(a+b)^* b(a+b)^*$
 D. $\{a+b\}^*$

gate2006-it theory-of-computation regular-expressions normal

Answer

5.14.17 Regular Expressions: GATE2007-IT-73 [top](#)

<https://gateoverflow.in/3525>

Consider the regular expression $R = (a+b)^* (aa+bb) (a+b)^*$

Which one of the regular expressions given below defines the same language as defined by the regular expression R ?

- A. $(a(ba)^* + b(ab)^*)(a+b)^+$
 B. $(a(ba)^* + b(ab)^*)^*(a+b)^*$
 C. $(a(ba)^*(a+bb) + b(ab)^*(b+aa))(a+b)^*$
 D. $(a(ba)^*(a+bb) + b(ab)^*(b+aa))(a+b)^+$

gate2007-it theory-of-computation regular-expressions normal

Answer

5.14.18 Regular Expressions: GATE2008-IT-5 [top](#)

<https://gateoverflow.in/3265>

Which of the following regular expressions describes the language over $\{0,1\}$ consisting of strings that contain exactly two 1's?

- A. $(0+1)^* 11(0+1)^*$
- B. $0^* 110^*$
- C. $0^* 10^* 10^*$
- D. $(0+1)^* 1(0+1)^* 1(0+1)^*$

gate2008-it theory-of-computation regular-expressions easy

[Answer](#)

5.14.19 Regular Expressions: GATE2009-15 [top](#)

<https://gateoverflow.in/1307>

Which one of the following languages over the alphabet $\{0,1\}$ is described by the regular expression: $(0+1)^* 0(0+1)^* 0(0+1)^*$?

- A. The set of all strings containing the substring 00
- B. The set of all strings containing at most two 0's
- C. The set of all strings containing at least two 0's
- D. The set of all strings that begin and end with either 0 or 1

gate2009 theory-of-computation regular-expressions easy

[Answer](#)

5.14.20 Regular Expressions: GATE2010-39 [top](#)

<https://gateoverflow.in/2340>

Let $L = \{w \in (0+1)^* \mid w \text{ has even number of } 1s\}$. i.e., L is the set of all the bit strings with even numbers of 1s. Which one of the regular expressions below represents L ?

- A. $(0^* 10^* 1)^*$
- B. $0^* (10^* 10^*)^*$
- C. $0^* (10^* 1)^* 0^*$
- D. $0^* 1 (10^* 1)^* 10^*$

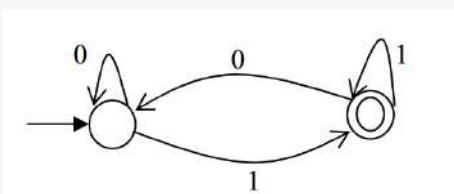
gate2010 theory-of-computation regular-expressions normal

[Answer](#)

5.14.21 Regular Expressions: GATE2014-1-36 [top](#)

<https://gateoverflow.in/1914>

Which of the regular expressions given below represent the following DFA?



- I. $0^* 1 (1 + 00^* 1)^*$
 - II. $0^* 1^* 1 + 11^* 0^* 1$
 - III. $(0+1)^* 1$
- A. I and II only
 - B. I and III only
 - C. II and III only

D. I, II and III

gate2014-1 theory-of-computation regular-expressions finite-automata easy

[Answer](#)

5.14.22 Regular Expressions: GATE2014-3-15 [top](#)

<https://gateoverflow.in/2049>

The length of the shortest string NOT in the language (over $\Sigma = \{a, b\}$) of the following regular expression is _____.

$$a^*b^*(ba)^*a^*$$

gate2014-3 theory-of-computation regular-expressions numerical-answers easy

[Answer](#)

5.14.23 Regular Expressions: GATE2016-1-18 [top](#)

<https://gateoverflow.in/39647>

Which one of the following regular expressions represents the language: *the set of all binary strings having two consecutive 0's and two consecutive 1's?*

- A. $(0+1)^*0011(0+1)^* + (0+1)^*1100(0+1)^*$
- B. $(0+1)^*(00(0+1)^*11 + 11(0+1)^*00)(0+1)^*$
- C. $(0+1)^*00(0+1)^* + (0+1)^*11(0+1)^*$
- D. $00(0+1)^*11 + 11(0+1)^*00$

gate2016-1 theory-of-computation regular-expressions normal

[Answer](#)

5.14.24 Regular Expressions: ISI2013-CS-4b [top](#)

<https://gateoverflow.in/47638>

Write a regular expression for all strings of 0's and 1's in which the total number of 0's to the right of each 1 is even. Justify your answer.

descriptive isi2013 theory-of-computation regular-expressions

[Answer](#)

5.14.25 Regular Expressions: TIFR2010-B-34 [top](#)

<https://gateoverflow.in/19047>

Let r, s, t be regular expressions. Which of the following identities is correct?

- A. $(r+s)^* = r^*s^*$
- B. $r(s+t) = rs + t$
- C. $(r+s)^* = r^* + s^*$
- D. $(rs+r)^*r = r(sr+r)^*$
- E. $(r^*s)^* = (rs)^*$

tifr2010 theory-of-computation regular-expressions

[Answer](#)

5.14.26 Regular Expressions: TIFR2015-B-7 [top](#)

<https://gateoverflow.in/29861>

Let a, b, c be regular expressions. Which of the following identities is correct?

- A. $(a+b)^* = a^*b^*$
- B. $a(b+c) = ab + c$
- C. $(a+b)^* = a^* + b^*$
- D. $(ab+a)^*a = a(ba+a)^*$
- E. None of the above.

tifr2015 theory-of-computation regular-expressions

Answer

5.14.27 Regular Expressions: TIFR2017-B-9 [top](#)

<https://gateoverflow.in/95705>

Which of the following regular expressions correctly accepts the set of all 0/1-strings with an even (possibly zero) number of 1s?

- A. $(10^*10^*)^*$
- B. $(0^*10^*1)^*$
- C. $0^*1(10^*1)^*10^*$
- D. $0^*1(0^*10^*10^*)^*10^*$
- E. $(0^*10^*1)^*0^*$

tifr2017 theory-of-computation regular-expressions

Answer

Answers: Regular Expressions

5.14.1 Regular Expressions: CMI2017-A-01 [top](#)

<https://gateoverflow.in/203254>



Selected Answer

(a* + b)* is equivalent to:

- $(a+b)^*$
- $(a+b^*)^*$
- $(a^*b^*)^*$
- basically any regular expression which can generate **all** strings in Σ^* .

Now looking at the options:

(a) a^*b^* - This can't generate *babab*

(b) $(a^*b+b)^*$ - This can't generate *aaa* (at least one *b* needs to be there in any non-empty string)

(c) $(a+b^*)^*$ - This one can generate all strings, i.e it is equivalent to $(a+b)^*$

(d) $(a^*b)^*$ - This can't generate *aaa* (at least one *b* needs to be there in any non-empty string)

So, correct option should be **(C) (a+b*)***.

4 votes

-- Rishabh Gupta (13.9k points)

5.14.2 Regular Expressions: GATE1987-10d top

<https://gateoverflow.in/82455>

0, 1, 01, 10, 11, 011, 110 :. RE : $(0 + 1 + 01 + 10 + 11 + 011 + 110)\epsilon$ and 0110 are not part of L

1 like 5 votes

-- Akshay (11.4k points)

5.14.3 Regular Expressions: GATE1991-03,xiii top

<https://gateoverflow.in/527>



Answer is A and C.

To know the answer let us check all the options:

- A. $L(s) \subseteq L(r)$: strings generated by s are any numbers of 1's followed by one 0, i.e., 10, 110, 1110, 1110, Strings generated by r are 1 followed by any combination of 0 or 1, i.e., 1, 10, 11, 1110, 101, 110, This shows that all the strings that can be generated by s , can also be generated by r it means $L(s) \subseteq L(r)$ is true.
 $L(s) \subseteq L(t)$: here strings generated by t are any numbers of 1 (here 1^* means we have strings as $\epsilon, 1, 11, 111, \dots$) followed by only one 0, i.e., 0, 10, 110, 1110, So we can see that all the strings that are present in s can also be generated by t , hence $L(s) \subseteq L(t)$ which shows that option A is true.
- B. $L(r) \subseteq L(s)$: this is false because string 1 which can be generated by r , cannot be generated by s .
- C. Same as option A.
- D. $L(t) \subseteq L(s)$: this is false because string 0 which can be generated by t , cannot be generated by s .

1 like 25 votes

-- neha pawar (4.2k points)

5.14.4 Regular Expressions: GATE1992-02,xvii top

<https://gateoverflow.in/575>



- a. is the answer
- b. RHS generates Σ^* while LHS can't generate strings where r comes after s like sr, srr etc. LHS \subset RHS.
- c. LHS generates Σ^* while RHS can't generate strings where r comes after an s . RHS \subset LHS.
- d. LHS contains all strings where after an s , no r comes. RHS contains all strings of either r or s but no combination of them. So, RHS \subset LHS.

1 like 27 votes

-- Arjun Suresh (350k points)

5.14.5 Regular Expressions: GATE1994-2.10 top

<https://gateoverflow.in/2477>



$$L = 0^* 1^*$$

L contains all binary strings where a 1 is not followed by a 0.

26 votes

-- Manu Thakur (39.6k points)

5.14.6 Regular Expressions: GATE1995-1.9 , ISRO2017-13 [top](#)



It is C

It has to be started by a letter followed by any number of letters (or) digits.

24 votes

-- Gate Keeda (19.6k points)

5.14.7 Regular Expressions: GATE1996-1.8 [top](#)

**Answer is C.**

You can have any no. of 0's as well as null.

A is false because you cannot have single 0 in ii). same for option B. In D you are forced to have single 0 in iv) whereas not in iii).

20 votes

-- Gate Keeda (19.6k points)

5.14.8 Regular Expressions: GATE1997-6.4 [top](#)



"A regular expression denoting a language (set of strings) means it should generate all string in L and not generate any string not in L"

- a. - generates 100
- b. doesn't generate 0 (start trying strings in lexicographic order- 0, 1, 00, 01, 10, ...)
- c. doesn't generate 1
- d. is the answer

31 votes

-- Arjun Suresh (350k points)

5.14.9 Regular Expressions: GATE1998-1.12 [top](#)



Only (a) and (b) can generate 1101.

In (c) after 11, we can not have 01 and so 1101 cannot be generated.

In (d) Every 11 followed by 0 and no single occurrence of 1 is possible. So it cannot generate 1101 or 11011.

29 votes

-- Arjun Suresh (350k points)

5.14.10 Regular Expressions: GATE1998-1.9 [top](#)

<https://gateoverflow.in/1646>



(d) $A = B$. Both generates all strings over $\{0, 1\}$ where a 0 is immediately followed by a 1.

19 votes

-- Arjun Suresh (350k points)

5.14.11 Regular Expressions: GATE1998-3b [top](#)

<https://gateoverflow.in/2941>



$$1^* 1^k (01^k)^* + 1^*$$

This is correct expression, this considering chance of not having any 0's (In that case string can also be empty string).

PS: The question assumes a fixed k , and we should expand the given expression for that k to get a valid regular expression.

20 votes

-- Akash Kanase (42.5k points)

5.14.12 Regular Expressions: GATE2000-1.4 [top](#)

<https://gateoverflow.in/627>



(c) $S = T$. Both generates all strings over Σ .

24 votes

-- Arjun Suresh (350k points)

5.14.13 Regular Expressions: GATE2003-14 [top](#)

<https://gateoverflow.in/905>



- A. Is the answer. Both (A) and the given expression generates all strings over Σ .
- B. doesn't generate 11
- C. doesn't generate 11

27 votes

-- Arjun Suresh (350k points)

5.14.14 Regular Expressions: GATE2004-IT-7 [top](#)

<https://gateoverflow.in/3648>



- A) $(a^* + b^* + c^*)^* = (\wedge + a + aa + \dots + b + bb + \dots + c + cc + \dots)^*$
 $= (a + b + c + aa + \dots + bb + \dots + cc + \dots)^*$
 $.)^* = (a + b + c)^*$
[any combination of rest of aa, bb, cc , etc already come in $(a + b + c)^*$]
- B) $(a^* b^* c^*)^* = (a^* + b^* + c^* + a^* b^* + b^* c^* + a^* c^* + \dots)^* = (a + b + c + \dots)^* = (a + b + c)^*$
- C) $((ab)^* + c^*)^* = (ab + c + \wedge + abab + \dots)^* = (ab + c)^*$
- D) $(a^* b^* + c^*)^* = (a^* + b^* + c^* + \dots)^* = (a + b + c + \dots)^* = (a + b + c)^*$

25 votes

-- Praveen Saini (54.6k points)

5.14.15 Regular Expressions: GATE2005-IT-5 top

<https://gateoverflow.in/3749>


Selected Answer

(B). Infinite set (because of $*$) of finite strings. A string is defined as a **FINITE sequence** of characters and hence can never be infinite.

29 votes

-- Arjun Suresh (350k points)

5.14.16 Regular Expressions: GATE2006-IT-5 top

<https://gateoverflow.in/3544>


Selected Answer

Well, **A** is answer.

Say s_1, s_2, s_3 and t are states (in sequence)

s_1 is start state

$s_1 = \wedge + s_1a + s_1b = \wedge + s_1(a + b) = (a + b)^*$ [using arden's theorem $R = Q + RP$, then $R = QP^*$] [\wedge bcoz of start state]

$s_2 = s_1a = (a + b)^*a$

$s_3 = s_2a + s_2b = s_2(a + b) = (a + b)^*a(a + b)$

$t = s_3b = (a + b)^*a(a + b)b$

t is final state so regular expression is $(a + b)^*a(a + b)b$

19 votes

-- Praveen Saini (54.6k points)

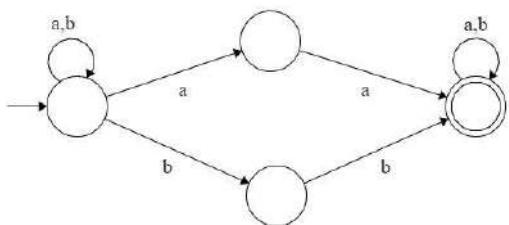
5.14.17 Regular Expressions: GATE2007-IT-73 top

<https://gateoverflow.in/3525>

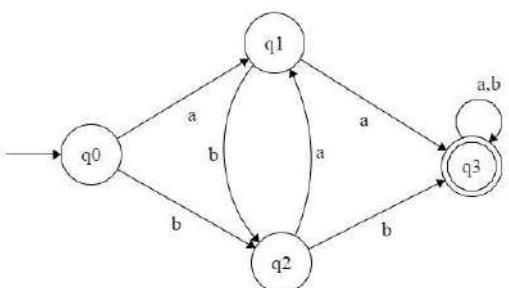
Selected Answer

$$R = (a+b)^*(aa+bb)(a+b)^*$$

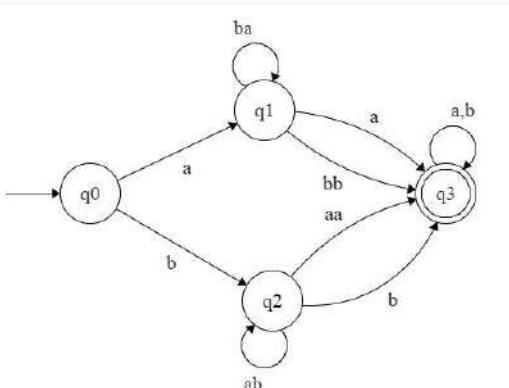
Having NFA:



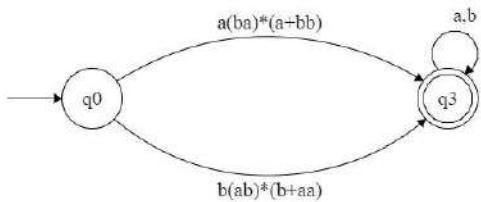
Equivalent DFA:



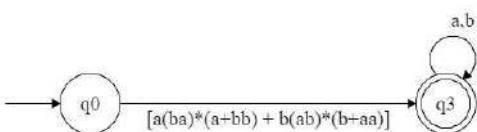
Which is equivalent Transition graph [by removing transition from $q1$ to $q2$ and $q2$ to $q1$ but does not effect on language, be careful]



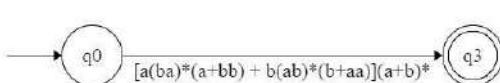
That is equivalent to:



Which is equivalent to:



Which is equivalent to:



So, equivalent regular expression is $[a(ba)^*(a+bb) + b(ab)^*(b+aa)](a+b)^*$

So, option **C** is answer.

29 votes

-- Praveen Saini (54.6k points)

Another quick approach of solving this question for keen observers :-

Observe that aa or bb is minimal string that is possible in first Regular Expression $(a+b)^*(aa+bb)(a+b)^*$.

- (A) We can have ba or ab as minimal strings which is not possible in $(a+b)^*(aa+bb)(a+b)^*$
- (B) We can have empty string, which is not possible in $(a+b)^*(aa+bb)(a+b)^*$.
- (D) Minimum string length is 3, aa or bb is not possible in this RE.

This rules out options A, B and D. So, option C must be the answer.

23 votes

-- Akash Kanase (42.5k points)

5.14.18 Regular Expressions: GATE2008-IT-5 top

<https://gateoverflow.in/3265>



Selected Answer

- A. with at least 2 consecutive 1's, any no of 0's and any no of 1's
- B. exactly two consecutive 1's
- C. exactly two 1's but need not be consecutive
- D. any no of 1's and 0's with at least two 1's

Hence, (C) is the correct option.

27 votes

-- Manu Thakur (39.6k points)

5.14.19 Regular Expressions: GATE2009-15 [top](#)

<https://gateoverflow.in/1307>



(C) is the answer.

Counter example for other choices:

- A. 1010 is accepted which doesn't contain 00
- B. 000 is accepted
- C. is the answer.
- D. 01 is not accepted

21 votes

-- Arjun Suresh (350k points)

5.14.20 Regular Expressions: GATE2010-39 [top](#)

<https://gateoverflow.in/2340>



- A. - If the string contains a 1, it must end in a 1 hence cannot generate all bit strings with even number of 1's (eg, 1010)
- B. - is the answer
- C. - between the second and third 1's a 0 is not allowed (eg, 011011)
- D. - 00 is not allowed, zero is an even number.

41 votes

-- Arjun Suresh (350k points)

5.14.21 Regular Expressions: GATE2014-1-36 [top](#)

<https://gateoverflow.in/1914>



(B) is the answer. (II) doesn't generate 11011 which is accepted by the given DFA.

38 votes

-- Arjun Suresh (350k points)

5.14.22 Regular Expressions: GATE2014-3-15 [top](#)

<https://gateoverflow.in/2049>



$$R = a^* b^* (ba)^* a^*$$

for finding shortest string that is not in language it is better to look strings of length 0, then of length 1 and so on

$\text{length}_0\{\epsilon\}$ is in L

$\text{length}_1\{a, b\}$ all belong to L

$\text{length}_2\{aa, ab, ba, bb\}$ all belong to L

length3{aaa, aab, aba, abb, baa, bab, bba, bbb} **bab does not belong to L**

👍 53 votes

-- Praveen Saini (54.6k points)

5.14.23 Regular Expressions: GATE2016-1-18 top

<https://gateoverflow.in/39647>



Selected Answer

Set of all binary strings having two consecutive 0s and two consecutive 1s

Anything 00 Anything 11 Anything + Anything 11 Anything 00 Anything

$(0+1)^*00(0+1)^*11(0+1)^* + (0+1)^*11(0+1)^*00(0+1)^*$

And it is same after taking common.

$(0+1)^*(00(0+1)^*11 + 11(0+1)^*00)(0+1)^*$

So, **option B** is answer.

Neither they said Both are immediate nor they give a predefined order, so it should be as above

👍 57 votes

-- Praveen Saini (54.6k points)

5.14.24 Regular Expressions: ISI2013-CS-4b top

<https://gateoverflow.in/47638>



Selected Answer

Here Total no. of 0's to the right of each 1's should be even. So, to the left side of every 1's no. of 0's can be anything.

$L = \{\epsilon, 0, 00, 000, 01, 011, 0111, 1, 11, 111, 1111, 100, 10000, 100100, 10010000, \dots\}$

Hence,

R.E. = $0^*(1+00)^* = 0^*(1(00)^*)^* = 0^*(1+(00)^*)^* = 0^*(1^*+(00))^* = 0^*(1^*+(00))$

👍 8 votes

-- Leen Sharma (40k points)

5.14.25 Regular Expressions: TIFR2010-B-34 top

<https://gateoverflow.in/19047>



Selected Answer

- | | |
|----------------------------|-------------------------------------|
| a. $(r+s)^* = r^*s^*$ | LHS can generate 'sr' but RHS not |
| b. $r(s+t) = rs + t$ | LHS can generate 'rt' but RHS not |
| c. $(r+s)^* = r^* + s^*$ | LHS can generate 'sr' but RHS not |
| d. $(rs+r)^*r = r(sr+r)^*$ | They are equivalent |
| e. $(r^*s)^* = (rs)^*$ | LHS can generate 'rrrs' but RHS not |

So option **D** is correct answer.

👍 13 votes

-- Umang Raman (15.7k points)

5.14.26 Regular Expressions: TIFR2015-B-7 [top](#)

<https://gateoverflow.in/29861>



Selected Answer

- a. $(a+b)^* = \{ \text{ any strings of over } \{a,b\} \}$
 $a^*b^* = \{ \text{ any number of } a's \text{ followed by any number of } b's \}$
False, as strings, ba, aba, bab , etc are not present in a^*b^*
- b. $a(b+c) = \{ab, ac\}$
 $ab + c = \{ab, c\}$
False
- c. $(a+b)^* = \{ \text{ any strings of over } \{a,b\} \}$
 $a^* + b^* = \{ \text{ any numbers of } a's \text{ or any numbers of } b's \}$
False, as strings, ab, ba, aba, bab etc are not present in $a^* + b^*$
- d. $(ab+a)^*a = a(ba+a)^*$, **True**
 $p(qp)^* = p\{\epsilon, qp, qpqp, qpqpqp, \dots\} = \{p, pqp, pqqpqp, pqqpqpqp, \dots\} = \{\epsilon, pq, pqpq, pqqpqpq, \dots\}$
 $(ab+a)^*a = (a(b+\epsilon))^*a = a((b+\epsilon)a)^* = a(ba+a)^*$

24 votes

-- Praveen Saini (54.6k points)

5.14.27 Regular Expressions: TIFR2017-B-9 [top](#)

<https://gateoverflow.in/95705>



Selected Answer

As, mentioned in the question, the regular expression must accept all strings of 0 and 1 but with even no of 1s (including no 1s). Hence, 00 must be in the language. Option A, B, C, D do not accept 00. Hence, option E is correct.

11 votes

-- tarun_svbk (1.5k points)

5.15

Regular Grammar(3) [top](#)

5.15.1 Regular Grammar: GATE1990-15a [top](#)

<https://gateoverflow.in/86864>

Is the language generated by the grammar G regular? If so, give a regular expression for it, else prove otherwise

$$G: \quad S \rightarrow aB$$

$$B \rightarrow bC$$

$$C \rightarrow xB$$

$$C \rightarrow c$$

[gate1990](#) [descriptive](#) [theory-of-computation](#) [regular-languages](#) [regular-grammar](#) [grammar](#)

Answer

5.15.2 Regular Grammar: GATE2006-IT-29 [top](#)

<https://gateoverflow.in/3568>

Consider the regular grammar below

$$\begin{aligned} S &\rightarrow bS \mid aA \mid \epsilon \\ A &\rightarrow aS \mid bA \end{aligned}$$

The Myhill-Nerode equivalence classes for the language generated by the grammar are

- A. $\{w \in (a+b)^* \mid \#a(w) \text{ is even}\}$ and $\{w \in (a+b)^* \mid \#a(w) \text{ is odd}\}$
- B. $\{w \in (a+b)^* \mid \#a(w) \text{ is even}\}$ and $\{w \in (a+b)^* \mid \#b(w) \text{ is odd}\}$
- C. $\{w \in (a+b)^* \mid \#a(w) = \#b(w)\}$ and $\{w \in (a+b)^* \mid \#a(w) \neq \#b(w)\}$
- D. $\{\epsilon\}, \{wa \mid w \in (a+b)^*\}$ and $\{wb \mid w \in (a+b)^*\}$

gate2006-it theory-of-computation normal regular-grammar

[Answer](#)

5.15.3 Regular Grammar: GATE2015-2-35 [top](#)

<https://gateoverflow.in/8159>

Consider the alphabet $\Sigma = \{0,1\}$, the null/empty string λ and the set of strings X_0, X_1 , and X_2 generated by the corresponding non-terminals of a regular grammar. X_0, X_1 , and X_2 are related as follows.

- $X_0 = 1X_1$
- $X_1 = 0X_1 + 1X_2$
- $X_2 = 0X_1 + \{\lambda\}$

Which one of the following choices precisely represents the strings in X_0 ?

- A. $10(0^* + (10)^*)1$
- B. $10(0^* + (10)^*)^*1$
- C. $1(0+10)^*1$
- D. $10(0+10)^*1 + 110(0+10)^*1$

gate2015-2 theory-of-computation regular-grammar normal

[Answer](#)

Answers: Regular Grammar

5.15.1 Regular Grammar: GATE1990-15a [top](#)

<https://gateoverflow.in/86864>



First of all this grammar is right linear. And we know:

Two special types of linear grammars are the following:

- the **left-linear** or [left regular grammars](#), in which **all nonterminals** in right hand sides are **at the left ends**;
- the **right-linear** or right regular grammars, in which **all nonterminals** in right hand sides are **at the right ends**.

Hence the given grammar is regular and hence the language generated by regular grammar will also be regular. Alternatively we can also write a regular expression for it. Let us see how to do it:

Given grammar:

$$G: S \rightarrow aB$$

$$B \rightarrow bC$$

$$C \rightarrow xB \mid c$$

So, we can reverse substitution from C onwards to S to see what S generates..

Substituting the yield of C in B , we have:

$$B \rightarrow bx B \mid bc$$

$$\text{which gives } B = (bx)^*bc$$

Now, substituting B in S we have:

$$S \rightarrow aB$$

$$S = a(bx)^*bc$$

Hence, the corresponding regular expression is : $a(bx)^*bc$

1 21 votes

-- HABIB MOHAMMAD KHAN (98.7k points)

5.15.2 Regular Grammar: GATE2006-IT-29 [top](#)

<https://gateoverflow.in/3568>

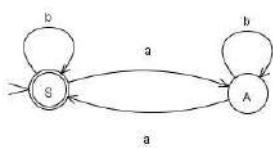


Selected Answer

Option A is correct.

The given grammar generates all string over the alphabet $\{a, b\}$ which have an even number of a 's.

The given right-linear grammar can be converted to the following **DFA**.



1 17 votes

-- Pragy Agarwal (20.6k points)

5.15.3 Regular Grammar: GATE2015-2-35 [top](#)

<https://gateoverflow.in/8159>



Selected Answer

Here we have little different version of Arden's Theorem.

If we have $R = PR + Q$ then it has a solution $R = P^*Q$

Proof :

$$R = PR + Q$$

$$= P(PR + Q) + Q \quad (\text{by putting } R = PR + Q)$$

$$= PPR + PQ + Q$$

$$= PP(PR + Q) + PQ + Q \quad (\text{by putting } R = PR + Q)$$

$$= PPPR + PPQ + PQ + Q$$

$$\text{and so on, we get } R = \{\dots + PPPPQ + PPPQ + PPQ + PQ + Q\} \\ = \{\dots + PPPP + PPP + PP + P + \lambda\}Q = P^*Q$$

or another way:

$$R = PR + Q \\ = P(P^*Q) + Q \quad (\text{by putting } R = P^*Q) \\ = (PP^* + \lambda)Q = P^*Q$$

So, equation is proved.

Now for the Above Question:

$$X_1 = 0X_1 + 1 X_2 \quad (\text{Equation 2}) \\ = 0X_1 + 1(0X_1 + \lambda) \quad (\text{Put the value of } X_2 \text{ from Equation 3}) \\ = 0X_1 + 10 X_1 + 1 = (0 + 10)X_1 + 1$$

$$X_1 = (0 + 10)^*1 \quad (\text{Apply if } R = PR + Q \text{ then } R = P^*Q)$$

$$X_0 = 1 X_1 \quad (\text{Equation 1})$$

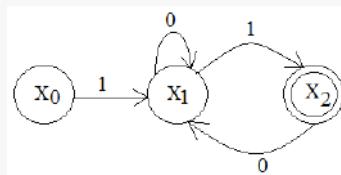
$$X_0 = 1(0 + 10)^*1 \quad (\text{Put the value of } X_1 \text{ we calculated}).$$

$$\text{So, } 1(0 + 10)^*1.$$

Option **C** is correct.

👍 34 votes

-- Praveen Saini (54.6k points)



Convert the given transitions to a state diagram

From the given diagram we can write

$$X_0 = 1(0 + 10)^*1 \text{ option C}$$

👍 45 votes

-- Umang Raman (15.7k points)

5.16

Regular Languages(37) top

5.16.1 Regular Languages: CMT2012-A-01 top

<https://gateoverflow.in/46529>

Let $L \subseteq \{0,1\}^*$. Which of the following is true?

- A. If L is regular, all subsets of L are regular.

- B. If all proper subsets of L are regular, then L is regular.
 C. If all finite subsets of L are regular, then L is regular.
 D. If a proper subset of L is not regular, then L is not regular.

[cmi2012](#) [theory-of-computation](#) [regular-languages](#)

[Answer](#)

5.16.2 Regular Languages: CMI2015-A-09 [top](#)

<https://gateoverflow.in/47046>

Let L_1 and L_2 be languages over an alphabet Σ such that $L_1 \subseteq L_2$. Which of the following is true:

- A. If L_2 is regular, then L_1 must also be regular.
 B. If L_1 is regular, then L_2 must also be regular.
 C. Either both L_1 and L_2 are regular, or both are not regular.
 D. None of the above.

[cmi2015](#) [theory-of-computation](#) [regular-languages](#)

[Answer](#)

5.16.3 Regular Languages: GATE1987-2h [top](#)

<https://gateoverflow.in/80589>

State whether the following statements are TRUE or FALSE:

Regularity is preserved under the operation of string reversal.

[gate1987](#) [regular-languages](#) [theory-of-computation](#)

[Answer](#)

5.16.4 Regular Languages: GATE1987-2i [top](#)

<https://gateoverflow.in/80590>

State whether the following statements are TRUE or FALSE:

All subsets of regular sets are regular.

[gate1987](#) [theory-of-computation](#) [regular-languages](#)

[Answer](#)

5.16.5 Regular Languages: GATE1990-3-viii [top](#)

<https://gateoverflow.in/84837>

Choose the correct alternatives (More than one may be correct).

Let R_1 and R_2 be regular sets defined over the alphabet Σ . Then:

- A. $R_1 \cap R_2$ is not regular.
 B. $R_1 \cup R_2$ is regular.
 C. $\Sigma^* - R_1$ is regular.
 D. R_1^* is not regular.

[gate1990](#) [normal](#) [theory-of-computation](#) [regular-languages](#)

[Answer](#)

5.16.6 Regular Languages: GATE1991-03,xiv [top](#)

<https://gateoverflow.in/528>

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Which of the following is the strongest correct statement about a finite language over some finite alphabet Σ ?

- A. It could be undecidable
- B. It is Turing-machine recognizable
- C. It is a context sensitive language.
- D. It is a regular language.
- E. None of the above,

gate1991 theory-of-computation easy regular-languages

[Answer](#)

5.16.7 Regular Languages: GATE1995-2.24 [top](#)

<https://gateoverflow.in/2637>

Let $\Sigma = \{0, 1\}$, $L = \Sigma^*$ and $R = \{0^n 1^n \mid n > 0\}$ then the languages $L \cup R$ and R are respectively

- A. regular, regular
- B. not regular, regular
- C. regular, not regular
- D. not regular, not regular

gate1995 theory-of-computation easy regular-languages

[Answer](#)

5.16.8 Regular Languages: GATE1996-1.10 [top](#)

<https://gateoverflow.in/2714>

Let $L \subseteq \Sigma^*$ where $\Sigma = \{a, b\}$. Which of the following is true?

- a. $L = \{x \mid x \text{ has an equal number of } a\text{'s and } b\text{'s}\}$ is regular
- b. $L = \{a^n b^n \mid n \geq 1\}$ is regular
- c. $L = \{x \mid x \text{ has more number of } a\text{'s than } b\text{'s}\}$ is regular
- d. $L = \{a^m b^n \mid m \geq 1, n \geq 1\}$ is regular

gate1996 theory-of-computation normal regular-languages

[Answer](#)

5.16.9 Regular Languages: GATE1998-2.6 [top](#)

<https://gateoverflow.in/1678>

Which of the following statements is false?

- a. Every finite subset of a non-regular set is regular
- b. Every subset of a regular set is regular
- c. Every finite subset of a regular set is regular
- d. The intersection of two regular sets is regular

gate1998 theory-of-computation easy regular-languages

[Answer](#)

5.16.10 Regular Languages: GATE1999-6 [top](#)

<https://gateoverflow.in/1505>

- A. Given that A is regular and $(A \cup B)$ is regular, does it follow that B is necessarily regular? Justify your answer.

- B. Given two finite automata $M1, M2$, outline an algorithm to decide if $L(M1) \subset L(M2)$. (note: strict subset)

gate1999 theory-of-computation normal regular-languages

Answer

5.16.11 Regular Languages: GATE2000-2.8 [top](#)

<https://gateoverflow.in/655>

What can be said about a regular language L over $\{a\}$ whose minimal finite state automaton has two states?

- A. L must be $\{a^n \mid n \text{ is odd}\}$
- B. L must be $\{a^n \mid n \text{ is even}\}$
- C. L must be $\{a^n \mid n \geq 0\}$
- D. Either L must be $\{a^n \mid n \text{ is odd}\}$, or L must be $\{a^n \mid n \text{ is even}\}$

gate2000 theory-of-computation easy regular-languages

Answer

5.16.12 Regular Languages: GATE2001-1.4 [top](#)

<https://gateoverflow.in/697>

Consider the following two statements:

$S_1 : \{0^{2n} \mid n \geq 1\}$ is a regular language

$S_2 : \{0^m 1^n 0^{m+n} \mid m \geq 1 \text{ and } n \geq 1\}$ is a regular language

Which of the following statement is correct?

- A. Only S_1 is correct
- B. Only S_2 is correct
- C. Both S_1 and S_2 are correct
- D. None of S_1 and S_2 is correct

gate2001 theory-of-computation easy regular-languages

Answer

5.16.13 Regular Languages: GATE2001-2.6 [top](#)

<https://gateoverflow.in/724>

Consider the following languages:

- $L1 = \{ww \mid w \in \{a,b\}^*\}$
- $L2 = \{ww^R \mid w \in \{a,b\}^*, w^R \text{ is the reverse of } w\}$
- $L3 = \{0^{2i} \mid i \text{ is an integer}\}$
- $L4 = \{0^{i^2} \mid i \text{ is an integer}\}$

Which of the languages are regular?

- A. Only $L1$ and $L2$
- B. Only $L2, L3$ and $L4$
- C. Only $L3$ and $L4$
- D. Only $L3$

[gate2001](#) [theory-of-computation](#) [normal](#) [regular-languages](#)
Answer

5.16.14 Regular Languages: GATE2006-29 [top](#)

<https://gateoverflow.in/992>

If s is a string over $(0+1)^*$ then let $n_0(s)$ denote the number of 0's in s and $n_1(s)$ the number of 1's in s . Which one of the following languages is not regular?

- A. $L = \{s \in (0+1)^* \mid n_0(s) \text{ is a 3-digit prime}\}$
- B. $L = \{s \in (0+1)^* \mid \text{for every prefix } s' \text{ of } s, |n_0(s') - n_1(s')| \leq 2\}$
- C. $L = \{s \in (0+1)^* \mid |n_0(s) - n_1(s)| \leq 4\}$
- D. $L = \{s \in (0+1)^* \mid n_0(s) \bmod 7 = n_1(s) \bmod 5 = 0\}$

[gate2006](#) [theory-of-computation](#) [normal](#) [regular-languages](#)
Answer

5.16.15 Regular Languages: GATE2006-IT-30 [top](#)

<https://gateoverflow.in/3569>

Which of the following statements about regular languages is NOT true ?

- A. Every language has a regular superset
- B. Every language has a regular subset
- C. Every subset of a regular language is regular
- D. Every subset of a finite language is regular

[gate2006-it](#) [theory-of-computation](#) [easy](#) [regular-languages](#)
Answer

5.16.16 Regular Languages: GATE2006-IT-80 [top](#)

<https://gateoverflow.in/3624>

Let L be a regular language. Consider the constructions on L below:

- I. repeat $(L) = \{ww \mid w \in L\}$
- II. prefix $(L) = \{u \mid \exists v : uv \in L\}$
- III. suffix $(L) = \{v \mid \exists u : uv \in L\}$
- IV. half $(L) = \{u \mid \exists v : |v| = |u| \text{ and } uv \in L\}$

Which of the constructions could lead to a non-regular language?

- A. Both I and IV
- B. Only I
- C. Only IV
- D. Both II and III

[gate2006-it](#) [theory-of-computation](#) [normal](#) [regular-languages](#)
Answer

5.16.17 Regular Languages: GATE2006-IT-81 [top](#)

<https://gateoverflow.in/3637>

Let L be a regular language. Consider the constructions on L below:

- I. repeat(L) = $\{ww \mid w \in L\}$
- II. prefix(L) = $\{u \mid \exists v : uv \in L\}$
- III. suffix(L) = $\{v \mid \exists u : uv \in L\}$

IV. $\text{half}(L) = \{u \mid \exists v : |v| = |u| \text{ and } uv \in L\}$

Which of the constructions could lead to a non-regular language?

- a. Both I and IV
- b. Only 1
- c. Only IV
- d. Both II and III

Which choice of L is best suited to support your answer above?

- A. $(a+b)^*$
- B. $\{\epsilon, a, ab, bab\}$
- C. $(ab)^*$
- D. $\{a^n b^n \mid n \geq 0\}$

[gate2006-it](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.16.18 Regular Languages: GATE2007-31 [top](#)

<https://gateoverflow.in/1229>

Which of the following languages is regular?

- A. $\{ww^R \mid w \in \{0,1\}^+\}$
- B. $\{ww^Rx \mid x, w \in \{0,1\}^+\}$
- C. $\{wxw^R \mid x, w \in \{0,1\}^+\}$
- D. $\{xww^R \mid x, w \in \{0,1\}^+\}$

[gate2007](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.16.19 Regular Languages: GATE2007-7 [top](#)

<https://gateoverflow.in/1205>

Which of the following is TRUE?

- A. Every subset of a regular set is regular
- B. Every finite subset of a non-regular set is regular
- C. The union of two non-regular sets is not regular
- D. Infinite union of finite sets is regular

[gate2007](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

[Answer](#)

5.16.20 Regular Languages: GATE2008-53 [top](#)

<https://gateoverflow.in/476>

Which of the following are regular sets?

- I. $\{a^n b^{2m} \mid n \geq 0, m \geq 0\}$
- II. $\{a^n b^m \mid n = 2m\}$
- III. $\{a^n b^m \mid n \neq m\}$

IV. $\{xxy \mid x, y \in \{a, b\}^*\}$

- A. I and IV only
- B. I and III only
- C. I only
- D. IV only

gate2008 theory-of-computation normal regular-languages

[Answer](#)

5.16.21 Regular Languages: GATE2008-IT-35 [top](#)

<https://gateoverflow.in/3345>

Which of the following languages is (are) non-regular?

- $L_1 = \{0^m 1^n \mid 0 \leq m \leq n \leq 10000\}$
- $L_2 = \{w \mid w \text{ reads the same forward and backward}\}$
- $L_3 = \{w \in \{0, 1\}^* \mid w \text{ contains an even number of 0's and an even number of 1's}\}$

- A. L_2 and L_3 only
- B. L_1 and L_2 only
- C. L_3 only
- D. L_2 only

gate2008-it theory-of-computation normal regular-languages

[Answer](#)

5.16.22 Regular Languages: GATE2011-24 [top](#)

<https://gateoverflow.in/3429>

Let P be a regular language and Q be a context-free language such that $Q \subseteq P$. (For example, let P be the language represented by the regular expression p^*q^* and Q be $\{p^nq^n \mid n \in N\}$). Then which of the following is **ALWAYS** regular?

- A. $P \cap Q$
- B. $P - Q$
- C. $\Sigma^* - P$
- D. $\Sigma^* - Q$

gate2011 theory-of-computation easy regular-languages

[Answer](#)

5.16.23 Regular Languages: GATE2012-25 [top](#)

<https://gateoverflow.in/1609>

Given the language $L = \{ab, aa, baa\}$, which of the following strings are in L^* ?

1. abaabaaaabaa
2. aaaabaaaaa
3. baaaaabaaaab
4. baaaaabaa

- A. 1, 2 and 3
- B. 2, 3 and 4
- C. 1, 2 and 4
- D. 1, 3 and 4

gate2012 theory-of-computation easy regular-languages

Answer**5.16.24 Regular Languages: GATE2013-8** [top](#)<https://gateoverflow.in/1417>

Consider the languages $L_1 = \phi$ and $L_2 = \{a\}$. Which one of the following represents $L_1 L_2^* \cup L_1^*$?

- A. $\{\epsilon\}$
- B. ϕ
- C. a^*
- D. $\{\epsilon, a\}$

[gate2013](#) [theory-of-computation](#) [normal](#) [regular-languages](#)**Answer****5.16.25 Regular Languages: GATE2014-1-15** [top](#)<https://gateoverflow.in/1781>

Which one of the following is **TRUE**?

- A. The language $L = \{a^n b^n \mid n \geq 0\}$ is regular.
- B. The language $L = \{a^n \mid n \text{ is prime}\}$ is regular.
- C. The language $L = \{w \mid w \text{ has } 3k+1 \text{ }b's \text{ for some } k \in N \text{ with } \Sigma = \{a, b\}\}$ is regular.
- D. The language $L = \{ww \mid w \in \Sigma^* \text{ with } \Sigma = \{0, 1\}\}$ is regular.

[gate2014-1](#) [theory-of-computation](#) [regular-languages](#) [normal](#)**Answer****5.16.26 Regular Languages: GATE2014-2-15** [top](#)<https://gateoverflow.in/1971>

If $L_1 = \{a^n \mid n \geq 0\}$ and $L_2 = \{b^n \mid n \geq 0\}$, consider

- I. $L_1 \cdot L_2$ is a regular language
- II. $L_1 \cdot L_2 = \{a^n b^n \mid n \geq 0\}$

Which one of the following is CORRECT?

- A. Only I
- B. Only II
- C. Both I and II
- D. Neither I nor II

[gate2014-2](#) [theory-of-computation](#) [normal](#) [regular-languages](#)**Answer****5.16.27 Regular Languages: GATE2014-2-36** [top](#)<https://gateoverflow.in/1995>

Let $L_1 = \{w \in \{0, 1\}^* \mid w \text{ has at least as many occurrences of } (110)' \text{ s as } (011)' \text{ s}\}$. Let $L_2 = \{w \in \{0, 1\}^* \mid w \text{ has at least as many occurrences of } (000)' \text{ s as } (111)' \text{ s}\}$. Which one of the following is **TRUE**?

- A. L_1 is regular but not L_2
- B. L_2 is regular but not L_1
- C. Both L_1 and L_2 are regular
- D. Neither L_1 nor L_2 are regular

gate2014-2 theory-of-computation normal regular-languages

Answer

5.16.28 Regular Languages: GATE2015-2-51 [top](#)

<https://gateoverflow.in/8254>

Which of the following is/are regular languages?

$L_1 : \{wxw^R \mid w, x \in \{a, b\}^* \text{ and } |w|, |x| > 0\}$, w^R is the reverse of string w

$L_2 : \{a^n b^m \mid m \neq n \text{ and } m, n \geq 0\}$

$L_3 : \{a^p b^q c^r \mid p, q, r \geq 0\}$

- A. L_1 and L_3 only
- B. L_2 only
- C. L_2 and L_3 only
- D. L_3 only

gate2015-2 theory-of-computation normal regular-languages

Answer

5.16.29 Regular Languages: GATE2016-2-17 [top](#)

<https://gateoverflow.in/39542>

Language L_1 is defined by the grammar: $S_1 \rightarrow aS_1b \mid \epsilon$

Language L_2 is defined by the grammar: $S_2 \rightarrow abS_2 \mid \epsilon$

Consider the following statements:

- P: L_1 is regular
- Q: L_2 is regular

Which one of the following is **TRUE**?

- A. Both P and Q are true.
- B. P is true and Q is false.
- C. P is false and Q is true.
- D. Both P and Q are false.

gate2016-2 theory-of-computation normal regular-languages

Answer

5.16.30 Regular Languages: GATE2018-52 [top](#)

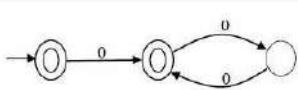
<https://gateoverflow.in/204127>

Given a language L , define L^i as follows:

$$L^0 = \{\epsilon\}$$

$$L^i = L^{i-1} \bullet L \text{ for all } i > 0$$

The order of a language L is defined as the smallest k such that $L^k = L^{k+1}$. Consider the language L_1 (over alphabet O) accepted by the following automaton.



The order of L_1 is _____

gate2018 theory-of-computation numerical-answers regular-languages

Answer

5.16.31 Regular Languages: ISI2014-CS-4b [top](#)

<https://gateoverflow.in/47445>

Consider the following statement:

For all languages $L \subseteq \{0,1\}^*$, if L^* is regular then L is regular.

Is the above statement true? Justify your answer.

descriptive isi2014 theory-of-computation regular-languages

Answer

5.16.32 Regular Languages: ISI2015-CS-5a [top](#)

<https://gateoverflow.in/47327>

Construct two nonregular languages L_1 and L_2 such that $L_1 \cup L_2$ is regular.

Prove that the languages L_1 and L_2 constructed above are nonregular and $L_1 \cup L_2$ is regular.

descriptive isi2015 theory-of-computation regular-languages

Answer

5.16.33 Regular Languages: TIFR2013-B-6 [top](#)

<https://gateoverflow.in/25667>

Let L and L' be languages over the alphabet Σ . The left quotient of L by L' is

$$L/L' \stackrel{\text{def}}{=} \{w \in \Sigma^* : wx \in L \text{ for some } x \in L'\}$$

Which of the following is true?

- A. If L/L' is regular then L' is regular.
- B. If L is regular then L/L' is regular.
- C. If L/L' is regular then L is regular.
- D. L/L' is a subset of L .
- E. If L/L' and L' are regular, then L is regular.

tifr2013 theory-of-computation regular-languages

Answer

5.16.34 Regular Languages: TIFR2013-B-8 [top](#)

<https://gateoverflow.in/25670>

Which one of the following languages over the alphabet $0, 1$ is regular?

- A. The language of balanced parentheses where $0, 1$ are thought of as $(,)$ respectively.
- B. The language of palindromes, i.e. bit strings x that read the same from left to right as well as right to left.
- C. $L = \{0^{m^2} : 3 \leq m\}$
- D. The Kleene closure L^* , where L is the language in (c) above.
- E. $\{0^m 1^n : 1 \leq m \leq n\}$

tifr2013 theory-of-computation regular-languages

Answer

5.16.35 Regular Languages: TIFR2014-B-12 [top](#)

<https://gateoverflow.in/27314>

Consider the following three statements:

- i. Intersection of infinitely many regular languages must be regular.
- ii. Every subset of a regular language is regular.
- iii. If L is regular and M is not regular then $L \bullet M$ is necessarily not regular.

Which of the following gives the correct true/false evaluation of the above?

- A. true, false, true.
- B. false, false, true.
- C. true, false, true.
- D. false, false, false.
- E. true, true, true.

[tifr2014](#) [theory-of-computation](#) [regular-languages](#)

Answer

5.16.36 Regular Languages: TIFR2015-B-10 [top](#)

<https://gateoverflow.in/30039>

Consider the languages

$$L_1 = \{a^m b^n c^p \mid (m = n \vee n = p) \wedge m + n + p \geq 10\}$$

$$L_2 = \{a^m b^n c^p \mid (m = n \vee n = p) \wedge m + n + p \leq 10\}$$

State which of the following is true?

- A. L_1 and L_2 are both regular.
- B. Neither L_1 nor L_2 is regular.
- C. L_1 is regular and L_2 is not regular.
- D. L_1 is not regular and L_2 is regular.
- E. Both L_1 and L_2 are infinite.

[tifr2015](#) [regular-languages](#)

Answer

5.16.37 Regular Languages: TIFR2015-B-6 [top](#)

<https://gateoverflow.in/29860>

Let B consist of all binary strings beginning with a 1 whose value when converted to decimal is divisible by 7.

- A. B can be recognized by a deterministic finite state automaton.
- B. B can be recognized by a non-deterministic finite state automaton but not by a deterministic finite state automaton.
- C. B can be recognized by a deterministic push-down automaton but not by a non-deterministic finite state automaton.
- D. B can be recognized by a non-deterministic push-down automaton but not by a deterministic push-down automaton.
- E. B cannot be recognized by any push down automaton, deterministic or non-deterministic.

[tifr2015](#) [theory-of-computation](#) [regular-languages](#)

Answer

Answers: Regular Languages

5.16.1 Regular Languages: CMI2012-A-01 top

<https://gateoverflow.in/46529>



Selected Answer

- A. **If L is regular, all subsets of L are regular.**

False.

Counter Ex. $L = \{a + b\}^*$ then its subset $\{a^n b^n; n > 0\}$ is not regular.

- B. **If all proper subsets of L are regular, then L is regular.**

True.

Proof: Take any proper subset of L says L_1 , L_1 is regular by definition.

$L_2 = L - L_1$.

L_2 is also regular because it is also a proper subset of L .

And we know that Union of two regular set is also regular.

$L_1 \cup L_2 = L$. Hence L is also regular.

I have a strong feeling that L is finite that's why all its proper subsets are regular.

- C. **If all finite subsets of L are regular, then L is regular.**

False.

Counter Ex: $L = \{a^n b^n; n > 0\}$, L is not regular, it is context free.

Now take any finite subset of L , this will be regular.

Note: Take only, finite subset of L.

- D. **If a proper subset of L is not regular, then L is not regular.**

False

e.g., $L = \{a + b\}^*$ is regular, Its proper subset $\{a^n b^n; n > 0\}$ is not regular.

16 votes

-- Hemant Parihar (14.8k points)

5.16.2 Regular Languages: CMI2015-A-09 top

<https://gateoverflow.in/47046>



Selected Answer

Answer should be option D.

Contradiction for A. Let $L_2 = \{a, b\}^*$... which is regular.

And $L_1 = a^n b^n$ which is CFL but not regular.

And here L_1 is subset of L_2 .

Contradiction for B. Let $L_1 = ab$,

which is regular. And $L_2 = a^n b^n$ which is CFL but not regular.

And here L_1 is subset of L_2 .

C → False, (reason A and B).

12 votes

-- Dhananjay Kumar Sharma (25.7k points)

5.16.3 Regular Languages: GATE1987-2h top

<https://gateoverflow.in/80589>**True!**Reverse all the edges and interchange final and initial states in **DFA**.

8 votes

-- Motamarri Anusha (12.3k points)

5.16.4 Regular Languages: GATE1987-2i top

<https://gateoverflow.in/80590>**False!** a^*b^* is regular but its subset $a^n b^n$ is not regular

12 votes

-- Motamarri Anusha (12.3k points)

5.16.5 Regular Languages: GATE1990-3-viii top

<https://gateoverflow.in/84837>

Regular Languages are closed under

1. Intersection
2. Union
3. Complement
4. Kleen-Closure

 $\Sigma^* - R_1$ is the complement of R_1 **B,C** are true.

15 votes

-- pC (22k points)

5.16.6 Regular Languages: GATE1991-03,xiv top

<https://gateoverflow.in/528>

(b), (c) and (d) are true. But the strongest answer would be **(d)** a regular language. It is trivial to say that a finite set of strings (finite language) can be accepted using a finite set of states. And regular language \subset context-free \subset context-sensitive \subset Turing recognizable, would imply that regular language is the strongest answer.

19 votes

-- gatecse (18k points)

5.16.7 Regular Languages: GATE1995-2.24 top

<https://gateoverflow.in/2637>

Selected Answer

Answer is (C). $L \cup R$ is nothing but L as R is a subset of L and hence regular. R is deterministic context-free but not regular as we require a stack to keep the count of 0's to match that of 1's.

15 votes

-- Arjun Suresh (350k points)

5.16.8 Regular Languages: GATE1996-1.10 [top](#)

<https://gateoverflow.in/2714>



Selected Answer

Only (d). because n and m are independent and thus no memory element required.

a and b are same and are DCFL's.

c is $L = \{a^n b^m \mid n > m\}$. which is not regular.

Correction: I think c should be that x has more a 's than b 's.

9 votes

-- Gate Keeda (19.6k points)

5.16.9 Regular Languages: GATE1998-2.6 [top](#)

<https://gateoverflow.in/1678>



Selected Answer

(b) is False. Any language is a subset of Σ^* which is a regular set. So, if we take any non-regular language, it is a subset of a regular language.

(a) and (c) are regular as any finite language is regular.

(d) is regular as regular set is closed under intersection.

17 votes

-- Arjun Suresh (350k points)

5.16.10 Regular Languages: GATE1999-6 [top](#)

<https://gateoverflow.in/1505>



Selected Answer

a. A is regular, $A \cup B$ is regular, then B is not necessarily regular
 example :- $A = (a+b)^*$ $B = a^n b^n \quad n \geq 0$ $A \cup B$ is $(a+b)^*$ while B is not regular.

b. We have two machines M_1 and M_2
 draw a DFA using M_1 and M_2 where start state is, say, $p_0 q_0$ (where p_0 is start state in M_1 and q_0 is start state in M_2)
 $\delta(p_0 q_0, 0) = \delta(p_0, 0) \cup \delta(q_0, 0)$
 if $L(M_1) \subseteq L(M_2)$

Then final state of M_1 will come together with final state of M_2 , while Final state of M_2 can come alone.

i.e. all inputs of M_1 are also in machine M_2 , and there may be different inputs in M_2 .

22 votes

-- Praveen Saini (54.6k points)

5.16.11 Regular Languages: GATE2000-2.8 top

<https://gateoverflow.in/655>



Selected Answer

Answer is (D). Either L must be $\{a^n \mid n \text{ is odd}\}$, or L must be $\{a^n \mid n \text{ is even}\}$

Because if we draw the minimal DFA for each of them, we will get two states each. Whereas, $\{a^n \mid n \geq 0\}$ requires only one state.

1 24 votes

-- Keith Kr (6.1k points)

5.16.12 Regular Languages: GATE2001-1.4 top

<https://gateoverflow.in/697>



Selected Answer

Only S_1 is correct!

A **DFA** with 3 states will be needed, as the strings in the language S_1 are 00, 0000, 000000, and so on. which is the set of all strings with even number of 0's and with length greater than 0. We would have needed only 2 states had empty string also been in the language but $n \geq 1$ prohibits it and so we need 3 states in our DFA. This assumes that the language is over {0} and not {0,1}.

S_2 is **DCFL** as we need to do infinite counting of 0's and 1's here.

1 15 votes

-- Bhagirathi Nayak (14.1k points)

5.16.13 Regular Languages: GATE2001-2.6 top

<https://gateoverflow.in/724>



Selected Answer

$L_1 = \{ww \mid w \in \{a,b\}^*\}$ CSL

$L_2 = \{ww^R \mid w \in \{a,b\}^*, w^R \text{ is the reverse of } w\}$ Palindrome, so CFL

$L_3 = \{0^{2i} \mid i \text{ is an integer}\}$ Linear power and regular expression can be stated as $(00)^*$

$L_4 = \{0^{i^2} \mid i \text{ is an integer}\}$ Non-linear power, so CSL

Therefore, answer is option D.

1 16 votes

-- Umang Raman (15.7k points)

5.16.14 Regular Languages: GATE2006-29 top

<https://gateoverflow.in/992>



Selected Answer

A. There are only finite 3 digit primes. Let the largest of them be X . Now we need $X + 2$ states in our DFA (including one for $count = 0$ and one for $count > X$). We do not need a change of state for any 1.

B. Here we need just 6 states to recognise L .

1. $\#0 - \#1 = 0$
2. $\#0 - \#1 = 1$
3. $\#0 - \#1 = 2$
4. $\#0 - \#1 = -1$
5. $\#0 - \#1 = -2$

If the difference goes above **2** or below **-2**, we go to a dead state. All other states are accepting. This transition to dead state is possible because of the words "for every prefix s' of s " in L and that is what makes this language regular.

C. L is not regular

```
#0 - #1 = 1
#0 - #1 = 2
#0 - #1 = 3
#0 - #1 = 4
#0 - #1 = 5
:
:
#0 - #1 = 1000
:
```

All these form distinct equivalent classes under Myhill-Nerode theorem meaning from the strings in each of these sets, we can append a string which will take the new string to \bar{L} , while the same string appended to string in any other set would not reach \bar{L} .

For example, for **000000**, we append **11**, for **0000000**, we append **111** etc. So, in short we need to maintain the count of **1's** and **0's** and the count here is not finite.

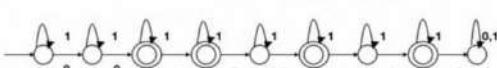
D. This is regular. We need a finite automata with $5 \times 7 = 35$ states for maintaining the counts of **0's mod 7** and **1's mod 5** and there cannot be more than **35** possibilities for this. With each input symbol, the transition must be going to one among these.

42 votes

-- Arjun Suresh (350k points)

A. $n_0(s)$ is a **3 digit prime**. It means no. of **0's** are in the range (set) $\{101, 103, \dots, 997\}$ which is finite. So, L will be regular.

For simplicity consider $n_0(s)$ is a **1 digit prime**. So set will be $\{2, 3, 5, 7\}$. DFA will look like

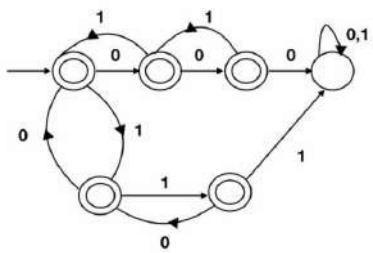


B. Prefix - For a string s , its prefix s' is any substring from the beginning of s . Example: $s = 10110, s' = \{1, 10, 101, 1011, 10110\}$.

$$L = \{s \in (0+1)^* \mid \text{for every prefix } s' \text{ of } s, |n_0(s') - n_1(s')| \leq 2\}$$

So, here we have to ensure that nowhere in our string the difference between number of **0's** and **1's** is greater than **2**. If anywhere the above condition fails, we will enter a dead state as we got a substring (prefix) that violates the condition.

DFA will be like



So, here the difference between number of 0's and 1's crosses 2 we will be in dead state.
e.g., $s = 10110, s' = \{1, 10, 101, 1011, 10110\}$

This string will be accepted as nowhere in s' the difference exceeds 2.

$s = 011110, s' = \{0, 01, 011, 0111, \underbrace{01111}_{\text{diff=3}}, 011110\}$

This string is not accepted.

C. The difference between the number of 0's and 1's should not exceed 4.

Here, once we exceed the limit there is a possibility that we might cover up the difference in further part of the string.

e.g., 0111111000

Here, just by seeing up to the last 1 we cannot move to dead state. So, it needs infinite counting. Hence, not regular.

D. This language can be stated as the set of strings where number of 0's is divisible by 7 and number of 1's is divisible by 5. We can make a DFA for this by cross-product method. Hence, regular.

59 votes

-- Rajendra Dangwal (1.1k points)

5.16.15 Regular Languages: GATE2006-IT-30 top

<https://gateoverflow.in/3569>



Selected Answer

Option C is not True.

- A. Every language has a regular superset: True. Σ^* is such a superset.
- B. Every language has a regular subset: True. \emptyset is such a subset.
- C. Every subset of a regular language is regular: False. $a^n b^n \subset \Sigma^*$, but $a^n b^n$ is not Regular.
- D. Every subset of a finite language is regular: True. Every subset of a finite set must be finite by definition. Every finite set is regular. Hence, every subset of a finite language is regular.

31 votes

-- Pragy Agarwal (20.6k points)

5.16.16 Regular Languages: GATE2006-IT-80 top

<https://gateoverflow.in/3624>



Selected Answer

Correct answer is B. Only I.

Repeat(L) = $\{ww \mid w \in L\}$ is non regular language

Half(L), suffix(L), and prefix(L) are regular languages

Reference:

https://gateoverflow.in/3637/gate2006-it_81

👍 9 votes

-- Praveen Saini (54.6k points)

5.16.17 Regular Languages: GATE2006-IT-81 [top](#)

<https://gateoverflow.in/3637>



Selected Answer

- I. $\text{repeat}(L)$ is non regular
<http://www.cs.odu.edu/~toida/nerzic/390teched/regular/reg-lang/non-regularity.html> [example 2]
- II. $\text{prefix}(L)$ is regular
<http://www.public.asu.edu/~ccolbou/src/355hw2sols09.pdf> [2(a)]
- III. $\text{suffix}(L)$ is regular
<http://www.public.asu.edu/~ccolbou/src/355hw2sols09.pdf> [2(b)]
- IV. $\text{Half}(L)$ is regular
<https://www.complang.tuwien.ac.at/lkovacs/ATCSHW/hw4-sol.pdf> [4.2(a)]

So, in **first part of question**, option **B** is correct only (i).

[Need someone to explain these language with more clear explanation @Arjun. that can be understandable by an average student]

For **second part of question**:

A is answer.

Explanation is in comment given below by Arjun (for option *A, C* and *D*)

Note : for option *B* $L = \{\epsilon, a, ab, bab\}$, $\text{Repeat}(L) = \{\epsilon, aa, abab, babbab\}$ is regular .

👍 23 votes

-- Praveen Saini (54.6k points)

5.16.18 Regular Languages: GATE2007-31 [top](#)

<https://gateoverflow.in/1229>



Selected Answer

- A. CFL
- B. CFL
- C. Regular, language is string starting and ending with the same symbol and having length at least 3. e.g. 0x0 or 1x1
- D. CFL

http://gatcse.in/wiki/Identify_the_class_of_the_language

👍 20 votes

-- Vikrant Singh (13.5k points)

5.16.19 Regular Languages: GATE2007-7 [top](#)

<https://gateoverflow.in/1205>



Selected Answer

(B) Every finite subset of a non-regular set is regular.

Any finite set is always regular.

Σ^* being regular any non regular language is a subset of this, and hence (A) is false.

If we take a CFL which is not regular, and takes union with its complement (complement of a CFL which is not regular won't be regular as regular is closed under complement), we get Σ^* which is regular. So, (C) is false.

Regular set is not closed under infinite union. Example:

Let $L_i = \{0^i 1^i\}, i \in N$

Now, if we take infinite union over all i , we get

$L = \{0^i 1^i \mid i \in N\}$, which is not regular.

So, (D) is false.

Upvote 21 votes

-- Omesh Pandita (2.5k points)

5.16.20 Regular Languages: GATE2008-53 top

<https://gateoverflow.in/476>



Selected Answer

Answer is A.

Since in option 2 and 3, n is dependent on m , therefore a comparison has to be done to evaluate those and hence are not regular.

I and IV are clearly regular sets.

Upvote 19 votes

-- Gate Keeda (19.6k points)

5.16.21 Regular Languages: GATE2008-IT-35 top

<https://gateoverflow.in/3345>



Selected Answer

L_1 is regular.. since 10000 is finite. So, finite states are required.

L_3 is also regular. We can make DFA for L_3 . States will represent mod 2 for 0 and mod 2 for 1, which is finite

L_2 is non regular..It is CFG $S \rightarrow aSa \mid \dots \mid zSz \mid \epsilon \mid [a - z]$

So, option is(D).

Upvote 20 votes

-- Vicky Bajoria (5k points)

5.16.22 Regular Languages: GATE2011-24 top

<https://gateoverflow.in/3429>



Selected Answer

(C) complement of regular Language is regular

Like 28 votes

-- VOOTLA SRINIVAS (335 points)

5.16.23 Regular Languages: GATE2012-25 top

<https://gateoverflow.in/1609>



Selected Answer

$$L = \{ab, aa, baa\}$$

- | | |
|---|---|
| 1. $abaabaaabaa = ab aa baa ab aa$
2. $aaaabaaaa = aa aa baa aa$
3. $baaaaabaaaab = baa aa ab aa aa b$
4. $baaaaabaa = baa aa ab aa$ | belong to L^* (combinations of strings in L)
belong to L^*
does not belong to L^*
belong to L^* |
|---|---|

Like 20 votes

-- Praveen Saini (54.6k points)

5.16.24 Regular Languages: GATE2013-8 top

<https://gateoverflow.in/1417>



Selected Answer

Concatenation of empty language with any language will give the empty language and $L_1^* = \phi^* = \epsilon$.

Therefore,

$$\begin{aligned} L_1 L_2^* \cup L_1^* \\ &= \phi \cdot (L_2)^* \cup \phi^* \\ &= \phi \cup \{\epsilon\} (\because \phi \text{ concatenated with anything is } \phi \text{ and } \phi^* = \{\epsilon\}) \\ &= \{\epsilon\}. \end{aligned}$$

Hence, option **(A)** is true.

PS: $\phi^* = \epsilon$, where ϵ is the regular expression and the language it generates is $\{\epsilon\}$.

Like 39 votes

-- Praveen Saini (54.6k points)

5.16.25 Regular Languages: GATE2014-1-15 top

<https://gateoverflow.in/1781>



Selected Answer

(A) is CFL and **(B)** and **(D)** are CSL.

(C) is regular and regular expression for **(C)** would be:

$$a^* b (a^* b a^* b a^* b)^+ a^*$$

Like 20 votes

-- Arjun Suresh (350k points)

5.16.26 Regular Languages: GATE2014-2-15 top

<https://gateoverflow.in/1971>

**Option A.**

$$L_1 = \{\varepsilon, a, aa, aaa, aaaa, \dots\}$$

$$L_2 = \{\varepsilon, b, bb, bbb, bbbb, \dots\}$$

$$L_1 \cdot L_2 = \left\{ \begin{array}{ll} \varepsilon, & \\ a, & b, \\ aa, & ab, bb \\ aaa, & aab, abb, bbb, \\ aaaa, & aaab, aabb, abbb, bbbb, \dots \end{array} \right\}$$

$$L_1 \cdot L_2 = a^* b^*$$

Thus, $L_1 \cdot L_2$ is Regular.

(Also, since both L_1 and L_2 are Regular, their concatenation has to be Regular since Regular languages are closed under concatenation)

However, $L_1 \cdot L_2 \neq a^n b^n$. This is because in $a^* b^*$, the number of a 's and b 's can be different whereas in $a^n b^n$ they have to be the same.

33 votes

-- Viral Kapoor (2.1k points)

5.16.27 Regular Languages: GATE2014-2-36 top

<https://gateoverflow.in/1995>



(A) is True. Though at first look both L_1 and L_2 looks non-regular, L_1 is in fact regular. The reason is the relation between 110 and 011.

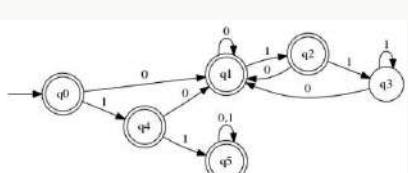
We cannot have two 110's in a string without a 011 or vice versa. And this would mean that we only need a finite number of states to check for acceptance of any word in this language.

That was just an intuitive explanation. Now I say that L contains all binary strings starting with 11. Yes, if a binary string starts with 11, it can never have more no. of 011 than 110.

Lets take an example:

11 011 011 -There are two 011's. But there are also two 110's. Similarly for any binary string starting with 11.

Using this property, DFA for L_1 can be constructed as follows:



68 votes

-- gatecse (18k points)

5.16.28 Regular Languages: GATE2015-2-51 [top](#)

<https://gateoverflow.in/8254>

Selected Answer

Answer is **A**.

L_1 : all strings of length 3 or more with same start and end letter- because everything in middle can be consumed by x as per the definition of L .

L_2 : We need to compare number of a 's and b 's and these are not bounded. So, we need at least a **DPDA**.

L_3 : Any number of a 's followed by any number of b 's followed by any number of c 's. Hence regular.

30 votes

-- Vikrant Singh (13.5k points)

5.16.29 Regular Languages: GATE2016-2-17 [top](#)

<https://gateoverflow.in/39542>

Selected Answer

Answer is **C**.

$$S_1 \rightarrow aS_1b|\epsilon$$

$L_1 = \{a^n b^n \mid n \geq 0\}$ is CFL

$$S_2 \rightarrow abS_2|\epsilon$$

$L_2 = \{(ab)^n \mid n \geq 0\}$ is Regular having regular expression $(ab)^*$

32 votes

-- Praveen Saini (54.6k points)

5.16.30 Regular Languages: GATE2018-52 [top](#)

<https://gateoverflow.in/204127>

Selected Answer

$$L_1 = \{\epsilon + 0(00)^*\}$$

Now that is given language L , we have find order of it.

$$L^0 = \{\epsilon\}$$

$$L^1 = L^0 \cdot L = \{\epsilon\} \cdot \{\epsilon + 0(00)^*\} = \{\epsilon + 0(00)^*\}$$

$$L^2 = L^1 \cdot L = \{\epsilon + 0(00)^*\} \cdot \{\epsilon + 0(00)^*\}$$

$$= \{\epsilon + 0(00)^* + 0(00)^*0(00)^*\}$$

$$L^2 = \{0^*\}$$

$$\begin{aligned} L^3 &= L^2 \cdot L = \{0^*\} \cdot \{\epsilon + 0(00)^*\} \\ &= \{0^* + 0^*0(00)^*\} \\ L^3 &= \{0^*\} \end{aligned}$$

Order of L is 2 such that $L^2 = L^{2+1}$

24 votes

-- Praveen Saini (54.6k points)

5.16.31 Regular Languages: ISI2014-CS-4b top

<https://gateoverflow.in/47445>



Selected Answer

If \mathbf{L}^* is regular, then \mathbf{L} is not necessarily regular.

Let's take one possible example.

Let $\Sigma = \{a\}$ and consider the language $\mathbf{L} = \{a^{2n} \mid n \in \mathbb{N}\}$.

This language is not regular

and the language \mathbf{L}^* is the language a^* , which is regular. To see this, notice that since \mathbf{L} contains the string a, the language \mathbf{L}^* contains all strings of the form a^n for any natural number n .

Hence, **Option(2) Need not Regular** is the correct choice.

9 votes

-- Leen Sharma (40k points)

5.16.32 Regular Languages: ISI2015-CS-5a top

<https://gateoverflow.in/47327>



Selected Answer

Consider a language $L_1 = \{a^n b^n \mid n \geq 0\}$ is a non regular language .

And take another language $L_2 = L_1^c$ which is also non regular .(Since regular set is closed under complementation.)

As we know **the union of any language and its complement is Σ^*** .

So,
 $L_1 \cup L_2 = \Sigma^*$ and this is regular.

5 votes

-- Manoj Kumar (38.6k points)

5.16.33 Regular Languages: TIFR2013-B-6 top

<https://gateoverflow.in/25667>



Selected Answer

- a. False because - $L = a^*b^*$, $L' = a^n b^n$ Here $L/L' = a^*.L/L'$ is regular, but L' is not.
- b. True. If L is regular, L/L' is prefix of language. Regular languages are closed under Quotient/Prefix. So this is correct.
- c. False $L' = \text{Empty set}$. Then L/L' is Empty set whatever L is. Here L can be say $a^n b^n$. See

definition of L/L' to see why L/L' should be empty set.

- d. False because L/L' can accept prefixes of string of Language L , which may or may not be accepted by L itself. So L/L' is not subset. (It is not Superset either , because L' can be empty set)
- e. False. Same explanation as C.

Answer is B.

8 votes

-- Akash Kanase (42.5k points)

5.16.34 Regular Languages: TIFR2013-B-8 [top](#)

<https://gateoverflow.in/25670>



Selected Answer

Here, **option D** is **regular**, reason is as follows:

$$L = \{0^{m^2} : m \geq 3\} 0^m 20$$

Now, in

L^* if we can **generate**

9 continuous powers of zero, then further every power can be generated by **concatenating** 0^9 .

$$\text{Here, } L = \{0^9, 0^{16}, 0^{25}, \dots\}$$

So, here are **9 continuous powers**:

$$0^{120} : 0^{16} 0^{16} 0^{16} 0^9 0^9 0^9 0^9 0^9 0^9 0^9$$

$$0^{121} : 0^{16} 0^{16} 0^{16} 0^{16} 0^{16} 0^{25}$$

$$0^{122} : 0^{16} 0^{16} 0^9 0^9 0^9 0^9 0^9 0^9 0^9 0^9$$

$$0^{123} : 0^{16} 0^{16} 0^{16} 0^{25} 0^{25}$$

$$0^{124} : 0^{16} 0^{18} 0^{18} 0^{18} 0^{18} 0^{18}$$

$$0^{125} : 0^{25} 0^{25} 0^{25} 0^{25} 0^{25}$$

$$0^{126} : 0^{18} 0^{18} 0^{18} 0^{18} 0^{18} 0^{18} \quad \{0^{18} \text{ can be generated as } 0^9 0^9\}$$

$$0^{127} : 0^{16} 0^{16} 0^{16} 0^{16} 0^9 0^9 0^9 0^9 0^9 0^9$$

$$0^{128} : 0^{16} 0^{16} 0^{16} 0^{16} 0^{16} 0^{16} 0^{16}$$

Now, 0^{129} can be given as $0^{120} 0^9$ and so on.

Every Further powers can be generated by concatenating 0^9 .

13 votes

-- Himanshu Agarwal (15.3k points)

5.16.35 Regular Languages: TIFR2014-B-12 [top](#)

<https://gateoverflow.in/27314>



Selected Answer

i. False

Regular Languages are not closed under Infinite Union and Intersection

$$L_1 \cup L_2 \cup L_3 \cup L_4 \cup \dots$$

For example :

$$\begin{aligned} ab \cup aabb \cup aaabbb \cup aaaabbbb \cup \dots \\ = \{a^n b^n, n \geq 1\} \end{aligned}$$

So Infinite Union is not closed

$$L_1 \cap L_2 \cap L_3 \cap L_4 \cap \dots$$

$$= (L_1' \cup L_2' \cup L_3' \cup L_4' \cup \dots)'$$

As Infinite Union is not closed , So Infinite Intersection is also not closed

ii. False.

$a^* b^*$ is regular

its subset $a^n b^n, n \geq 1$ is not regular

a^* is regular

a^p, p is prime, is not regular

iii. False

$L = \{\}$ is regular

M be non-regular like $\{0^n 1^n \mid n > 0\}$.

$L \cdot M = \{\}$, is regular

20 votes

-- Praveen Saini (54.6k points)

5.16.36 Regular Languages: TIFR2015-B-10 top

<https://gateoverflow.in/30039>



Selected Answer

L_2 is finite, so regular.

L_1 is non-regular.

(It seems CFL to me as I think it can be implemented with the help of PDA , as stack can ensure $(m = n \vee n = p)$ and we can also ensure $(m + n + p \geq 10)$ with minimum states changes along with transitions).

L_2 is actually $\{c^p \mid p \leq 10\} \cup \{abc^p \mid p \leq 8\} \cup \{a^2b^2c^p \mid p \leq 6\} \cup \{a^3b^3c^p \mid p \leq 4\} \cup \{a^4b^4c^p \mid p \leq 2\} \cup \{a^5b^5\} \cup \{a^p \mid p \leq 10\} \cup \{a^pbc \mid p \leq 8\} \cup \{a^pb^2c^2 \mid p \leq 6\} \cup \{a^pb^3c^3 \mid p \leq 4\} \cup \{a^pb^4c^4 \mid p \leq 2\} \cup \{b^5c^5 \mid p \leq 10\}$

18 votes

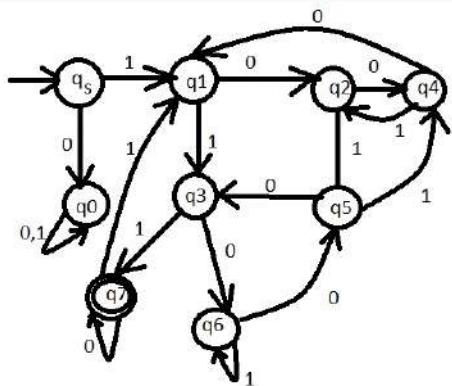
-- Praveen Saini (54.6k points)

5.16.37 Regular Languages: TIFR2015-B-6 top

<https://gateoverflow.in/29860>



Selected Answer



Answer will be (A).

If it starts with 1 it goes to final state.

If it starts with 0 it will go to the reject state.

12 votes

-- srestha (87k points)

5.17

Turing Machine(8) top

5.17.1 Turing Machine: GATE1987-2m top

<https://gateoverflow.in/80606>

State whether the following statements are TRUE or FALSE:

The problem as to whether a Turing machine M accepts input w is undecidable.

gate1987 theory-of-computation turing-machine decidability

Answer

5.17.2 Turing Machine: GATE2001-7 top

<https://gateoverflow.in/748>

Let a decision problem X be defined as follows:

X : Given a Turing machine M over Σ and any word $w \in \Sigma$, does M loop forever on w ?

You may assume that the halting problem of Turing machine is undecidable but partially decidable.

- A. Show that X is undecidable
- B. Show that X is not even partially decidable

gate2001 theory-of-computation decidability turing-machine easy descriptive

Answer

5.17.3 Turing Machine: GATE2002-14 top

<https://gateoverflow.in/867>

The aim of the following question is to prove that the language $\{M \mid M \text{ is the code of the Turing Machine which, irrespective of the input, halts and outputs a } 1\}$, is undecidable. This is to be done by reducing from the language $\{M', x \mid M' \text{ halts on } x\}$, which is known to be undecidable. In parts (a) and (b) describe the 2 main steps in the construction of M . In part (c) describe the key property which relates the behaviour of M on its input w to the behaviour of M' on x .

- A. On input w , what is the first step that M must make?
 B. On input w , based on the outcome of the first step, what is the second step M must make?
 C. What key property relates the behaviour of M on w to the behaviour of M' on x ?

gate2002 theory-of-computation decidability normal turing-machine descriptive difficult

Answer

5.17.4 Turing Machine: GATE2003-53 [top](#)

<https://gateoverflow.in/941>

A single tape Turing Machine M has two states q_0 and q_1 , of which q_0 is the starting state. The tape alphabet of M is $\{0, 1, B\}$ and its input alphabet is $\{0, 1\}$. The symbol B is the blank symbol used to indicate end of an input string. The transition function of M is described in the following table.

	0	1	B
q_0	$q_1, 1,$ R	$q_1, 1,$ R	<i>Halt</i>
q_1	$q_1, 1,$ R	$q_0, 1,$ L	$q_0, B,$ L

The table is interpreted as illustrated below.

The entry $(q_1, 1, R)$ in row q_0 and column 1 signifies that if M is in state q_0 and reads 1 on the current page square, then it writes 1 on the same tape square, moves its tape head one position to the right and transitions to state q_1 .

Which of the following statements is true about M ?

- A. M does not halt on any string in $(0 + 1)^+$
 B. M does not halt on any string in $(00 + 1)^*$
 C. M halts on all strings ending in a 0
 D. M halts on all strings ending in a 1

gate2003 theory-of-computation turing-machine normal

Answer

5.17.5 Turing Machine: GATE2003-54 [top](#)

<https://gateoverflow.in/355>

Define languages L_0 and L_1 as follows :

$$L_0 = \{\langle M, w, 0 \rangle \mid M \text{ halts on } w\}$$

$$L_1 = \{\langle M, w, 1 \rangle \mid M \text{ does not halt on } w\}$$

Here $\langle M, w, i \rangle$ is a triplet, whose first component M is an encoding of a Turing machine, second component w is a string, and third component i is a bit.

Let $L = L_0 \cup L_1$. Which of the following is true?

- A. L is recursively enumerable, but L' is not
 B. L' is recursively enumerable, but L is not
 C. Both L and L' are recursive
 D. Neither L nor L' is recursively enumerable

theory-of-computation turing-machine gate2003 difficult

Answer**5.17.6 Turing Machine: GATE2004-89** [top](#)<https://gateoverflow.in/1083>

L_1 is a recursively enumerable language over Σ . An algorithm A effectively enumerates its words as w_1, w_2, w_3, \dots . Define another language L_2 over $\Sigma \cup \{\#\}$ as $\{w_i \# w_j \mid w_i, w_j \in L_1, i < j\}$. Here $\#$ is new symbol. Consider the following assertions.

- $S_1 : L_1$ is recursive implies L_2 is recursive
- $S_2 : L_2$ is recursive implies L_1 is recursive

Which of the following statements is true?

- Both S_1 and S_2 are true
- S_1 is true but S_2 is not necessarily true
- S_2 is true but S_1 is not necessarily true
- Neither is necessarily true

[gate2004](#) [theory-of-computation](#) [turing-machine](#) [difficult](#)
Answer**5.17.7 Turing Machine: GATE2014-2-35** [top](#)<https://gateoverflow.in/1994>

Let $\langle M \rangle$ be the encoding of a Turing machine as a string over $\Sigma = \{0, 1\}$. Let

$$L = \{\langle M \rangle \mid M \text{ is a Turing machine that accepts a string of length 2014}\}.$$

Then L is:

- decidable and recursively enumerable
- undecidable but recursively enumerable
- undecidable and not recursively enumerable
- decidable but not recursively enumerable

[gate2014-2](#) [theory-of-computation](#) [turing-machine](#) [normal](#)
Answer**5.17.8 Turing Machine: GATE2016-2-44** [top](#)<https://gateoverflow.in/39596>

Consider the following languages.

- $L_1 = \{\langle M \rangle \mid M \text{ takes at least 2016 steps on some input}\},$
- $L_2 = \{\langle M \rangle \mid M \text{ takes at least 2016 steps on all inputs}\}$ and
- $L_3 = \{\langle M \rangle \mid M \text{ accepts } \epsilon\},$

where for each Turing machine M , $\langle M \rangle$ denotes a specific encoding of M . Which one of the following is TRUE?

- L_1 is recursive and L_2, L_3 are not recursive
- L_2 is recursive and L_1, L_3 are not recursive
- L_1, L_2 are recursive and L_3 is not recursive
- L_1, L_2, L_3 are recursive

[gate2016-2](#) [theory-of-computation](#) [turing-machine](#)

Answer

Answers: Turing Machine

5.17.1 Turing Machine: GATE1987-2m top

<https://gateoverflow.in/80606>


Yes. The problem as to whether a Turing machine M accepts input w is undecidable which is the well known Halting Problem.

If string w is going to loop then we can not determine if it will be eventually accepted by TM or not.

10 votes

-- Prashant Singh (59.8k points)

5.17.2 Turing Machine: GATE2001-7 top

<https://gateoverflow.in/748>


The question asks if M loop forever on w . If M loop forever on w , M wouldn't halt on w . And if M doesn't halt on w , M should loop forever. So, this problem is exactly same as asking if " M doesn't halt on w ", which is the complement of halting problem and is not even partially decidable. So, X is not even partially decidable.

21 votes

-- Arjun Suresh (350k points)

5.17.3 Turing Machine: GATE2002-14 top

<https://gateoverflow.in/867>


- M erases its input w and simulate the moves of M' on x . Thus if M' halts on x , M accepts any input (Σ^*) and if M' doesn't halt on x , M accepts no string (ϕ)
- Give the description of $M - \langle M \rangle$ to the TM that decides L . If TM accepts $\langle M \rangle$, M halts on all inputs $\rightarrow M'$ accept x . If TM rejects $\langle M \rangle$, M doesn't halt on some input $\rightarrow M'$ doesn't halt on x , due to our construction of M in 1st step. Thus we decide halting problem
- M halting on all inputs w is the key property relating to M' which is halting on a given input x

4 votes

-- Arjun Suresh (350k points)

5.17.4 Turing Machine: GATE2003-53 top

<https://gateoverflow.in/941>


Option A. Or epsilon is only accepted i.e tape contain **B** as the first character.

15 votes

-- Supromit Roy (795 points)

5.17.5 Turing Machine: GATE2003-54 top

<https://gateoverflow.in/355>



Both L and L' are undecidable and not even semi-decidable (not recursively-enumerable). Because halting problem can be solved with both L and L' .

Halting problem can be stated as follows: A machine M and a word w are given. You have to tell, if M halts on w .

So, to solve halting problem $\langle M, w \rangle$ using L , just give $\langle M, w, 0 \rangle$ and $\langle M, w, 1 \rangle$ to two instances of T which is the assumed Turing machine for L . If T accepts the triplet $\langle M, w, 0 \rangle$, it means M halts on $w \Rightarrow$ we have solved halting problem. If T accepts the triplet $\langle M, w, 1 \rangle$, it means M doesn't halt on $w \Rightarrow$ we have solved halting problem. We know that either $\langle M, w, 0 \rangle$ or $\langle M, w, 1 \rangle$ is in L . So, if L is recursively enumerable, T is bound to stop on at least one of these inputs (TM for a recursively enumerable language stops and accepts, when provided with a word in its language).

Hence, if L is recursively enumerable we can solve halting problem $\Rightarrow L$ is not recursively enumerable.

Similarly, we can also show that halting problem can be solved with L' . (shown at end)

Hence, neither L nor L' is recursively enumerable.

To solve halting problem $\langle M, w \rangle$ using L' , just give $\langle M, w, 0 \rangle$ and $\langle M, w, 1 \rangle$ to two instances of T' which is the assumed Turing machine for L' . If T' accepts the triplet $\langle M, w, 0 \rangle$, it means M does not halt on $w \Rightarrow$ we have solved halting problem. If T' accepts the triplet $\langle M, w, 1 \rangle$, it means M halt on $w \Rightarrow$ we have solved halting problem. We know that either $\langle M, w, 0 \rangle$ or $\langle M, w, 1 \rangle$ is in L' . So, if L' is recursively enumerable, T' is bound to stop on at least one of these inputs (TM for a recursively enumerable language stops and accepts, when provided with a word in its language).

Hence, if L' is recursively enumerable we can solve halting problem $\Rightarrow L'$ is not recursively enumerable.

PS: If the bit part of the triplet is absent then L_0 is halting problem and L_1 is its complement and $L_0 \cup L_1 = \Sigma^*$, which is regular. Lets see how it happens.

Let the alphabet set be $\{0, 1\}$. Now for any string like 0010101 there are only two options- belong to L or belong to L' as this is what complement says. Now, lets take the case for L_0 and a string $001\dots10 - 01 - 1$, ("-" shown for notation purpose only) where the first component describes a TM M followed by input " $w = 01$ " and last bit "1". Now suppose M halts on "01". Still the given input is not in L_0 as the last bit is "1" and not "0" as required by L_0 . So, this input must be in L'_0 . But since M halts on w , this input is not in L_1 either. Similarly, we can get an infinite set of strings which does not belong to both L_0 and L_1 and this makes their union not Σ^* but an irregular (not r.e. as proved earlier) set. If the last bit is removed from the definition of L_0 and L_1 , then any string should be present in either L_0 or L_1 and their union would be Σ^* .

42 votes

-- gatecse (18k points)

5.17.6 Turing Machine: GATE2004-89 top

<https://gateoverflow.in/1083>



S_1 is TRUE.

If L_1 is recursive L_2 must also be recursive. Because to check if a word $w = w_i \# w_j$ belongs to L_2 , we can give w_i and w_j to the decider for L_1 and if both are accepted then w belongs to L_1 and not otherwise.

S_2 is TRUE.

With a decider for L_2 we can make a decider for L_1 as follows. Let w_1 be the first string enumerated by algorithm A for L_1 . Now, to check if a word w belongs to L_1 , make a string $w' = w_1 \# w$ and give it to the decider for L_2 and if accepted, then w belongs to L_1 and not otherwise.

So, answer must be **A**.

PS: For the second part, the given construction can fail if L_1 happens to be a finite language (more specifically if L_1 is empty). But all finite languages are anyway decidable.

23 votes

-- Arjun Suresh (350k points)

5.17.7 Turing Machine: GATE2014-2-35 top

<https://gateoverflow.in/1994>



There are only a finite number of strings of length 2014. So, we can give all those strings to TM simulating each string for 1 step, then 2 step and so on (dovetailing), and if the TM accepts any of them ("yes" case of TM), we can say "yes". So, L is recursively enumerable.

(If the TM doesn't accept any string of length 2014, it can go to an infinite loop ("no" case of TM), and hence we can't say the method is decidable).

Now, to prove whether the problem is decidable or not we can make use of Rice's theorem. Rice's theorem (I) states that any non-trivial property of $L(TM)$ is undecidable. $L(TM)$ has a string of length 2014 is a non-trivial property as there are TMs whose language contains such a string and there are TMs whose language doesn't have such a string. So, the given problem is undecidable.

http://gatcse.in/wiki/Rice%27s_Theorem_with_Examples

56 votes

-- Arjun Suresh (350k points)

5.17.8 Turing Machine: GATE2016-2-44 top

<https://gateoverflow.in/39596>



L_3 is not recursive as it asks if $L(M)$ contains ϵ which is a non-trivial property of r.e. languages and hence undecidable as per Rice's theorem.

L_1 and L_2 are slightly trickier as these are not describing properties of recursively enumerable languages, but rather of Turing machines. So, we can see if there is some procedure for deciding these.

For L_1 we can give the TM an input of length 2016. Now, it should at least make 2016 steps or reach the halt state before completing the input processing. The second case is possible only if the TM reaches a halt state before reaching the end of string (blank) of input, for all possible inputs of length at least 2016 and can be decided. So, we can be sure that otherwise TM will have at least

2016 steps making L_1 recursive.

L_2 is recursive and it is more easier to prove. For the complement of L_2 we need M to make less than 2016 steps for some input and we can just give it all possible inputs of length less than 2016 and see if it reaches a halt state within 2016 steps. Thus complement of L_2 is recursive $\implies L_2$ is recursive.

So, answer here is **C**.

44 54 votes

-- Arjun Suresh (350k points)

One more possible approach:

L_3 is not recursive (Can be proved using [Rice theorem](#)).

Lets talk about L_1 and L_2 , and let me take L_2 first

$L_2 = \{\langle M \rangle \mid M \text{ takes at least 2016 steps on all inputs}\},$

I want to check if for all strings in Σ^* M takes more than or equal to 2016 steps or not.

First of all I will restrict number of steps in M to 2016, and I will never run M more than 2016 steps. Because for any string, if M halts (accepts then halts or rejects then halts, does not matter) in less than 2016 steps then that string is not in L_2 . And if M does not halt within 2016 steps (after 2016 steps, I am not interested whether M is in infinite loop or will halt eventually) then string is in L_2 .

\implies Number of steps to be run in M is not more than 2016.

Since, we bound the number of steps that M runs on an input, then there is no point on looking at any strings that are longer

than that number, since if a TM is allowed to run for at most c steps, it is not possible for that TM to "process" any input symbol beyond the c^{th} symbol!

\implies Length of input string is less than 2016. (If I can decide for these strings then L_2 is Recursive otherwise not Recursive) And there are finite strings having length less than 2016.

Now my task reduces to : "Take each string in this finite set and run M for finite number of steps"

The number of possible inputs is finite, and the number of steps M runs on each input is finite, therefore M is guaranteed to halt and decide the language. Hence L_2 is recursive.

(If we can decide for all inputs then we can also decide for some inputs therefore L_1 is also recursive (Reduction), but we can think of L_1 as an independent problem)

$L_1 = \{\langle M \rangle \mid M \text{ takes at least 2016 steps on some input}\},$

I want to check if there exist any string in Σ^* for which M takes more than or equal to 2016 steps.

With same reasoning I can say that we will run M for finite number of steps and input string set is also finite. The only difference is, we can stop giving input once we find any string taking at least 2016 steps, whereas in L_2 we have to check for all set of input strings length less than 2016.

Therefore, L_1 is also recursive.

L_1, L_2 : Recursive

L_3 : Not Recursive.

C is answer.

Ref: Problem number one in this
pdf: https://www.cs.rice.edu/~nakhleh/COMP481/final_review_sp06_sol.pdf.

4 50 votes

-- Sachin Mittal (15.7k points)

Brought to you by GATE Overflow

<https://gateoverflow.in/>

