

DATA STRUCTURES

MCQ'S

SORTINGS:

What is recurrence for worst case of QuickSort and what is the time complexity in Worst case?

- ☐ A Recurrence is $T(n) = T(n-2) + O(n)$ and time complexity is $O(n^2)$
- ☒ B Recurrence is $T(n) = T(n-1) + O(n)$ and time complexity is $O(n^2)$
- ☐ C Recurrence is $T(n) = 2T(n/2) + O(n)$ and time complexity is $O(n \log n)$
- ☐ D Recurrence is $T(n) = T(n/10) + T(9n/10) + O(n)$ and time complexity is $O(n \log n)$

Which of the following is not a stable sorting algorithm in its typical implementation.

- ☐ A Insertion Sort
- ☐ B Merge Sort
- ☒ C Quick Sort
- ☐ D Bubble Sort

Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced).

- ☐ A Quick Sort
- ☐ B Heap Sort
- ☐ C Merge Sort
- ☒ D Insertion Sort

Consider a situation where swap operation is very costly. Which of the following sorting algorithms should be preferred so that the number of swap operations are minimized in general?

- ☐ A Heap Sort
- ☒ B Selection Sort
- ☐ C Insertion Sort
- ☐ D Merge Sort

Which sorting algorithms is most efficient to sort string consisting of ASCII characters?

- ☐ A Quick sort
- ☐ B Heap sort
- ☐ C Merge sort
- ☒ D Counting sort

Which sorting algorithm will take least time when all elements of input array are identical? Consider typical implementations of sorting algorithms.

- ☒ A Insertion Sort
- ☐ B Heap Sort
- ☐ C Merge Sort
- ☐ D Selection Sort

A sorting technique is called stable if

- ☐ A If it takes $O(n \log n)$ time
- ☐ B It uses divide and conquer technique
- ☒ C Relative order of occurrence of non-distinct elements is maintained
- ☐ D It takes $O(n)$ space

Sorting ISRO CS 2017

Which of the below given sorting techniques has highest best-case runtime complexity.

- ☐ A Quick sort
- ☒ B Selection sort
- ☐ C Insertion sort
- ☐ D Bubble sort

Sorting GATE 2017 Mock
Discuss it

Question 43 Explanation:

Quick sort best case time complexity is $O(n \log n)$
Selection sort best case time complexity is $O(n^2)$
Insertion sort best case time complexity is $O(n)$
Bubble sort best case time complexity is $O(n)$

Selection sort algorithm design technique is an example of



A Greedy method



B Divide-and-conquer



C Dynamic Programming



D Backtracking

Sorting ISRO CS 2007

- What is an external sorting algorithm == Algorithm that uses tape or disk during the sort
- What is an internal sorting algorithm? == Algorithm that uses main memory during the sort
- An insertion algorithm consists of $N-1$ passes when an array of N elements is given.
- Insertion sort is similar to that of a binary heap algorithm because of the use of temporary variable to swap.
- The total number of pairs in a list L is $N(N-1)/2$. Thus, an average list has half this amount, or $N(N-1)/4$ inversions.
- Which of the following real time examples is based on insertion sort?
arranging a pack of playing cards

10. In C, what are the basic loops required to perform an insertion sort?

- a) do- while
- b) if else
- c) for and while
- d) for and if

View Answer

Answer: c

Explanation: To perform an insertion sort, we use two basic loops- an outer for loop and an inner while loop.

- An insertion sort is stable, adaptive, in-place and incremental in nature.
- The worst case input for an insertion sort algorithm will be an array sorted in reverse order and its running time is quadratic
- Insertion sort is an example of an incremental algorithm.
- In Exchange sorts, we compare each element of an array and swap those elements, Insertion sort is not an exchange sort

1. What is an in-place sorting algorithm?

- a) It needs $O(1)$ or $O(\log n)$ memory to create auxiliary locations
- b) The input is already sorted and in-place
- c) It requires additional storage
- d) It requires additional space

View Answer

Answer: a

Explanation: Auxiliary memory is required for storing the data temporarily.

2. In the following scenarios, when will you use selection sort?

- a) The input is already sorted
- b) A large file has to be sorted
- c) Large values need to be sorted with small keys
- d) Small values need to be sorted with large keys

View Answer

Answer: c

Explanation: Selection is based on keys, hence a file with large values and small keys can be efficiently sorted with selection sort.

- What is the advantage of selection sort over other sorting techniques?
It requires no additional storage space
- What is the disadvantage of selection sort?
It is not scalable, as the input size increases, the performance of selection sort decreases.

-
- What is the auxiliary space complexity of merge sort?
 $O(n)$
 - Merge sort can be implemented using $O(1)$ auxiliary space.
true
 - Merge sort is not an adaptive sorting algorithm. This is because it takes $O(n \log n)$ time complexity irrespective of any case.

An additional space of $O(n)$ is required in order to merge two sorted arrays. Thus merge sort is not an in place sorting algorithm.

- Merge sort is a stable sorting algorithm.
- Merge sort is preferred for arrays over linked lists.
false

11. Which of the following is not a stable sorting algorithm?

- a) Quick sort
- b) Cocktail sort
- c) Bubble sort
- d) Merge sort

▼ View Answer

Answer: a

- Quick sort is the fastest known sorting algorithm because of its highly optimized inner loop.
- Insertion sort is used along with quick sort to sort the sub arrays. It is used only at the end.
- Choosing the first element as pivot is the worst method because if the input is pre-sorted or in reverse order, then the pivot provides a poor partition.
- The Quick sort is best suited to sort the array of 1 million elements. The practical implementations of Quick sort use randomised version.
- Quick sort is a space-optimised version of the binary tree sort.
- What is a randomized QuickSort? -- Any element in the array is chosen as the pivot
- Auxiliary space complexity of randomized quick sort is $O(\log n)$

8. A machine needs a minimum of 200 sec to sort 1000 elements by Quick sort. The minimum time needed to sort 200 elements will be approximately

- a) 60.2 sec
- b) 45.54 sec
- c) 31.11 sec
- d) 20 sec

▼ View Answer

Answer: c

Explanation: The Quick sort requires $n \log_2 n$ comparisons in best case, where n is size of input array. So, $1000 * \log_2 1000 \approx 9000$ comparisons are required to sort 1000 elements, which takes 200 sec. To sort 200 elements minimum of $200 * \log_2 200 \approx 1400$ comparisons are required. This will take $200 * 1400 / 9000 \approx 31.11$ sec.

Sorting Algorithms	In - Place	Stable
Bubble Sort	Yes	Yes
Selection Sort	Yes	No
Insertion Sort	Yes	Yes
Quick Sort	Yes	No
Merge Sort	No (because it requires an extra array to merge the sorted subarrays)	Yes
Heap Sort	Yes	No

Name	Average Case	Worst Case	Method	Advantage/Disadvantage
Bubble Sort	$O(n^2)$	$O(n^2)$	Exchange	1. Straightforward, simple and slow. 2. Stable. 3. Inefficient on large tables.
Insertion Sort	$O(n^2)$	$O(n^2)$	Insertion	1. Efficient for small list and mostly sorted list. 2. Sort big array slowly. 3. Save memory
Selection Sort	$O(n^2)$	$O(n^2)$	Selection	1. Improves the performance of bubble sort and also slow. 2. Unstable but can be implemented as a stable sort. 3. Quite slow for large amount of data.
Heap Sort	$O(n \log n)$	$O(n \log n)$	Selection	1. More efficient version of selection sort. 2. No need extra buffer. 3. Its does not require recursion. 4. Slower than Quick and Merge sorts.
Merge Sort	$O(n \log n)$	$O(n \log n)$	Merge	1. Well for very large list, stable sort. 2. A fast recursive sorting. 3. Both useful for internal and external sorting. 4. It requires an auxiliary array that is as large as the original array to be sorted.
In place-merge Sort	$O(n \log n)$	$O(n \log n)$	Merge	1. Unstable sort. 2. Slower than heap sort. 3. Needs only a constant amount of extra space in addition to that needed to store keys.

SEARCHING:

Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	$O(1)$	$O(n)$	$O(n)$	$O(1)$
Binary Search	$O(1)$	$O(\log n)$	$O(\log n)$	$O(1)$

The recurrence relation that arises in relation with the complexity of binary search is:

A

$T(n) = 2T(n/2) + k$, where k is constant

✓

$T(n) = T(n/2) + k$, where k is constant

C

$T(n) = T(n/2) + \log n$

D

$T(n) = T(n/2) + n$

Balanced Binary Search Trees Searching ISRO CS 2017 - May

- The Iterative algorithm is faster than the latter as recursive algorithm has overheads like calling function and registering stacks repeatedly.

7. Which of the following is not an application of binary search?

- a) To find the lower/upper bound in an ordered sequence
- b) Union of intervals
- c) Debugging
- d) To search in unordered list

View Answer

Answer: d

9. Binary Search can be categorized into which of the following?

- a) Brute Force technique
- b) Divide and conquer
- c) Greedy algorithm
- d) Dynamic programming

View Answer

Answer: b

- $T(n) = T(n/2) + \theta(1)$
Using the divide and conquer master theorem, we get the time complexity as $O(\log n)$

STACK & QUEUE:

Which one of the following is an application of Stack Data Structure?

- ☒ A Managing function calls
- ☐ B The stock span problem
- ☐ C Arithmetic expression evaluation
- ☒ D All of the above

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

Stack GATE-CS-2002

Any expression can be converted into Postfix or Prefix form. Prefix and postfix evaluation can be done using a single stack.

Suppose a stack is to be implemented with a linked list instead of an array. What would be the effect on the time complexity of the push and pop operations of the stack implemented using linked list (Assuming stack is implemented efficiently)?

- ☐ A $O(1)$ for insertion and $O(n)$ for deletion
- ☒ B $O(1)$ for insertion and $O(1)$ for deletion
- ☐ C $O(n)$ for insertion and $O(1)$ for deletion
- ☐ D $O(n)$ for insertion and $O(n)$ for deletion

Stack can be implemented using link list having $O(1)$ bounds for both insertion as well as deletion by inserting and deleting the element from the beginning of the list.

Consider n elements that are equally distributed in k stacks. In each stack, elements of it are arranged in ascending order (min is at the top in each of the stack and then increasing downwards). Given a queue of size n in which we have to put all n elements in increasing order. What will be the time complexity of the best known algorithm?

- ☒ A $O(n \log k)$

The minimum number of stacks needed to implement a queue is == 2

How many queues are needed to implement a stack. Consider the situation where no other data structure like arrays, linked list is available to you. == 2

- The best data structure to check whether an arithmetic expression has balanced parenthesis is a **STACK**
- What data structure would you mostly likely see in a non recursive implementation of a recursive algorithm? **STACK**
- Stacks are used for the implementation of Recursion.

Which of the following is not an inherent application of stack?

- ☐ **A** Implementation of recursion
- ☐ **B** Evaluation of a postfix expression
- ☒ **C** Job scheduling
- ☐ **D** Reverse a string

Which one of the following is an application of Queue Data Structure?

- ☐ **A** When a resource is shared among multiple consumers.
- ☐ **B** When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes
- ☐ **C** Load Balancing
- ☒ **D** All of the above

Which of the following is true about linked list implementation of queue?

- ☐ **A** In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
- ☐ **B** In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.
- ☒ **C** Both of the above
- ☐ **D** None of the above

- The data structure required for Breadth First Traversal on a graph is? QUEUE
- Circular Queue is also called as Ring Buffer. Circular Queue is a linear data structure in which last position is connected back to the first position to make a circle. It forms a ring structure.

6. A data structure in which elements can be inserted or deleted at/from both the ends but not in the middle is?

- a) Queue
- b) Circular queue
- c) Dequeue
- d) Priority queue

▼ View Answer

Answer: c

8. Queues serve major role in _____

- a) Simulation of recursion
- b) Simulation of arbitrary linked list
- c) Simulation of limited resource allocation
- d) Simulation of heap sort

▼ View Answer

Answer: c

9. Which of the following is not the type of queue?

- a) Ordinary queue
- b) Single ended queue
- c) Circular queue
- d) Priority queue

▼ View Answer

Answer: b

Explanation: Queue always has two ends. So, single ended queue is not the type of queue.

Linked List:

Which of the following points is/are true about Linked List data structure when it is compared with array

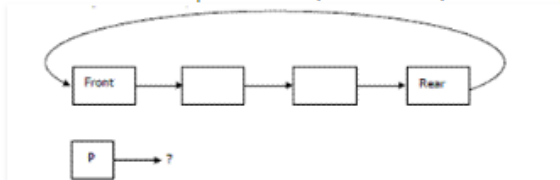
- ☐ A Arrays have better cache locality that can make them better in terms of performance.
- ☐ B It is easy to insert and delete elements in Linked List
- ☐ C Random access is not allowed in a typical implementation of Linked Lists
- ☐ D The size of array has to be pre-decided, linked lists can change their size any time.
- ☒ All of the above

Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

- ☐ A Insertion Sort
- ☐ B Quick Sort
- ☐ C Heap Sort
- ☒ Merge Sort

However Both Merge sort and Insertion sort can be used for linked lists.

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? (GATE 2004)

- ☒ rear node
- ☐ B front node
- ☐ C not possible with a single pointer
- ☐ D node next to front

In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is (GATE CS 2002)

- ☐ A $\log_2 n$
- ☐ B $n/2$
- ☐ C $\log_2 n - 1$
- ☒ n

The concatenation of two lists is to be performed in $O(1)$ 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 lists

The minimum number of fields with each node of doubly linked list is

- ☐ A 1
- ☐ B 2
- ☒ C 3
- ☐ D 4

1. What kind of linked list is best to answer question like "What is the item at position n?"

- a) Singly linked list
- b) Doubly linked list
- c) Circular linked list
- d) Array implementation of linked list

Answer: d

- Linked lists are not suitable to for the implementation of?

Binary Search

1. Which of the following is not a disadvantage to the usage of array?

- a) Fixed size
- b) There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
- c) Insertion based on position
- d) Accessing elements at specified positions

Answer: d

- What is the time complexity of inserting at the end in dynamic arrays?

Depending on whether the array is full or not, the complexity in dynamic array varies. If you try to insert into an array which is not full, then the element is simply stored at the end, this takes $O(1)$ time. If you try to insert into an array which is full, first you will have to allocate an array with double the size of the current array and then copy all the elements into it and finally insert the new element, this takes $O(n)$ time.

- To implement file system, for separate chaining in hash-tables and to implement non-binary trees linked lists are used. Elements are accessed sequentially in linked list. Random access of elements is not an applications of linked list.

1. Which of the following is false about a doubly linked list?

- a) We can navigate in both the directions
- b) It requires more space than a singly linked list
- c) The insertion and deletion of a node take a bit longer
- d) Implementing a doubly linked list is easier than singly linked list

▼ View Answer

Answer: d

6. What is the worst case time complexity of inserting a node in a doubly linked list?

- a) $O(n \log n)$
- b) $O(\log n)$
- c) $O(n)$
- d) $O(1)$

▼ View Answer

Answer: c

1. What differentiates a circular linked list from a normal linked list?

- a) You cannot have the 'next' pointer point to null in a circular linked list
- b) It is faster to traverse the circular linked list
- c) You may or may not have the 'next' pointer point to null in a circular linked list
- d) Head node is known in circular linked list

▼ View Answer

Answer: c

4. What is the time complexity of searching for an element in a circular linked list?

- a) $O(n)$
- b) $O(n \log n)$
- c) $O(1)$
- d) $O(n^2)$

▼ View Answer

Answer: a

5. Which of the following application makes use of a circular linked list?

- a) Undo operation in a text editor
- b) Recursive function calls
- c) Allocating CPU to resources
- d) Implement Hash Tables

▼ View Answer

Answer: c

Explanation: Generally, round robin fashion is employed to allocate CPU time to resources which makes use of the circular linked list data structure. Recursive function calls use stack data structure. Undo Operation in text editor uses doubly linked lists. Hash tables uses singly linked lists.

9. Which of the following is false about a circular linked list?

- a) Every node has a successor
- b) Time complexity of inserting a new node at the head of the list is $O(1)$
- c) Time complexity for deleting the last node is $O(n)$
- d) We can traverse the whole circular linked list by starting from any point

▼ View Answer

Answer: b

Explanation: Time complexity of inserting a new node at the head of the list is $O(n)$ because you have to traverse through the list to find the tail node.

Binary Tree:

A full binary tree (sometimes proper binary tree or 2-tree or strictly binary tree) is a tree in which every node other than the leaves has two children. A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled.

What are the main applications of tree data structure?

- 1) Manipulate hierarchical data
- 2) Make information easy to search (see tree traversal).
- 3) Manipulate sorted lists of data
- 4) Router algorithms
- 5) Form of a multi-stage decision-making, like Chess Game.
- 6) As a workflow for compositing digital images for visual effects

- The maximum number of binary trees that can be formed with three unlabeled nodes is:

Note that nodes are unlabeled. If the nodes are labeled, we get more number of trees. We can find the number of binary tree by Catalan number : Here $n = 3$ Number of binary tree = $(2nCn)/n+1 = (2*3C3)/4+1 = 5$.

- In Complete Binary tree : $2^{d+1} - 1$ where $2^d =$ leaf node and $2^d - 1$ Non leaf node
- A Strict Binary tree with n leaves, will have $2n-1$ node

- A binary tree T has 20 leaves. The number of nodes in T having two children is :

It is solve by : Handshaking Lemma and Interesting Tree Properties for proof.

$$(K-1)*I + 1 = L$$

Given $k = 2$, $L=20$

$$\implies (2-1)*I + 1 = 20$$

$$\implies I = 19$$

$\implies T$ has 19 internal nodes which are having two children.

- $2n-1$ when strictly binary tree

A strictly binary tree with 10 leaves

- ☐ A cannot have more than 19 nodes
- ☒ has exactly 19 nodes
- ☐ C has exactly 17 nodes
- ☐ D has exactly 20 nodes

Binary Trees ISRO CS 2017

What is the maximum height of any AVL tree with 7 nodes? Assume that height of tree with single node is 0.

- ☐ A 2
- ☒ 3
- ☐ C 4
- ☐ D 5

Binary Trees ISRO CS 2017

- A full binary tree with n non leaf nodes contain $2n+1$ nodes.

Suppose you are given a binary tree with n nodes, such that each node has exactly either zero or two children. The maximum height of the tree will be

- ☐ A $n / 2 - 1$
- ☐ B $n / 2 + 1$
- ☒ $(n - 1) / 2$
- ☐ D $(n + 1) / 2$

Binary Trees UGC NET CS 2016 July – II

The number of different binary trees with 6 nodes is _____.

- ☐ A 6
- ☐ B 42
- ☒ 132

From Equation = $(2nCn) / n+1$

A full binary tree with n leaves contains:

- ☐ A n nodes
- ☐ B $\log_2 n$ nodes
- ☒ $2n-1$
- ☐ X 2^n

Again Full binary tree = strictly binary tree

A complete binary tree with the property that the value at each node is as least as large as the values at its children is known as

- ☐ A binary search tree
- ☐ B AVL tree
- ☐ C completely balanced tree
- ☒ Heap

Binary Trees ISRO CS 2008

5. What is the average case time complexity for finding the height of the binary tree?

- a) $h = O(\log \log n)$
- b) $h = O(n \log n)$
- c) $h = O(n)$
- d) $h = O(\log n)$

Answer: d

7. In a full binary tree if number of internal nodes is I , then number of leaves L are?

- a) $L = 2 * I$
- b) $L = I + 1$
- c) $L = I - 1$
- d) $L = 2 * I - 1$

Answer: b

Explanation: Number of Leaf nodes in full binary tree is equal to $1 + \text{Number of Internal Nodes}$ i.e $L = I + 1$

8. In a full binary tree if number of internal nodes is I , then number of nodes N are?

- a) $N = 2 * I$
- b) $N = I + 1$
- c) $N = I - 1$
- d) $N = 2 * I + 1$

▼ View Answer

Answer: d

Explanation: Relation between number of internal nodes(I) and nodes(N) is $N = 2 * I + 1$.

1. Advantages of linked list representation of binary trees over arrays?

- a) dynamic size
- b) ease of insertion/deletion
- c) ease in randomly accessing a node
- d) both dynamic size and ease in insertion/deletion

▼ View Answer

Answer: d

4. Level order traversal of a tree is formed with the help of

- a) breadth first search
- b) depth first search
- c) dijkstra's algorithm
- d) prims algorithm

▼ View Answer

Answer: a

Explanation: Level order is similar to bfs.

- A binary tree is a rooted tree and also an ordered tree (i.e) every node in a binary tree has at most two children.
- Three common operations are performed in a binary tree- they are insertion, deletion and traversal.
- Two kinds of insertion operation is performed in a binary tree- inserting a leaf node and inserting an internal node.
- General ordered tree can be mapped into binary tree by representing them in a left-child-right-sibling way.
- The maximum number of nodes in a tree for which post-order and pre-order traversals may be equal is _____

Binary Search Tree:

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? (GATE CS 2011)

- ☐ A 0
- ☒ B 1
- ☐ C $n!$
- ☐ D $(1/(n+1)) \cdot 2n C n$

How many distinct binary search trees can be created out of 4 distinct keys?

- ☐ A 4
- ☒ B 14
- ☐ C 24
- ☐ D 42

By formula $2n C n / (n+1)$

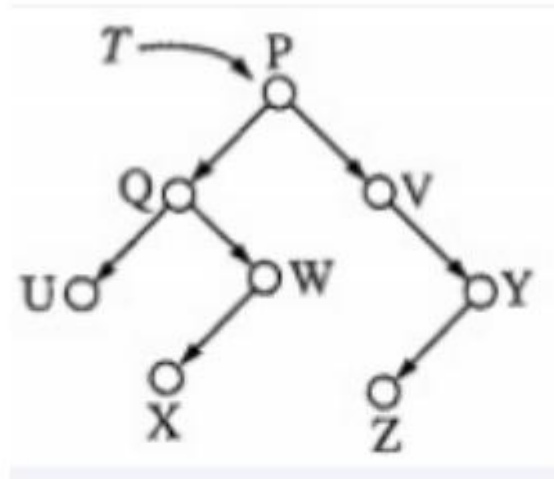
- When we know either preorder or postorder traversal, we can construct the BST. Note that we can always sort the given traversal and get the inorder traversal. Inorder traversal of BST is always sorted.

Access time of the symbolic table will be logarithmic if it is implemented by

- ☐ A Linear list
- ☒ B Search tree
- ☐ C Hash table
- ☐ D Self organization list

Binary Search Trees ISRO CS 2016

Consider the following binary search tree T given below: Which node contains the fourth smallest element in T?



- ☒ A Q
- ☐ B V
- ☒ C W
- ☐ D X

Binary Search Trees ISRO CS 2014

3. What is the speciality about the inorder traversal of a binary search tree?

- a) It traverses in a non increasing order
- b) It traverses in an increasing order
- c) It traverses in a random fashion
- d) It traverses based on priority of the node

[View Answer](#)

Answer: b

Explanation: As a binary search tree consists of elements lesser than the node to the left and the ones greater than the node to the right, an inorder traversal will give the elements in an increasing order.

Interface in Java:

Which of the following is true about interfaces in java.

- 1) An interface can contain following type of members.
....public, static, final fields (i.e., constants)
....default and static methods with bodies
- 2) An instance of interface can be created.
- 3) A class can implement multiple interfaces.
- 4) Many classes can implement the same interface.



1, 3 and 4

B

1, 2 and 4

C

2, 3 and 4

D

1, 2, 3 and 4

We can make a class abstract by

A

Declaring it abstract using the virtual keyword

B

Making at least one member function as virtual function



Making at least one member function as pure virtual function

D

Making all member function const

Java Abstract Class and Interface ISRO CS 2008

Q) A java interface can contain _____

- A. public static Final Variables only
- B. public Abstract methods
- C. Abstract methods(unimplemented) and implemented methods both
- D. public static Final Variables and abstract methods both

^ View Answer

Answer: D

An interface can have both final variables and abstract methods.

Q) which of the following is true about methods in an interface in java?

- A. An interface can contain only abstract method.
- B. We can define a method in an interface
- C. Private and protected access modifiers can also be used to declare methods in interface
- D. None

^ View Answer

Answer: A

In java, an interface contains only abstract method that can be public and it does not have any method implementation.

Q) Can we declare an interface as final?

- A. YES
- B. NO

^ View Answer

Answer: B

Explanation:

No, we cannot declare an interface as final. In *Java final keyword is used to stop extension* or inheritance by sub classes. But interface is meant to be used in inheritance. Hence, we can't, declare an interface as final because if we declare final no use of that Interface. That's why final is illegal for Interfaces.

Q) The fields in an interface are implicitly specified as

- A. Static
- B. Protected
- C. Private
- D. Static and final

^ View Answer

Answer: d

Fields in an interface are by default, static and final.

Q) Which is correct option about java interface?

- A. Interface is used to achieve multiple inheritance in java
- B. Object of an interface cannot be created.
- C. An interface can extend another interface.
- D. All of the above

^ View Answer

Answer: D

Object of an interface cannot be created. Java does not support multiple inheritance using classes, but, you can achieve multiple inheritance using interfaces in java. Read [how to achieve multiple inheritance in java using interfaces](#). A interface can also extends another interface or multiple interfaces in java programming.

8. What happens when a constructor is defined for an interface?

- a) Compilation failure
- b) Runtime Exception
- c) The interface compiles successfully
- d) The implementing class will throw exception

▼ View Answer

Answer: a

Explanation: Constructor is not provided by interface as objects cannot be instantiated.

Generics:

6. Which of these Exception handlers cannot be type parameterized?

- a) catch
- b) throw
- c) throws
- d) all of the mentioned

▼ View Answer

Answer: d

Explanation: we cannot Create, Catch, or Throw Objects of Parameterized Types as generic class cannot extend the Throwable class directly or indirectly.

7. Which of the following cannot be Type parameterized?

- a) Overloaded Methods
- b) Generic methods
- c) Class methods
- d) Overriding methods

▼ View Answer

Answer: a

Explanation: Cannot Overload a Method Where the Formal Parameter Types of Each Overload Erase to the Same Raw Type.

2. Which of these type parameters is used for a generic methods to return and accept any type of object?

- a) K
- b) N
- c) T
- d) V

▼ View Answer

Answer: c

Explanation: T is used for type, A type variable can be any non-primitive type you specify: any class type, any interface type, any array type, or even another type variable.

3. Which of these type parameters is used for a generic methods to return and accept a number?

- a) K
- b) N
- c) T
- d) V

▼ View Answer

Answer: b

Explanation: N is used for Number.

5. Which of the following allows us to call generic methods as a normal method?

- a) Type Interface
- b) Interface
- c) Inner class
- d) All of the mentioned

▼ View Answer

Answer: a

Explanation: Type inference, allows you to invoke a generic method as an ordinary method, without specifying a type between angle brackets.

1. Which of these types cannot be used to initiate a generic type?

- a) Integer class
- b) Float class
- c) Primitive Types
- d) Collections

▼ View Answer

Answer: c

Explanation: None.

2. Which of these instance cannot be created?

- a) Integer instance
- b) Generic class instance
- c) Generic type instance
- d) Collection instances

▼ View Answer

Answer: c

Explanation: It is not possible to create generic type instances. Example - "E obj = new E();" will give a compilation error.

3. Which of these data type cannot be type parameterized?

- a) Array
- b) List
- c) Map
- d) Set

▼ View Answer

Answer: a

Explanation: None.

HashMap:

1. Map implements collection interface?

- a) True
- b) False

▼ View Answer

Answer: b

Explanation: Collection interface provides add, remove, search or iterate while map has clear, get, put, remove, etc.

4. What happens if we put a key object in a HashMap which exists?

- a) The new object replaces the older object
- b) The new object is discarded
- c) The old object is removed from the map
- d) It throws an exception as the key already exists in the map

▼ View Answer

Answer: a

Explanation: HashMap always contains unique keys. If same key is inserted again, the new object replaces the previous object.

5. While finding the correct location for saving key value pair, how many times the key is hashed?

- a) 1
- b) 2
- c) 3
- d) unlimited till bucket is found

▼ View Answer

Answer: b

Explanation: The key is hashed twice; first by hashCode() of Object class and then by internal hashing method of HashMap class.

1. Which of these class object uses the key to store value?

- a) Dictionary
- b) Map
- c) Hashtable
- d) All of the mentioned

▼ View Answer

Answer: d

7. What is the load factor?

- a) Average array size
- b) Average key size
- c) Average chain length
- d) Average hash table length

▼ View Answer

Answer: c

Explanation: In simple chaining, load factor is the average number of elements stored in a chain, and is given by the ratio of number of elements stored to the number of slots in the array.

2. If several elements are competing for the same bucket in the hash table, what is it called?

- a) Diffusion
- b) Replication
- c) Collision
- d) Duplication

▼ View Answer

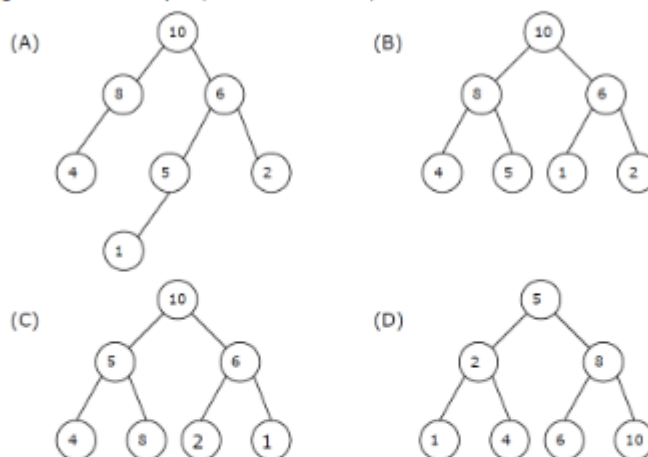
Answer: c

Heap:

What is the time complexity of Build Heap operation. Build Heap is used to build a max(or min) binary heap from a given array. Build Heap is used in Heap Sort as a first step for sorting.

- ☐ A $O(n \log n)$
- ☐ B $O(n^2)$
- ☐ C $O(\log n)$
- ☒ D $O(n)$

A max-heap is a heap where the value of each parent is greater than or equal to the values of its children. Which of the following is a max-heap? (GATE CS 2011)



- ☐ A
- ☒ B
- ☐ C
- ☐ D

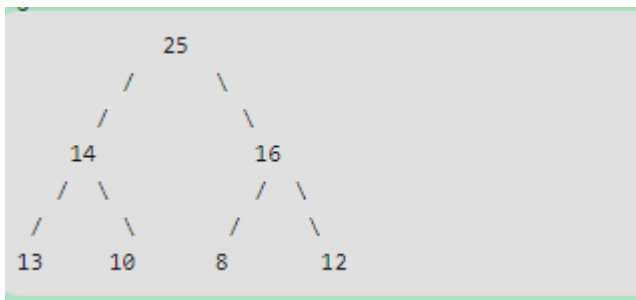
The Important Note is here that Option A also follows the rule of Max-heap parent Node but for a Heap It must be a Complete Binary Tree.

Consider a binary max-heap implemented using an array. Which one of the following array represents a binary max-heap? (GATE CS 2009)

- ☐ A 25,12,16,13,10,8,14
- ☐ B 25,12,16,13,10,8,14
- ☒ C 25,14,16,13,10,8,12
- ☐ D 25,14,12,13,10,8,16

Always fill heap tree element from left to right and take care of complete binary tree with Max, Min Heap tree.

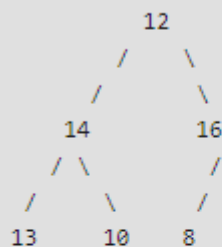
Like:



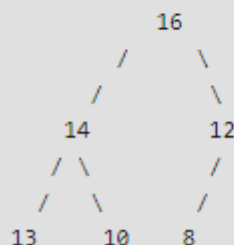
What is the content of the array after two delete operations on the correct answer to the previous question?

- A** 14,13,12,10,8
- B** 14,12,13,8,10
- C** 14,13,8,12,10
- ✓ 14,13,12,8,10

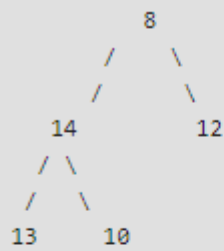
For Heap trees, deletion of a node includes following two operations. 1) Replace the root with last element on the last level. 2) Starting from root, heapify the complete tree from top to bottom.. Let us delete the two nodes one by one: 1) Deletion of 25: Replace 25 with 12



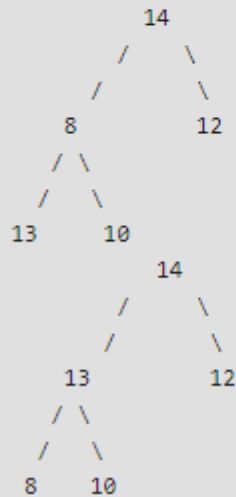
Since heap property is violated for root (16 is greater than 12), make 16 as root of the tree.



2) Deletion of 16: Replace 16 with 8



Heapify from root to bottom.



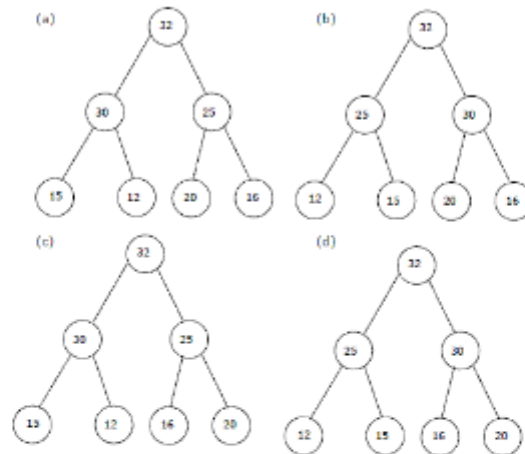
We have a binary heap on n elements and wish to insert n more elements (not necessarily one after another) into this heap. The total time required for this is (A) $\theta(\log n)$ (B) $\theta(n)$ (C) $\theta(n \log n)$ (D) $\theta(n^2)$

- ☒ A
- ☒ B
- ☐ C
- ☐ D

In a binary max heap containing n numbers, the smallest element can be found in time (GATE CS 2006)

- ☒ $O(n)$
- ☐ $O(\log n)$
- ☐ $O(\log \log n)$
- ☐ $O(1)$

The elements 32, 15, 20, 30, 12, 25, 16 are inserted one by one in the given order into a Max Heap. The resultant Max Heap is.



- ☒ a
- ☐ b
- ☐ c
- ☐ d

In this question after inserting the element as complete binary tree. Heap tree needs to be upheapified to make a Max heap tree.

Which languages necessarily need heap allocation in the run time environment?

- ☐ A Those that support recursion
- ☐ B Those that use dynamic scoping
- ☐ C Those that use global variables
- ☒ D Those that allow dynamic data structures

C Dynamic Memory Allocation **Heap** **ISRO CS 2017**

3. What is the complexity of adding an element to the heap.

- a) $O(\log n)$
- b) $O(h)$
- c) $O(\log n)$ & $O(h)$
- d) $O(n)$

[View Answer](#)

Answer: c

4. The worst case complexity of deleting any arbitrary node value element from heap is

- a) $O(\log n)$
- b) $O(n)$
- c) $O(n \log n)$
- d) $O(n^2)$

▼ View Answer

Answer: a

5. Heap can be used as _____

- a) Priority queue
- b) Stack
- c) A decreasing order array
- d) Normal Array

▼ View Answer

Answer: a

1. What is the space complexity of searching in a heap?

- a) $O(\log n)$
- b) $O(n)$
- c) $O(1)$
- d) $O(n \log n)$

▼ View Answer

Answer: b

1. Descending priority queue can be implemented using

- a) max heap
- b) min heap
- c) min-max heap
- d) trie

▼ View Answer

Answer: a

2. Min heap can be used to implement selection sort.

- a) True
- b) False

▼ View Answer

Answer: a

Graph:

The degree sequence of a simple graph is the sequence of the degrees of the nodes in the graph in decreasing order. Which of the following sequences can not be the degree sequence of any graph?

- I. 7, 6, 5, 4, 4, 3, 2, 1
- II. 6, 6, 6, 6, 3, 3, 2, 2
- III. 7, 6, 6, 4, 4, 3, 2, 2
- IV. 8, 7, 7, 6, 4, 2, 1, 1

- ☐ A I and II
- ☐ B III and IV
- ☐ C IV only
- ☒ D II and IV

Use Havel hakimi theorem starts from $n-1$

Which of the following statements is/are TRUE for an undirected graph? P: Number of odd degree vertices is even Q: Sum of degrees of all vertices is even

- ☐ A P Only
- ☐ B Q Only
- ☒ C Both P and Q
- ☐ D Neither P nor Q

Graph
Discuss it

Question 7 Explanation:

P is true for undirected graph as adding an edge always increases degree of two vertices by 1.
Q is true: If we consider sum of degrees and subtract all even degrees, we get an even number because every edge increases the sum of degrees by 2. So total number of odd degree vertices must be even.

Given an undirected graph G with V vertices and E edges, the sum of the degrees of all vertices is

- ☒ A E
- ☒ B $2E$
- ☐ C V
- ☐ D $2V$

Graph
Discuss it

Question 9 Explanation:

Since the given graph is undirected, every edge contributes as 2 to sum of degrees. So the sum of degrees is $2E$.

What is the maximum number of edges in an acyclic undirected graph with n vertices?

- ☒ A $n-1$
- ☐ B n
- ☐ C $n + 1$
- ☐ D $2n-1$

Graph GATE-IT-2004

$n * (n - 1) / 2$ when cyclic. But acyclic graph with the maximum number of edges is actually a spanning tree and therefore, correct answer is $n-1$ edges.

What is the largest integer m such that every simple connected graph with n vertices and n edges contains at least m different spanning trees?

- ☐ A 1
- ☐ B 2
- ☒ C 3
- ☐ D n

Graph Graph Minimum Spanning Tree Gate IT 2007

Which of the following data structure is useful in traversing a given graph by breadth first search?

- ☐ A Stack
- ☐ B List
- ☒ C Queue
- ☐ D None of the above.

Graph **Graph Shortest Paths** **ISRO CS 2017 - May**

Trie is also known as _____

- ☐ A Treap
- ☐ B Binomial Tree
- ☐ C 2-3 Tree
- ☒ D Digital Tree

Which kind of traversal order trie gives the lexicographical sorting of the set of the strings?

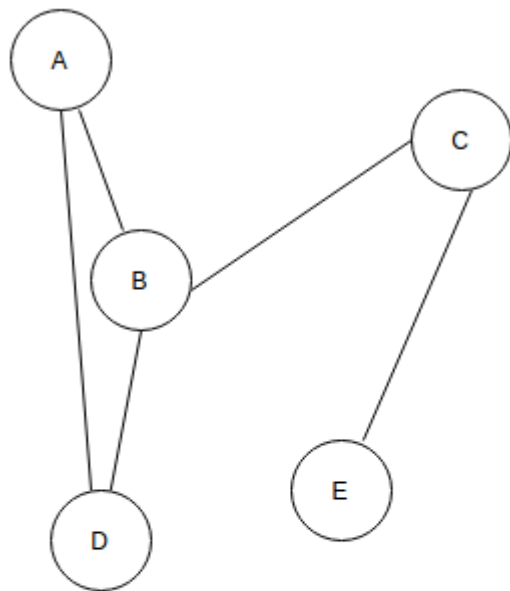
- ☒ A Preorder
- ☐ B Postorder
- ☐ C Level order
- ☐ D Inorder

Graph
Discuss it

Question 30 Explanation:

To print the string in alphabetical order we have to first insert in the trie and then perform preorder traversal to print in alphabetical order.

2. In the given graph identify the cut vertices.



a) B and E

b) C and D

c) A and E

d) C and B

View Answer

Answer: d

Explanation: After removing either B or C, the graph becomes disconnected.

14. A graph with all vertices having equal degree is known as a _____

a) Multi Graph

b) Regular Graph

c) Simple Graph

d) Complete Graph

View Answer

Answer: b

Explanation: The given statement is the definition of regular graphs.

1. The number of elements in the adjacency matrix of a graph having 7 vertices is _____

a) 7

b) 14

c) 36

d) 49

View Answer

Answer: d

Explanation: There are $n \times n$ elements in the adjacency matrix of a graph with n vertices.

3. Adjacency matrix of all graphs are symmetric.

a) False

b) True

▼ View Answer

Answer: a

Explanation: Only undirected graphs produce symmetric adjacency matrices.

2. Given a plane graph, G having 2 connected component, having 6 vertices, 7 edges and 4 regions. What will be the number of connected components?

a) 1

b) 2

c) 3

d) 4

▼ View Answer

Answer: b

Explanation: Euler's Identity says $V - E + R = 1 + \text{number of connected components}$.

1. Dijkstra's Algorithm will work for both negative and positive weights?

a) True

b) False

▼ View Answer

Answer: b

Explanation: Dijkstra's Algorithm assumes all weights to be non-negative.

2. A graph having an edge from each vertex to every other vertex is called a _____

a) Tightly Connected

b) Strongly Connected

c) Weakly Connected

d) Loosely Connected

▼ View Answer

Answer: a

Explanation: This is a part of the nomenclature followed in Graph Theory.

1. Every Directed Acyclic Graph has at least one sink vertex.

a) True

b) False

▼ View Answer

Answer: a

Explanation: A sink vertex is a vertex which has an outgoing degree of zero.

4. The topological sorting of any DAG can be done in _____ time.

a) cubic

b) quadratic

c) linear

d) logarithmic

▼ View Answer

Object Oriented Programming in Java:

Which of the following is associated with objects?

- ☐ A State
- ☐ B Behaviour
- ☐ C Identity
- ☒ All of the above

OOP Concepts ISRO CS 2017 - May

The feature in object-oriented programming that allows the same operation to be carried out differently, depending on the object, is:

- ☐ A Inheritance
- ☒ Polymorphism
- ☐ C Overfunctioning
- ☐ D Overriding

Converting a primitive type data into its corresponding wrapper class object instance is called

- ☐ A Boxing
- ☐ B Wrapping
- ☐ C Instantiation
- ☒ Autoboxing

3. When does method overloading is determined?

- a) At run time
- b) At compile time
- c) At coding time
- d) At execution time

[View Answer](#)

Answer: b

Explanation: Overloading is determined at compile time. Hence, it is also known as compile time polymorphism.

5. Which concept of Java is a way of converting real world objects in terms of class?

- a) Polymorphism
- b) Encapsulation
- c) Abstraction
- d) Inheritance

▼ View Answer

Answer: c

Explanation: Abstraction is the concept of defining real world objects in terms of classes or interfaces.

6. Which concept of Java is achieved by combining methods and attribute into a class?

- a) Encapsulation
- b) Inheritance
- c) Polymorphism
- d) Abstraction

▼ View Answer

Answer: a

Explanation: Encapsulation is implemented by combining methods and attribute into a class. The class acts like a container of encapsulating properties.

8. What is it called where child object gets killed if parent object is killed?

- a) Aggregation
- b) Composition
- c) Encapsulation
- d) Association

▼ View Answer

Answer: b

10. Method overriding is combination of inheritance and polymorphism?

- a) True
- b) false

▼ View Answer

Answer: a

6. Abstract class cannot have a constructor.

- a) True
- b) False

▼ View Answer

Answer: b

Explanation: No instance can be created of abstract class. Only pointer can hold instance of object.

7. What is true about protected constructor?

- a) Protected constructor can be called directly
- b) Protected constructor can only be called using super()
- c) Protected constructor can be used outside package
- d) protected constructor can be instantiated even if child is in a different package

▼ View Answer

Answer: b

2. Which of these can be overloaded?

- a) Methods
- b) Constructors
- c) All of the mentioned
- d) None of the mentioned

▼ View Answer

Answer: c

1. Which of this keyword can be used in a subclass to call the constructor of superclass?

- a) super
- b) this
- c) extent
- d) extends

▼ View Answer

Answer: a

3. Which of these keywords can be used to prevent Method overriding?

- a) static
- b) constant
- c) protected
- d) final

▼ View Answer

Answer: d

4. Which of these is correct way of calling a constructor having no parameters, of superclass A by subclass B?

- a) super(void);
- b) superclass.();
- c) super.A();
- d) super();

▼ View Answer

Answer: d

1. Which of these class is superclass of every class in Java?

- a) String class
- b) Object class
- c) Abstract class
- d) ArrayList class

▼ View Answer

Answer: b

2. Which of these is not abstract?

- a) Thread
- b) AbstractList
- c) List
- d) None of the Mentioned

▼ View Answer

Answer: a

5. Which of these packages contains abstract keyword?

- a) java.lang
- b) java.util
- c) java.io
- d) java.system

▼ View Answer

Answer: a

10. Does Java support multiple level inheritance?

- a) True
- b) False

▼ View Answer

Answer: a

Explanation: Java supports multiple level inheritance through implementing multiple interfaces.