

Explore More

Subscription : Premium CDAC NOTES & MATERIAL @99



Contact to Join
Premium Group



Click to Join
Telegram Group

<CODEWITHARRAY'S/>

For More E-Notes

Join Our Community to stay Updated

TAP ON THE ICONS TO JOIN!

	codewitharrays.in freelance project available to buy contact on 8007592194	
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySql
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySql
3	Tour and Travel management System	React+Springboot+MySql
4	Election commition of India (online Voting System)	React+Springboot+MySql
5	HomeRental Booking System	React+Springboot+MySql
6	Event Management System	React+Springboot+MySql
7	Hotel Management System	React+Springboot+MySql
8	Agriculture web Project	React+Springboot+MySql
9	AirLine Reservation System / Flight booking System	React+Springboot+MySql
10	E-commerce web Project	React+Springboot+MySql
11	Hospital Management System	React+Springboot+MySql
12	E-RTO Driving licence portal	React+Springboot+MySql
13	Transpotation Services portal	React+Springboot+MySql
14	Courier Services Portal / Courier Management System	React+Springboot+MySql
15	Online Food Delivery Portal	React+Springboot+MySql
16	Muncipal Corporation Management	React+Springboot+MySql
17	Gym Management System	React+Springboot+MySql
18	Bike/Car ental System Portal	React+Springboot+MySql
19	CharityDonation web project	React+Springboot+MySql
20	Movie Booking System	React+Springboot+MySql

freelance_Project available to buy contact on 8007592194		
21	Job Portal web project	React+Springboot+MySql
22	LIC Insurance Portal	React+Springboot+MySql
23	Employee Management System	React+Springboot+MySql
24	Payroll Management System	React+Springboot+MySql
25	RealEstate Property Project	React+Springboot+MySql
26	Marriage Hall Booking Project	React+Springboot+MySql
27	Online Student Management portal	React+Springboot+MySql
28	Resturant management System	React+Springboot+MySql
29	Solar Management Project	React+Springboot+MySql
30	OneStepService LinkLabourContractor	React+Springboot+MySql
31	Vehical Service Center Portal	React+Springboot+MySql
32	E-wallet Banking Project	React+Springboot+MySql
33	Blogg Application Project	React+Springboot+MySql
34	Car Parking booking Project	React+Springboot+MySql
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySql
36	Society management Portal	React+Springboot+MySql
37	E-College Portal	React+Springboot+MySql
38	FoodWaste Management Donate System	React+Springboot+MySql
39	Sports Ground Booking	React+Springboot+MySql
40	BloodBank mangement System	React+Springboot+MySql

41	Bus Tickit Booking Project	React+Springboot+MySql
42	Fruite Delivery Project	React+Springboot+MySql
43	Woodworks Bed Shop	React+Springboot+MySql
44	Online Dairy Product sell Project	React+Springboot+MySql
45	Online E-Pharma medicine sell Project	React+Springboot+MySql
46	FarmerMarketplace Web Project	React+Springboot+MySql
47	Online Cloth Store Project	React+Springboot+MySql
48	Train Ticket Booking Project	React+Springboot+MySql
49	Quizz Application Project	JSP+Springboot+MySql
50	Hotel Room Booking Project	React+Springboot+MySql
51	Online Crime Reporting Portal Project	React+Springboot+MySql
52	Online Child Adoption Portal Project	React+Springboot+MySql
53	online Pizza Delivery System Project	React+Springboot+MySql
54	Online Social Complaint Portal Project	React+Springboot+MySql
55	Electric Vehical management system Project	React+Springboot+MySql
56	Online mess / Tiffin management System Project	React+Springboot+MySql
57		React+Springboot+MySql
58		React+Springboot+MySql
59		React+Springboot+MySql
60		React+Springboot+MySql

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9clHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/VXz0kZQi5to?si=ILOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FlzsAmIBCbk?si=7ujQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/lynlouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyl4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKZ

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrkg7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizlyJlh
15	Employee management system project	https://youtu.be/ID1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSIsm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWAOzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vN0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

1. _____ is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output.

- A. Code
- B. Algorithm
- C. Program
- D. All of the above

[View Answer](#)

Ans : B
Explanation: Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output.

2. Which of the following are characteristics of an algorithm?

- A. Algorithm should be clear
- B. Algorithm should be unambiguous
- C. Algorithms must terminate after a finite number of steps
- D. All of the above

[View Answer](#)

Ans : D
Explanation: All of the above are Characteristics of an Algorithm

3. Not all procedures can be called an algorithm

- A. TRUE
- B. FALSE
- C. Can be true or false
- D. Can not say

[View Answer](#)

Ans : A
Explanation: True, Not all procedures can be called an algorithm.

4. An algorithm should have _____ well-defined inputs.

- A. 0
- B. 1
- C. 0 or more
- D. 1 or more

[View Answer](#)

Ans : C
Explanation: An algorithm should have 0 or more well-defined inputs.

5. An algorithm should have _____ well-defined outputs

- A. 0
- B. 1
- C. 0 or more
- D. 1 or more

[View Answer](#)

Ans : D

Explanation: An algorithm should have 1 or more well-defined outputs, and should match the desired output.

6. Which of the following is a theoretical analysis of an algorithm?

- A. Posterior Analysis
- B. Priori Analysis
- C. Simple Analysis
- D. Preori Analysis

[View Answer](#)

Ans : B

Explanation: A Priori Analysis : This is a theoretical analysis of an algorithm. Efficiency of an algorithm is measured by assuming that all other factors, for example, processor speed, are constant and have no effect on the implementation.

7. _____ is measured by counting the number of key operations such as comparisons in the sorting algorithm.

- A. Space
- B. Lines
- C. Time
- D. None of the above

[View Answer](#)

Ans : C

Explanation: Time is measured by counting the number of key operations such as comparisons in the sorting algorithm.

8. The complexity of an algorithm $f(n)$ gives the running time and/or the storage space required by the algorithm in terms of n as the size of input data.

- A. Yes
- B. No
- C. Can be yes or no
- D. Can not say

[View Answer](#)

Ans : A

Explanation: Yes, The complexity of an algorithm $f(n)$ gives the running time and/or the storage space required by the algorithm in terms of n as the size of input data.

9. The space required by an algorithm is equal to the sum of _____ components.

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

[View Answer](#)

Ans : B

Explanation: The space required by an algorithm is equal to the sum of the following two components : fixed part and variable part.

10. Space complexity $S(P)$ of any algorithm P is $S(P) = C + SP(I)$, where C is the?

- A. fixed part
- B. variable part
- C. space complexity
- D. None of the above

[View Answer](#)

Ans : A

Explanation: Space complexity $S(P)$ of any algorithm P is $S(P) = C + SP(I)$, where C is the fixed part and $S(I)$ is the variable part of the algorithm, which depends on instance characteristic I .

codewitharrays.in 8007592194

1. 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

[View Answer](#)

Ans : A

Explanation: An array of size 50 looks the best option to store number of students for each score. We need to store frequencies of scores above 50. We can ignore scores below 50 and to index the scores above 50, we can subtract 50 from the score value.

2. Which of these best describes an array?

- A. A data structure that shows a hierarchical behavior
- B. Container of objects of similar types
- C. Container of objects of mixed types
- D. All of the mentioned

[View Answer](#)

Ans : B

Explanation: Array contains elements only of the same type.

3. Let A be a square matrix of size $n \times n$. Consider the following program. 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 matrix A
- C. Adding 100 to the upper diagonal elements & subtracting 100 from diagonal elements of A
- D. None of the above

[View Answer](#)

Ans : A

Explanation: If we take look at the inner statements of first loops, we can notice that the statements swap $A[i][j]$ and $A[j][i]$ for all i and j . Since the loop runs for all elements, every element $A[i][m]$ would be swapped twice, once for $i = l$ and $j = m$ and then for $i = m$ and $j = l$. Swapping twice means the matrix doesn't change.

4. What is the output of the following piece of code?

```
public class array
{
public static void main(String args[])
{
int []arr = {1,2,3,4,5};
System.out.println(arr[2]);
System.out.println(arr[4]);
}
}
```

- A. 3 and 5
- B. 5 and 3
- C. 2 and 4
- D. 4 and 2

[View Answer](#)

Ans : A

Explanation: Array indexing starts from 0.

5. Consider an array A[20, 10], assume 4 words per memory cell and the base address of array A is 100. What is the address of A[11, 5] ? Assume row major storage.

- A. 560
- B. 565
- C. 570
- D. 575

[View Answer](#)

Ans : A

Explanation: No Explanation.

6. What is the output of the following piece of code?

```
public class array
{
public static void main(String args[])
{
int []arr = {1,2,3,4,5};
System.out.println(arr[5]);
}
}
```

- A. 4
- B. 5
- C. ArrayIndexOutOfBoundsException
- D. InavlidInputException

[View Answer](#)

Ans : C

Explanation: Trying to access an element beyond the limits of an array gives ArrayIndexOutOfBoundsException.

7. Which of the following is an illegal array definition?

- A. Type COLONGE : (LIME, PINE, MUSK, MENTHOL); var a : array [COLONGE] of REAL;
B. var a : array [REAL] of REAL;
C. var a : array ['A'...'Z'] of REAL;
D. var a : array [BOOLEAN] of REAL;

View Answer

Ans : B

Explanation:No explanation.

8. Which of the following concepts make extensive use of arrays?

- A. Binary trees
B. Scheduling of processes
C. Caching
D. Spatial locality

View Answer

Ans : D

Explanation:Whenever a particular memory location is referred, it is likely that the locations nearby are also referred, arrays are stored as contiguous blocks in memory, so if you want to access array elements, spatial locality makes it to access quickly.

9. Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A . What is the expected number of inversions in any permutation on n elements?

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

View Answer

Ans : D

Explanation:The expected number of inversions in any permutation on n elements is $\theta(n^2)$.

10. Which of the following operations is not $O(1)$ for an array of sorted data. You may assume that array elements are distinct.

- A. Find the i th largest element
B. Delete an element
C. Find the i th smallest element
D. All of the above

View Answer

Ans : B

Explanation:The worst case time complexity for deleting an element from array can become $O(n)$.

11. The smallest element of an array's index is called its

- A. lower bound.
B. upper bound.
C. range.
D. extraction.

View Answer

Ans : A

12. The extra key inserted at the end of the array is called a,

- A. End key.
- B. Stop key.
- C. Sentinel.
- D. Transposition.

View Answer

Ans : C
Explanation: None

13. The largest element of an array index is called its

- A. lower bound.
- B. range.
- C. upper bound.
- D. All of these.

View Answer

Ans : C
Explanation:None

14. Each array declaration need not give, implicitly or explicitly, the information about

- A. the name of array
- B. the data type of array
- C. the first data from the set to be stored
- D. the index set of the array

View Answer

Ans : C
Explanation:None

15. The elements of an array are stored successively in memory cells because

- A. by this way computer can keep track only the address of the first element and the addresses of other elements can be calculated
- B. the architecture of computer memory does not allow arrays to store other than serially
- C. both of above
- D. none of above

View Answer

Ans : A
Explanation:No Explanation

1. _____ of an algorithm refers to defining the mathematical boundation/framing of its run-time performance.

- A. Symptotic analysis
- B. Asymptotic analysis
- C. Posterior Analysis
- D. Priori Analysis

[View Answer](#)

Ans : B

Explanation: Asymptotic analysis of an algorithm refers to defining the mathematical boundation/framing of its run-time performance.

2. Using asymptotic analysis, we can very well conclude the _____ scenario of an algorithm.

- A. best case
- B. average case
- C. worst case
- D. best case, average case, and worst case

[View Answer](#)

Ans : D

Explanation: Using asymptotic analysis, we can very well conclude the best case, average case, and worst case scenario of an algorithm.

3. Which case indicate the minimum time required for program execution?

- A. best case
- B. average case
- C. worst case
- D. None of the above

[View Answer](#)

Ans : A

Explanation: Best Case : Minimum time required for program execution.

4. _____ is the formal way to express the upper bound of an algorithm's running time.

- A. Omega Notation
- B. Theta Notation
- C. Big Oh Notation
- D. All of the above

[View Answer](#)

Ans : C

Explanation: The notation $O(n)$ is the formal way to express the upper bound of an algorithm's running time. It measures the worst case time complexity or the longest amount of time an algorithm can possibly take to complete.

5. Worst Case indicates maximum time required for program execution.

- A. Yes
- B. No

- C. Can be yes or no
- D. Can not say

[View Answer](#)

Ans : A

Explanation: Yes, Worst Case indicate maximum time required for program execution

6. Which of the following is linear asymptotic notations?

- A. $O(1)$
- B. $O(\log n)$
- C. $O(n)$
- D. $O(n \log n)$

[View Answer](#)

Ans : C

Explanation: linear : $O(n)$

7. $O(\log n)$ is?

- A. constant asymptotic notations
- B. logarithmic asymptotic notations
- C. polynomial asymptotic notations
- D. quadratic asymptotic notations

[View Answer](#)

Ans : B

Explanation: logarithmic : $O(\log n)$

8. Omega Notation is the formal way to express the lower bound of an algorithm's running time.

- A. TRUE
- B. FALSE
- C. Can be true or false
- D. Can not say

[View Answer](#)

Ans : A

Explanation: True, Omega Notation is the formal way to express the lower bound of an algorithm's running time.

9. The Theta notation is the formal way to express _____ of an algorithm's running time.

- A. upper bound
- B. lower bound
- C. lower bound and upper bound
- D. None of the above

[View Answer](#)

Ans : C

Explanation: The notation $\theta(n)$ is the formal way to express both the lower bound and the upper bound of an algorithm's running time.

10. Asymptotic analysis is _____ bound.

- A. output
- B. input
- C. outer
- D. inner

[View Answer](#)

Ans : B

Explanation: Asymptotic analysis is input bound i.e., if there's no input to the algorithm, it is concluded to work in a constant time. Other than the input all other factors are considered constant.

codewitharrays.in 8007592194

What will be the output of `cout << (2 << 2) << endl;`

- 1) 4
- 2) 8
- 3) 2
- 4) None of the above

Answer : 8

What is the output of `cout << (5 || 0) << endl;`

- 1) 1
- 2) 0
- 3) True
- 4) False

Answer : 1

Which of the following is a correct way to define a destructor in C++?

- 1) `~MyClass() {}`
- 2) `MyClass::~~MyClass() {}`
- 3) `MyClass() ~ {}`
- 4) None of the above

Answer : `MyClass::~~MyClass() {}`

What is the output of `cout << (3 & 4) << endl;`

- 1) 0
- 2) 1
- 3) 3
- 4) 7

Answer : Option 1

Which of the following cannot be used as a function argument in C++?

- 1) `int`
- 2) `float`

- 3) char
- 4) None of the above

Answer : None of the above

What is the output of `cout << (6 - 2 * 2) << endl;`

- 1) 2
- 2) 4
- 3) 6
- 4) None of the above

Answer : 2

Which of the following is a correct way to define a nested class in C++?

- 1) `class Outer { class Inner {}; };`
- 2) `class Outer { Inner class {}; };`
- 3) `class Outer::Inner {};`
- 4) None of the above

Answer : `class Outer { class Inner {}; };`

What will be the output of `cout << (10 / 2 * 3) << endl;`

- 1) 15
- 2) 10
- 3) 5
- 4) None of the above

Answer : 15

Which of the following access specifiers allows access to members of a derived class?

- 1) public
- 2) private
- 3) protected
- 4) All of the above

Answer : protected

What is the output of `cout << (4 == 4 && 2 < 5) << endl;`

- 1) 1
- 2) 0
- 3) true
- 4) false

Answer : 1

codewitharrays.in 8007592194

1. Which of the following case does not exist in complexity theory?

- A. Best case
- B. Worst case
- C. Average case
- D. Null case

[View Answer](#)

Ans : D
Explanation: Null case does not exist in complexity Theory.

2. What is the time, space complexity of following code:

```
int a = 0, b = 0;
for (i = 0; i < N; i++)
{
a = a + rand();
}
for (j = 0; j < M; j++)
{
b = b + rand();
}
```

- A. $O(N * M)$ time, $O(1)$ space
- B. $O(N + M)$ time, $O(N + M)$ space
- C. $O(N + M)$ time, $O(1)$ space
- D. $O(N * M)$ time, $O(N + M)$ space

[View Answer](#)

Ans : C
Explanation: The first loop is $O(N)$ and the second loop is $O(M)$. Since we don't know which is bigger, we say this is $O(N + M)$. This can also be written as $O(\max(N, M))$. Since there is no additional space being utilized, the space complexity is constant / $O(1)$.

3. The complexity of linear search algorithm is

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

[View Answer](#)

Ans : A
Explanation: The worst case complexity of linear search is $O(n)$.

4. What is the time complexity of following code:

```
int a = 0;
for (i = 0; i < N; i++)
{
for (j = N; j > i; j--)
{
a = a + i + j;
}
```

```
}  
}
```

- A. $O(N)$
- B. $O(N \cdot \log(N))$
- C. $O(N \cdot \sqrt{N})$
- D. $O(N^2)$

[View Answer](#)

Ans : D

Explanation: $= N + (N - 1) + (N - 2) + \dots + 1 + 0$
 $= N \cdot (N + 1) / 2$
 $= 1/2 \cdot N^2 + 1/2 \cdot N$
 $O(N^2)$ times.

5. The Worst case occur in linear search algorithm when

- A. Item is somewhere in the middle of the array
- B. Item is not in the array at all
- C. Item is the last element in the array
- D. Item is the last element in the array or is not there at all

[View Answer](#)

Ans : D

Explanation: The Worst case occur in linear search algorithm when Item is the last element in the array or is not there at all.

6. What is the time complexity of following code:

```
int i, j, k = 0;  
for (i = n / 2; i <= n; i++)  
{  
    for (j = 2; j <= n; j = j * 2)  
    {  
        k = k + n / 2;  
    }  
}
```

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

[View Answer](#)

Ans : B

Explanation: Let's take the examples here.

for $n = 16$, $j = 2, 4, 8, 16$

for $n = 32$, $j = 2, 4, 8, 16, 32$

So, j would run for $O(\log n)$ steps.

i runs for $n/2$ steps.

So, total steps = $O(n/2 \cdot \log(n)) = O(n \log n)$

7. The worst case occur in quick sort when

- A. Pivot is the median of the array
- B. Pivot is the smallest element

- C. Pivot is the middle element
- D. None of the mentioned

[View Answer](#)

Ans : B
Explanation: This happens when the pivot is the smallest (or the largest) element. Then one of the partitions is empty, and we repeat recursively the procedure for N-1 elements.

8. What does it mean when we say that an algorithm X is asymptotically more efficient than Y?

- A. X will always be a better choice for small inputs
- B. X will always be a better choice for large inputs
- C. Y will always be a better choice for small inputs
- D. X will always be a better choice for all inputs

[View Answer](#)

Ans : B
Explanation: An algorithm X is said to be asymptotically better than Y if X takes smaller time than y for all input sizes n larger than a value n_0 where $n_0 > 0$.

9. The complexity of Fibonacci series is

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

[View Answer](#)

Ans : A
Explanation: Fibonacci is $f(n) = f(n-1) + f(n-2)$, $f(0) = 0$, $f(1) = 1$. Let $g(n) = 2n$. Now prove inductively that $f(n) \geq g(n)$.

10. What is the time complexity of following code:

```
int a = 0, i = N;
while (i > 0)
{
    a += i;
    i /= 2;
}
```

- A. $O(N)$
- B. $O(\text{Sqrt}(N))$
- C. $O(N / 2)$
- D. $O(\log N)$

[View Answer](#)

Ans : D
Explanation: We have to find the smallest x such that $N / 2^x \times N^x = \log(N)$.

11. What is the time complexity of following code:

```
int a = 0, i = N;
while (i > 0)
{
```

```
a += i;  
i /= 2;  
}
```

- A. $O(N)$
- B. $O(\text{Sqrt}(N))$
- C. $O(N / 2)$
- D. $O(\log N)$

[View Answer](#)

Ans : D

Explanation: We have to find the smallest x such that $N / 2^x N^x = \log(N)$

12. The complexity of Binary search algorithm is

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

[View Answer](#)

Ans : B

Explanation: The compexity of binary search is $O(\log n)$.

13. The complexity of merge sort algorithm is

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

[View Answer](#)

Ans : D

Explanation: The worst case complexity for merge sort is $O(n \log n)$.

14. The complexity of Bubble sort algorithm is

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

[View Answer](#)

Ans : C

Explanation: The worst case complexity for Bubble sort is $O(n^2)$ ans best case is $O(n)$.

15. The worst case complexity for insertion sort is

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

View Answer

Ans : C

Explanation: In worst case n th comparison are required to insert the n th element into correct position.

16. The worst case complexity of quick sort is

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

View Answer

Ans : C

Explanation: The worst case complexity of quick sort is $O(n^2)$.

17. To measure Time complexity of an algorithm Big O notation is used which:

- A. describes limiting behaviour of the function
- B. characterises a function based on growth of function
- C. upper bound on growth rate of the function
- D. all of the mentioned

View Answer

Ans : D

Explanation: Big O notation describes limiting behaviour, and also gives upper bound on growth rate of a function.

18. If for an algorithm time complexity is given by $O(1)$ then complexity of it is:

- A. constant
- B. polynomial
- C. exponential
- D. none of the mentioned

View Answer

Ans : A

Explanation: The growth rate of that function will be constant.

19. If for an algorithm time complexity is given by $O(\log^2 n)$ then complexity will:

- A. constant
- B. polynomial
- C. exponential
- D. none of the mentioned

View Answer

Ans : D

Explanation: The growth rate of that function will be logarithmic therefore complexity will be logarithmic.

20. If for an algorithm time complexity is given by $O(n)$ then complexity of it is:

- A. constant
- B. linear

- C. exponential
- D. none of the mentioned

[View Answer](#)

Ans : B

Explanation: The growth rate of that function will be linear.

21. if for an algorithm time complexity is given by $O(n^2)$ then complexity will:

- A. constant
- B. quadratic
- C. exponential
- D. none of the mentioned

[View Answer](#)

Ans : B

Explanation: The growth rate of that function will be quadratic therefore complexity will be quadratic..

22. If for an algorithm time complexity is given by $O((3/2)^n)$ then complexity will:

- A. constant
- B. quadratic
- C. exponential
- D. none of the mentioned

[View Answer](#)

Ans : C

Explanation: The growth rate of that function will be exponential therefore complexity will be exponential.

23. the time complexity of binary search is given by:

- A. constant
- B. quadratic
- C. exponential
- D. none of the mentioned

[View Answer](#)

Ans : D

Explanation: It is $O(\log_2 n)$, therefore complexity will be logarithmic.

24. The time complexity of linear search is given by:

- A. $O(\log_2 n)$
- B. $O(1)$
- C. exponential
- D. none of the mentioned

[View Answer](#)

Ans : D

Explanation: It is $O(n)$, therefore complexity will be linear.

25. Which algorithm is better for sorting between bubble sort and quicksort?

- A. bubble sort
- B. quick sort
- C. both are equally good
- D. none of the mentioned

[View Answer](#)

Ans : B

Explanation: Running time of quicksort is logarithmic whereas for bubble sort it is quadratic

26. State true or false

Time complexity of binary search algorithm is constant

- A. True
- B. False

[View Answer](#)

Ans : B

Explanation: It is $O(\log 2n)$, therefore complexity will be logarithmic.

27. Two main measures for the efficiency of an algorithm are

- A. Time and space
- B. Processor and memory
- C. Complexity and capacity
- D. Data and space

[View Answer](#)

Ans : A

Explanation: None

28. Which is the best data structure for round robin algorithm for CPU scheduling?

- A. Stack implemented using queues
- B. Doubly linked list
- C. Circular queue
- D. Queue implemented using stacks

[View Answer](#)

Ans : C

Explanation: None

29. Which algorithm is having highest space complexity?

- A. Bubble sort
- B. Insertion Sort
- C. Quick Sort
- D. Merge Sort

[View Answer](#)

Ans : D

Explanation: None

30. If the array is already sorted, then the running time for merge sort is: ?

- A. $O(1)$
- B. $O(n \log n)$
- C. $O(n)$
- D. $O(n^2)$

[View Answer](#)

Ans : B

Explanation:None

codewitharrays.in 8007592194

1. What is a hash table?

- A. A structure that maps values to keys
- B. A structure that maps keys to values
- C. A structure used for storage
- D. A structure used to implement stack and queue

View Answer

Ans : B

Explanation: A hash table is used to implement associative arrays which has a key-value pair, so the has table maps keys to values.

2. 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

View Answer

Ans : C

Explanation: In a valid insertion sequence, the elements 42, 23 and 34 must appear before 52 and 33, and 46 must appear before 33.

Total number of different sequences = $3! \times 5 = 30$

In the above expression, $3!$ is for elements 42, 23 and 34 as they can appear in any order, and 5 is for element 46 as it can appear at 5 different places.

3. If 'h' is a hashing function and it is used to hash 'n' keys into a table of size 'm' where $n \leq m$. What is the expected number of collisions involving a particular key 'x'?

- A. less than 1.
- B. less than n.
- C. less than m.
- D. less than $n / 2$.

View Answer

Ans : A

Explanation: No Explanation

4.Hashing technique which allocates fixed number of buckets is classified as

- A. dynamic hashing
- B. static hashing
- C. external hashing
- D. internal hashing

View Answer

Ans : C

Explanation:Hashing technique which allocates fixed number of buckets is classified as external hashing.

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

- A. Diffusion
- B. Replication
- C. Collision
- D. None of the mentioned

View Answer

Ans : C

Explanation: If several elements are competing for the same bucket in the hash table, it is called Collision.

6. 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

View Answer

Ans : B

Explanation: hashing and probing. Let us put values 1, 3, 8, 10 in the hash of size 7. Initially, hash table is empty

0 1 2 3 4 5 6

The value of function $(3x + 4) \bmod 7$ for 1 is 0, so let us put the value at 0

1 -----

0 1 2 3 4 5 6

The value of function $(3x + 4) \bmod 7$ for 3 is 6, so let us put the value at 6

1 ----- 3

0 1 2 3 4 5 6

The value of function $(3x + 4) \bmod 7$ for 8 is 0, but 0 is already occupied, let us put the value(8) at next available space(1)

1 8 ----- 3

0 1 2 3 4 5 6

The value of function $(3x + 4) \bmod 7$ for 10 is 6, but 6 is already occupied, let us put the value(10) at next available space(2)

1 8 10 --- 3

0 1 2 3 4 5 6

7. What is the advantage of using a dynamic set in direct addressing?

- A. It saves time
- B. It saves space
- C. It saves both time and space
- D. None of the mentioned

View Answer

Ans : B

Explanation: Using a dynamic set, the size of the array is restricted to the number of keys, hence saves space.

8. In linear hashing, formula used to calculate number of records if blocking factor, loading factor and file buckets are known is as

- A. $r = l + bfr + N$
- B. $r = l - bfr - N$
- C. $r = l + bfr - N$
- D. $r = l * bfr * N$

[View Answer](#)

Ans : D
Explanation: No Explanation

9. What is the load factor?

- A. Average array size
- B. Average key size
- C. Average chain length
- D. None of the mentioned

[View Answer](#)

Ans : 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. .

10. Given a hash table T with 25 slots that stores 2000 elements, the load factor α for T is _____

- A. 80
- B. 0.0125
- C. 8000
- D. 1.25

[View Answer](#)

Ans : A
Explanation: load factor = (no. of elements) / (no. of table slots) = $2000/25 = 80$.

11. What is direct addressing?

- A. Distinct array position for every possible key
- B. Fewer array positions than keys
- C. Fewer keys than array positions
- D. None of the mentioned

[View Answer](#)

Ans : A
Explanation: Direct addressing is possible only when we can afford to allocate an array that has one position for every possible key.

12. What is the search complexity in direct addressing?

- A. $O(n)$
- B. $O(\log n)$
- C. $O(n \log n)$
- D. $O(1)$

[View Answer](#)

Ans : D
Explanation: Since every key has a unique array position, searching takes a constant time.

13. What is a hash function?

- A. A function has allocated memory to keys
- B. A function that computes the location of the key in the array

- C. A function that creates an array
- D. None of the mentioned

[View Answer](#)

Ans : B

Explanation: In a hash table, there are fewer array positions than the keys, so the position of the key in the array has to be computed, this is done using the hash function.

14. What can be the techniques to avoid collision?

- A. Make the hash function appear random
- B. Use the chaining method
- C. Use uniform hashing
- D. All of the mentioned

[View Answer](#)

Ans : D

Explanation: Making the hash function random is not really a good choice, although it is considered one of the techniques to avoid collisions along with chaining and simple uniform hashing.

15. What is simple uniform hashing?

- A. Every element has equal probability of hashing into any of the slots
- B. A weighted probabilistic method is used to hash elements into the slots
- C. All of the mentioned
- D. None of the mentioned

[View Answer](#)

Ans : A

Explanation: In simple uniform hashing, any given element is equally likely to hash into any of the slots available in the array.

16. In simple uniform hashing, what is the search complexity?

- A. $O(n)$
- B. $O(\log n)$
- C. $O(n \log n)$
- D. $O(1)$

[View Answer](#)

Ans : D

Explanation: There are two cases, once when the search is successful and when it is unsuccessful, but in both the cases, the complexity is $O(1 + \alpha)$ where 1 is to compute the hash function and α is the load factor.

17. In simple chaining, what data structure is appropriate?

- A. Singly linked list
- B. Doubly linked list
- C. Circular linked list
- D. Binary trees

[View Answer](#)

Ans : B

Explanation: Deletion becomes easier with doubly linked list, hence it is appropriate.

18. Which of these are core interfaces in the collection framework. Select the one correct answer.

- A. Tree
- B. Stack
- C. Queue
- D. Map
- E. LinkedList

[View Answer](#)

Ans : D
Explanation:None

19. 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

[View Answer](#)

Ans : C
Hash function given is $\bmod(10)$.
9679, 1989 and 4199 all these give same hash value i.e 9
1471 and 6171 give hash value 1

codewitharrays.in 8007592194

1. Which of the following statements for a simple graph is correct?

- A. Every path is a trail
- B. Every trail is a path
- C. Every trail is a path as well as every path is a trail
- D. None of the mentioned

View Answer

Ans : A
Explanation: In a walk if the vertices are distinct it is called a path, whereas if the edges are distinct it is called a trail.

2. What is the maximum number of possible non zero values in an adjacency matrix of a simple graph with n vertices?

- A. $(n*(n-1))/2$
- B. $(n*(n+1))/2$
- C. $n*(n-1)$
- D. $n*(n+1)$

View Answer

Ans : C
Explanation: Out of $n*n$ possible values for a simple graph the diagonal values will always be zero.

3. Which of the following is an advantage of adjacency list representation over adjacency matrix representation of a graph?

- A. In adjacency list representation, space is saved for sparse graphs.
- B. DFS and BSF can be done in $O(V + E)$ time for adjacency list representation. These operations take $O(V^2)$ time in adjacency matrix representation. Here V and E are number of vertices and edges respectively.
- C. Adding a vertex in adjacency list representation is easier than adjacency matrix representation.
- D. All of the above

View Answer

Ans : D
Explanation: No explanation.

4. A connected planar graph having 6 vertices, 7 edges contains _____ regions.

- A. 15
- B. 3
- C. 1
- D. 11

View Answer

Ans : B
Explanation: By Euler's formula the relation between vertices(n), edges(q) and regions(r) is given by $n-q+r=2$.

5. On which of the following statements does the time complexity of checking if an edge exists between two particular vertices is not, depends?

- A. Depends on the number of edges
- B. Depends on the number of vertices

- C. Is independent of both the number of edges and vertices
D. It depends on both the number of edges and vertices

[View Answer](#)

Ans : C

Explanation: To check if there is an edge between two vertices i and j , it is enough to see if the value of $A[i][j]$ is 1 or 0, here A is the adjacency matrix.

6. 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

[View Answer](#)

Ans : D

Explanation: No explanation

7. What is the maximum number of edges in a bipartite graph having 10 vertices?

- A. 24
B. 21
C. 25
D. 16

[View Answer](#)

Ans : C

Explanation: Let one set have n vertices another set would contain $10-n$ vertices.

Total number of edges would be $n*(10-n)$, differentiating with respect to n , would yield the answer.

8. Possible number of labelled simple Directed, Pseudo and Multigraphs exist having 2 vertices?

- A. 3, Infinite, 4
B. 4, 3, Infinite
C. 4, Infinite, infinite
D. 4, Infinite, Infinite

[View Answer](#)

Ans : D

Explanation: MultiGraphs and PseudoGraphs may have infinite number of edges, while 4 possible simple graphs exist.

9. 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) $\theta(mn)$

- A. A
- B. B
- C. C
- D. D

[View Answer](#)

Ans : C

Explanation: Connected components can be found in $O(m + n)$ using Tarjan's algorithm. Once we have connected components, we can count them.

10. What is time complexity to check if a string(length S_1) is a substring of another string(length S_2) stored in a Directed Acyclic Word Graph, given S_2 is greater than S_1 ?

- A. $O(S_1)$
- B. $O(S_2)$
- C. $O(S_1 + S_2)$
- D. $O(1)$

[View Answer](#)

Ans : A

Explanation: For each check of a word of length S_1 , we need to follow at most S_1 edges.

11. What is the number of edges present in a complete graph having n vertices?

- A. $(n*(n+1))/2$
- B. $(n*(n-1))/2$
- C. n
- D. Information given is insufficient

[View Answer](#)

Ans : B

Explanation: Number of ways in which every vertex can be connected to each other is nC_2 .

12. In a simple graph, the number of edges is equal to twice the sum of the degrees of the vertices.

- A. True
- B. False

[View Answer](#)

Ans : B

Explanation: The sum of the degrees of the vertices is equal to twice the number of edges.

13. If a simple graph G , contains n vertices and m edges, the number of edges in the Graph G' (Complement of G) is _____

- A. $(n*n - n - 2*m)/2$
- B. $(n*n + n + 2*m)/2$
- C. $(n*n - n - 2*m)/2$
- D. $(n*n - n + 2*m)/2$

[View Answer](#)

Ans : A

Explanation: The union of G and G' would be a complete graph so, the number of edges in G' = number of edges in the complete form of G(nC_2)-edges in G(m).

14. Which of the following properties does a simple graph not hold?

- A. Must be connected
- B. Must be unweighted
- C. Must have no loops or multiple edges
- D. All of the mentioned

[View Answer](#)

Ans : A

Explanation: A simple graph may be connected or disconnected.

15. Which of the following is true?

- A. A graph may contain no edges and many vertices
- B. A graph may contain many edges and no vertices
- C. A graph may contain no edges and no vertices
- D. None of the mentioned

[View Answer](#)

Ans : B

Explanation: A graph must contain at least one vertex.

16. For a given graph G having v vertices and e edges which is connected and has no cycles, which of the following statements is true?

- A. $v=e$
- B. $v = e+1$
- C. $v + 1 = e$
- D. None of the mentioned

[View Answer](#)

Ans : B

Explanation: For any connected graph with no cycles the equation holds true.

17. for which of the following combinations of the degrees of vertices would the connected graph be eulerian?

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

[View Answer](#)

Ans : A

Explanation: A graph is eulerian if either all of its vertices are even or if only two of its vertices are odd.

18. 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

Ans : B

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

19. Which of the following ways can be used to represent a graph?

- A. Adjacency List and Adjacency Matrix
- B. Incidence Matrix
- C. Adjacency List, Adjacency Matrix as well as Incidence Matrix
- D. None of the mentioned

View Answer

Ans :C

Explanation: The 3 listed methods can be used, the choice depends on the ease of use.

20. The time complexity to calculate the number of edges in a graph whose information is stored in form of an adjacency matrix is _____

- A. $O(V)$
- B. $O(E^2)$
- C. $O(E)$
- D. $O(V^2)$

View Answer

Ans : D

Explanation: As V entries are 0, a total of $V^2 - V$ entries are to be examined.

1. A _____ is a special Tree-based data structure in which the tree is a complete binary tree.?

- A. Graph
- B. Heap
- C. List
- D. Stack

[View Answer](#)

Ans : B

Explanation: A Heap is a special Tree-based data structure in which the tree is a complete binary tree.

2. How many type of heap are there?

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

[View Answer](#)

Ans : A

Explanation: There are 2 types of heap : max-heap and min-heap.

3. In which heap the root node must be greatest among the keys present at all of its children?

- A. min-heap
- B. max-heap
- C. Both A and B
- D. None of the above

[View Answer](#)

Ans : B

Explanation : Max-Heap: In a Max-Heap the key present at the root node must be greatest among the keys present at all of its children

4. What is the complexity of adding an element to the heap?

- A. $O(\log n)$
- B. $O(\log h)$
- C. $O(h)$
- D. Both A and C

[View Answer](#)

Ans : D

Explanation : The total possible operation in re locating the new location to a new element will be equal to height of the heap.

5. Heap can be used as _____

- A. Priority queue
- B. Stack
- C. A decreasing order array
- D. Normal Array

[View Answer](#)

Ans : A
Explanation : The property of heap that the value of root must be either greater or less than both of its children makes it work like a priority queue.

6. An array consists of n elements. We want to create a heap using the elements. The time complexity of building a heap will be in order of

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

[View Answer](#)

Ans : B
Explanation: The total time taken will be N times the complexity of adding a single element to the heap. And adding a single element takes $\log N$ time, so That is equal to $N \log N$.

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

- A. True
- B. False

[View Answer](#)

Ans : B
Explanation: In min heap, the insertion and deletion operation takes $O(\log n)$ time. Therefore, a selection sort with n insertions and n deletions can be implemented using a min heap in $O(n \log n)$ operations.

8. Which one of the following array elements represents a binary min heap?

- A. 12 10 8 25 14 17
- B. 8 10 12 25 14 17
- C. 25 17 14 12 10 8
- D. 14 17 25 10 12 8

[View Answer](#)

Ans : B
Explanation : A tree is min heap when data at every node in the tree is smaller than or equal to it's children's data. So, only 8 10 12 25 14 17 generates required tree.

9. Given an array of element 5, 7, 9, 1, 3, 10, 8, 4. Which of the following is the correct sequences of elements after inserting all the elements in a min-heap?

- A. 1,3,4,5,7,8,9,10
- B. 1,4,3,9,8,5,7,10
- C. 1,3,4,5,8,7,9,10
- D. 1,3,7,4,8,5,9,10

[View Answer](#)

Ans : A
Explanation : Building a min-heap the result will a sorted array so the 1, 3, 4, 5, 7, 8, 9, 10 is correct. If we change the implementation strategy 1, 4, 3, 8, 9, 5, 7, 10 is also correct. (First filling the right child rather than left child first).

10. What is the amortized cost per operation of a skew heap?

- A. $O(N)$
- B. $O(N \log N)$
- C. $O(N^2)$
- D. $O(\log N)$

[View Answer](#)

Ans : D

Explanation : The amortized cost per operation of a skew heap is $O(\log N)$ since the worst case analysis of skew heap is $O(N)$ and splay tree is $O(M \log N)$.

codewitharrays.in 8007592194

1. What does the following function do for a given Linked List with first node as head?

```
void fun1(struct node* head)
{
    if(head == NULL)
        return;
    fun1(head->next);
    printf("%d ", head->data);
}
```

- A. Prints all nodes of linked lists
- B. Prints all nodes of linked list in reverse order
- C. Prints alternate nodes of Linked List
- D. Prints alternate nodes in reverse order

[View Answer](#)

Ans : B
Explanation: fun1() prints the given Linked List in reverse manner. For Linked List 1->2->3->4->5, fun1() prints 5->4->3->2->1.

2. A linear collection of data elements where the linear node is given by means of pointer is called?

- A. linked list
- B. node list
- C. primitive list
- D. None of these

[View Answer](#)

Ans : A
Explanation: A linear collection of data elements where the linear node is given by means of pointer is called linked list.

3. What is the time complexity to count the number of elements in the linked list?

- A. O(1)
- B. O(n)
- C. O(logn)
- D. None of the mentioned

[View Answer](#)

Ans : B
Explanation: To count the number of elements, you have to traverse through the entire list, hence complexity is O(n).

4. What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?

- A. $O(1)$
- B. $O(n)$
- C. $\theta(n)$
- D. $\theta(1)$

[View Answer](#)

Ans : C
Explanation: No Explanation.

5. What is the output of following function for start pointing to first node of following linked list? 1->2->3->4->5->6

```
void fun(struct node* start)
{
    if(start == NULL)
        return;
    printf("%d ", start->data);
    if(start->next != NULL )
        fun(start->next->next);
    printf("%d ", start->data);
}
```

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

[View Answer](#)

Ans : D
Explanation:fun() prints alternate nodes of the given Linked List, first from head to end, and then from end to head. If Linked List has even number of nodes, then skips the last node.

6. What is the functionality of the following piece of code?

```
public int function(int data)
{
    Node temp = head;
    int var = 0;
    while(temp != null)
    {
        if(temp.getData() == data)
        {
            return var;
        }
        var = var+1;
        temp = temp.getNext();
    }
    return Integer.MIN_VALUE;
}
```

- A. Find and delete a given element in the list
- B. Find and return the given element in the list

- C. Find and return the position of the given element in the list
- D. Find and insert a new element in the list

[View Answer](#)

Ans : C

Explanation: When temp is equal to data, the position of data is returned.

7. Linked lists are not suitable to for the implementation of?

- A. Insertion sort
- B. Radix sort
- C. Polynomial manipulation
- D. Binary search

[View Answer](#)

Ans : D

Explanation: Linked lists are not suitable to for the implementation of Binary search.

8. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is

- A. $\log_2 n$
- B. $n/2$
- C. $\log_2 n - 1$
- D. n

[View Answer](#)

Ans : D

Explanation: In the worst case, the element to be searched has to be compared with all elements of linked list.

9. Which of these is an application of linked lists?

- A. To implement file systems
- B. For separate chaining in hash-tables
- C. To implement non-binary trees
- D. All of the mentioned

[View Answer](#)

Ans : D

Explanation: Linked lists can be used to implement all of the above mentioned applications.

10. In circular linked list, insertion of node requires modification of?

- A. One pointer
- B. Two pointer
- C. Three pointer
- D. None

[View Answer](#)

Ans : B

Explanation: In circular linked list, insertion of node requires modification of Two pointer.

11. Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only.

Given the representation, which of the following operation can be implemented in $O(1)$ time?

- i) Insertion at the front of the linked list
- ii) Insertion at the end of the linked list
- iii) Deletion of the front node of the linked list
- iv) Deletion of the last node of the linked list

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

[View Answer](#)

Ans : B
Explanation: None.

12. In linked list each node contain minimum of two fields. One field is data field to store the data second field is?

- A. Pointer to character
- B. Pointer to integer
- C. Pointer to node
- D. Node

[View Answer](#)

Ans : C
Explanation: None.

13. What would be the asymptotic time complexity to find an element in the linked list?

- A. $O(1)$
- B. $O(n)$
- C. $O(n^2)$
- D. None of the mentioned

[View Answer](#)

Ans : B
Explanation: None.

14. The concatenation of two list can performed in $O(1)$ time. Which of the following variation of linked list can be used?

- A. Singly linked list
- B. Doubly linked list
- C. Circular doubly linked list
- D. Array implementation of list

[View Answer](#)

Ans : C
Explanation: None.

15. Consider the following definition in c programming language.Which of the following c code is used to create new node?

```
struct node
{
int data;
struct node * next;
}
typedef struct node NODE;
NODE *ptr;
```

- A. ptr = (NODE*)malloc(sizeof(NODE));
- B. ptr = (NODE*)malloc(NODE);
- C. ptr = (NODE*)malloc(sizeof(NODE*));
- D. ptr = (NODE)malloc(sizeof(NODE));

View Answer

Ans :A
Explanation: As it represents the right way to create a node.

16. 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

View Answer

Ans : D
Explanation: None.

17. Linked lists are not suitable to for the implementation of?

- A. Insertion sort
- B. Radix sort
- C. Polynomial manipulation
- D. Binary search

View Answer

Ans : D
Explanation: It cannot be implemented using linked lists.

18. Linked list is considered as an example of _____ type of memory allocation.

- A. Dynamic
- B. Static
- C. Compile time
- D. None of the mentioned

View Answer

Ans : A
Explanation: As memory is allocated at the run time.

19. In Linked List implementation, a node carries information regarding

- A. Data
- B. Link
- C. Data and Link
- D. None of the mentioned

[View Answer](#)

Ans : B
Explanation: None.

20. Linked list data structure offers considerable saving in

- A. Computational Time
- B. Space Utilization
- C. Space Utilization and Computational Time
- D. None of the mentioned

[View Answer](#)

Ans : C
Explanation: Linked lists saves both space and time.

21. 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. All of the mentioned

[View Answer](#)

Ans : D
Explanation: None.

22. 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
- D. Merge Sort

[View Answer](#)

Ans : D
Explanation: Both Merge sort and Insertion sort can be used for linked lists. The slow random-access performance of a linked list makes other algorithms (such as quicksort) perform poorly, and others (such as heapsort) completely impossible. Since worst case time complexity of Merge Sort is $O(n\log n)$ and Insertion sort is $O(n^2)$, merge sort is preferred.

23. Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list?

- A. Possible if X is not last node
- B. Possible if size of linked list is even

- C. Possible if size of linked list is odd
- D. Possible if X is not first node

[View Answer](#)

Ans : A

Explanation: Following are simple steps.

```
struct node *temp = X->next;
```

```
X->data = temp->data;
```

```
X->next = temp->next;
```

```
free(temp);
```

24. The following function reverse() is supposed to reverse a singly linked list. There is one line missing at the end of the function. What should be added in place of "/*ADD A STATEMENT HERE*/", so that the function correctly reverses a linked list.

```
struct node
{
int data;
struct node* next;
};
static void reverse(struct node** head_ref)
{
struct node* prev = NULL;
struct node* current = *head_ref;
struct node* next;
while (current != NULL)
{
next = current->next;
current->next = prev;
current = next;
}
/*ADD A STATEMENT HERE*/
}
```

- A. *head_ref = prev;
- B. *head_ref = current;
- C. *head_ref = next;
- D. *head_ref = NULL;

[View Answer](#)

Ans : A

Explanation: *head_ref = prev; At the end of while loop, the prev pointer points to the last node of original linked list.

We need to change *head_ref so that the head pointer now starts pointing to the last node.

25. 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 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

[View Answer](#)

Ans : B

Explanation: The function rearrange() exchanges data of every node with its next node. It starts exchanging data from the first node itself.

codewitharrays.in 8007592194

1. 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

[View Answer](#)

Ans : D
Explanation: No explanation.

2. In linked list implementation of queue, if only front pointer is maintained, which of the following operation take worst case linear time?

- A. Insertion
- B. Deletion
- C. To empty a queue
- D. Both Insertion and To empty a queue

[View Answer](#)

Ans : D
Explanation: Since front pointer is used for deletion, so worst time for the other two cases.

3. Let the following circular queue can accommodate maximum six elements with the following data.What will happen after ADD O operation takes place?

front = 2 rear = 4
queue = _____; L, M, N, ___, __

- A. front = 2 rear = 5
queue = _____; L, M, N, O, __
- B. front = 3 rear = 5
queue = L, M, N, O, __
- C. front = 3 rear = 4
queue = _____; L, M, N, O, __
- D. front = 2 rear = 4
queue = L, M, N, O, __

[View Answer](#)

Ans : A
Explanation: No Explanation

4.How many stacks are needed to implement a queue. Consider the situation where no other data structure like arrays, linked list is available to you.

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

[View Answer](#)

Ans : B

Explanation: A queue can be implemented using two stacks.

5. In a Queue, if a user tries to remove an element from empty Queue it is called _____.

- A. Underflow
- B. Empty collection
- C. Overflow
- D. Garbage Collection

[View Answer](#)

Ans : A

Explanation: In a Queue, if a user tries to remove an element from empty Queue it is called Underflow.

6. If the elements "A", "B", "C" and "D" are placed in a queue and are deleted one at a time, in what order will they be removed?

- A. ABCD
- B. DCBA
- C. DCAB
- D. ABCD

[View Answer](#)

Ans : A

Explanation: Because Queue follow FIFO rule.

7. A priority queue can efficiently implemented using which of the following data structures? Assume that the number of insert and peek (operation to see the current highest priority item) and extraction (remove the highest priority item) operations are almost same

- A. Array
- B. Linked List
- C. Heap Data Structures like Binary Heap, Fibonacci Heap
- D. None of the above

[View Answer](#)

Ans : C

Explanation: No explanation.

8. In case of insertion into a linked queue, a node borrowed from the _____ list is inserted in the queue.

- A. AVAIL
- B. FRONT
- C. REAR
- D. None of the mentioned

[View Answer](#)

Ans : A

Explanation: All the nodes are collected in AVAIL list.

9.If the MAX_SIZE is the size of the array used in the implementation of circular queue. How is rear manipulated while inserting an element in the queue?

- A. rear=(rear%1)+MAX_SIZE
- B. rear=rear%(MAX_SIZE+1)
- C. rear=(rear+1)%MAX_SIZE
- D. rear=rear+(1%MAX_SIZE)

[View Answer](#)

Ans : C
Explanation: No explanation.

10.Suppose a circular queue of capacity (n – 1) elements is implemented with an array of n elements. Assume that the insertion and deletion operation 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) mod n == FRONT, empty: REAR == FRONT
- B. Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR

- C. Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT
- D. Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT

[View Answer](#)

Ans : A
Explanation:No Explanation.

11. A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as a ?

- A Queue
- B Stack
- C. Tree
- D. Linked list

[View Answer](#)

Ans : A
Explanation: None.

12. The data structure required for Breadth First Traversal on a graph is?

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

[View Answer](#)

Ans : C
Explanation: None.

13. A queue is a ?

- A. FIFO (First In First Out) list
- B. LIFO (Last In First Out) list
- C. Ordered array
- D. Linear tree

[View Answer](#)

Ans : A
Explanation: None.

14. In Breadth First Search of Graph, which of the following data structure is used?

- A. Stack
- B. Queue
- C. Linked list
- D. None of the mentioned

[View Answer](#)

Ans : B
Explanation: None

15. 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](#)

Ans : C
Explanation: None.

16. A normal queue, if implemented using an array of size MAX_SIZE, gets full when

- A. $\text{Rear} = \text{MAX_SIZE} - 1$
- B $\text{Front} = (\text{rear} + 1) \bmod \text{MAX_SIZE}$
- C. $\text{Front} = \text{rear} + 1$
- D> $\text{Rear} = \text{front}$

[View Answer](#)

Ans : C
Explanation: Condition for size of queue.

17. Queues serve major role in

- A. Simulation of recursion
- B. Simulation of arbitrary linked list
- C. Simulation of limited resource allocation
- D. All of the mentioned

[View Answer](#)

Ans : C
Explanation: Rest all are implemented using other data structures.

18. 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](#)

Ans : B
Explanation: Queue always has two ends.

19. Which of the following properties is associated with a queue?

- A. First In Last Out
- B. First In First Out
- C. Last In First Out
- D. None of the mentioned

[View Answer](#)

Ans : B
Explanation: Queue follows First In First Out structure.

20. What does the following piece of code do?

```
public Object function()  
{  
    if(isEmpty())  
        return -999;  
    else  
    {  
        Object high;  
        high = q[front];  
        return high;  
    }  
}
```

- A. Dequeue
- B. Enqueue
- C. Return the front element
- D. None of the mentioned

[View Answer](#)

Ans : C

Explanation: `q[front]` gives the element at the front of the queue, since we are not moving the 'front' to the next element, it is not a dequeue operation.

codewitharrays.in 8007592194

1. 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 $O(n \log n)$

[View Answer](#)

Ans : B
Explanation: No explanation.

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

- A. Insertion sort
- B. Selection sort
- C. Bubble sort
- D. Merge sort

[View Answer](#)

Ans : B
Explanation: Selection sort is not a stable sorting algorithm.

3. What is an external sorting algorithm?

- A. Algorithm that uses tape or disk during the sort
- B. Algorithm that uses main memory during the sort
- C. Algorithm that involves swapping
- D. Algorithm that are considered 'in place'

[View Answer](#)

Ans : A
Explanation: As the name suggests, external sorting algorithm uses external memory like tape or disk.

4. If the number of records to be sorted is small, then sorting can be efficient.

- A. Merge
- B. Heap
- C. Selection
- D. Bubble

[View Answer](#)

Ans : C
Explanation: Selection sorting can be efficient.

5. Suppose we have a $O(n)$ time algorithm that finds median of an unsorted array. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. What will be the worst case time complexity of this modified QuickSort.

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

[View Answer](#)

6. Which of the following is not an in-place sorting algorithm?

- A. Selection sort
- B. Heap sort
- C. Quick sort
- D. Merge sort

[View Answer](#)

Ans : D

Explanation: Merge sort is not an in-place sorting algorithm.

7. What is the advantage of bubble sort over other sorting techniques?

- A. It is faster
- B. Consumes less memory
- C. Detects whether the input is already sorted
- D. All of the mentioned

[View Answer](#)

Ans : C

Explanation: Bubble sort is one of the simplest sorting techniques and perhaps the only advantage it has over other techniques is that it can detect whether the input is already sorted.

8. The complexity of sorting algorithm measures the as a function of the number n of items to be sorted.

- A. average time
- B. running time
- C. average-case complexity
- D. case-complexity

[View Answer](#)

Ans : B

Explanation: The complexity of sorting algorithm measures the running time as a function of the number n of items to be sorted.

9. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this:

2 5 1 7 9 12 11 10

Which statement is correct?

- A. The pivot could be either the 7 or the 9.
- B. The pivot could be the 7, but it is not the 9
- C. The pivot is not the 7, but it could be the 9
- D. Neither the 7 nor the 9 is the pivot.

[View Answer](#)

Ans : A

Explanation: 7 and 9 both are at their correct positions (as in a sorted array). Also, all elements on left of 7 and 9 are smaller than 7 and 9 respectively and on right are greater than 7 and 9 respectively.

10. Consider the situation in which assignment operation is very costly. Which of the following sorting algorithm should be performed so that the number of assignment operations is minimized in general?

- A. Insertion sort
- B. Selection sort
- C. Heap sort
- D. None

[View Answer](#)

Ans : B

Explanation: Selection sort.

11. 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](#)

Ans : C

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

12. What is the worst case complexity of selection sort?

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

[View Answer](#)

Ans : DC

Explanation: Selection sort creates a sub-list, LHS of the 'min' element is already sorted and RHS is yet to be sorted. Starting with the first element the 'min' element moves towards the final element.

13. What is the advantage of selection sort over other sorting techniques?

- a) It requires no additional storage space
- b) It is scalable
- c) It works best for inputs which are already sorted
- d) It is faster than any other sorting technique

[View Answer](#)

Ans : A

Explanation: Since selection sort is an in-place sorting algorithm, it does not require additional storage.

14. What is the average case complexity of selection sort?

- A. $O(n \log n)$
- B. $O(\log n)$

- C. $O(n)$
- D. $O(n^2)$

[View Answer](#)

Ans : D

Explanation: In the average case, even if the input is partially sorted, selection sort behaves as if the entire array is not sorted. Selection sort is insensitive to input.

15. What is the disadvantage of selection sort?

- A. It requires auxiliary memory
- B. It is not scalable
- C. It can be used for small keys
- D. None of the mentioned

[View Answer](#)

Ans : B

Explanation: As the input size increases, the performance of selection sort decreases.

16. The given array is $arr = \{3, 4, 5, 2, 1\}$. The number of iterations in bubble sort and selection sort respectively are,

- A. 5 and 4
- B. 4 and 5
- C. 2 and 4
- D. 2 and 5

[View Answer](#)

Ans : A

Explanation: Since the input array is not sorted, bubble sort takes 5 iterations and selection sort takes $4(n-1)$ iterations.

17. The given array is $arr = \{1, 2, 3, 4, 5\}$. (bubble sort is implemented with a flag variable) The number of iterations in selection sort and bubble sort respectively are,

- A. 5 and 4
- B. 1 and 4
- C. 0 and 4
- D. 4 and 1

[View Answer](#)

Ans : B

Explanation: Selection sort is insensitive to input, hence $4(n-1)$ iterations. Whereas bubble sort iterates only once to set the flag to 0 as the input is already sorted.

18. What is the best case complexity of selection sort?

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

[View Answer](#)

Ans : D

Explanation: The best, average and worst case complexities of selection sort is $O(n^2)$.
 $(n-1) + (n-2) + (n-3) + \dots + 1 = (n(n-1))/2 \sim (n^2)/2$.

19. What is an internal sorting algorithm?

- A. Algorithm that uses tape or disk during the sort
- B. Algorithm that uses main memory during the sort
- C. Algorithm that involves swapping
- D. Algorithm that are considered 'in place'

[View Answer](#)

Ans : B

Explanation: As the name suggests, internal sorting algorithm uses internal main memory.

20. What is the worst case complexity of bubble sort?

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

[View Answer](#)

Ans : D

Explanation: Bubble sort works by starting from the first element and swapping the elements if required in each iteration.

21. What is the average case complexity of bubble sort?

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

[View Answer](#)

Ans : D

Explanation: Bubble sort works by starting from the first element and swapping the elements if required in each iteration even in the average case.

22. The given array is $arr = \{1, 2, 4, 3\}$. Bubble sort is used to sort the array elements. How many iterations will be done to sort the array?

- A. 4
- B. 2
- C. 1
- D. 0

[View Answer](#)

Ans : A

Explanation: Even though the first two elements are already sorted, bubble sort needs 4 iterations to sort the given array.

23. What is the best case efficiency of bubble sort in the improvised version?

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

[View Answer](#)

Ans : B

Explanation: Only 2 elements in the given array are not sorted, hence only 2 iterations are required to sort them.

24. QuickSort can be categorized into which of the following?

- A. Brute Force technique
- B. Divide and conquer
- C. Greedy algorithm
- D. Dynamic programming

[View Answer](#)

Ans : B

Explanation: First you divide(partition) the array based on the pivot element and sort accordingly.

25. What is a randomized QuickSort?

- A. The leftmost element is chosen as the pivot
- B. The rightmost element is chosen as the pivot
- C. Any element in the array is chosen as the pivot
- D. A random number is generated which is used as the pivot

[View Answer](#)

Ans : C

Explanation: QuickSort is randomized by placing the input data in the randomized fashion in the array or by choosing a random element in the array as a pivot.

codewitharrays.in 8007592194

1. 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

View Answer

Ans : D
Explanation: No explanation.

2. Process of inserting an element in stack is called _____.

- A. Create
- B. Push
- C. Evaluation
- D. Pop

View Answer

Ans : B
Explanation: Process of inserting an element in stack is called Push.

3. The postfix form of $A*B+C/D$ is?

- A. $*AB/CD+$
- B. $AB*CD/+$
- C. $A*BC+/D$
- D. $ABCD+/*$

View Answer

Ans : B
Explanation: The postfix form of $A*B+C/D$ is $AB*CD/+$.

4. Which of the following is true about linked list implementation of stack?

- 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

View Answer

Ans : D
Explanation: To keep the Last In First Out order, a stack can be implemented using linked list in two ways: a) In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from beginning. b) In push operation, if new nodes are inserted at the end of linked list, then in pop operation, nodes must be removed from end.

5. In a stack, if a user tries to remove an element from empty stack it is called _____.

- A. Underflow
- B. Empty collection

- C. Overflow
- D. Garbage Collection

[View Answer](#)

Ans : A

Explanation: In a stack, if a user tries to remove an element from empty stack it is called Underflow.

6. Consider the following pseudocode that uses a stack. What is output for input "letsfindc"?

```
declare a stack of characters
while ( there are more characters in the word to read )
{
  read a character
  push the character on the stack
}
while ( the stack is not empty )
{
  pop a character off the stack
  write the character to the screen
}
```

- A. letsfindcletsfindc
- B. cdnifstel
- C. letsfindc
- D. cdnifstelcdnifstel

[View Answer](#)

Ans : B

Explanation: No explanation.

7. Entries in a stack are "ordered". What is the meaning of this statement?

- A. A collection of stacks is sortable
- B. Stack entries may be compared with the '<' operation
- C. The entries are stored in a linked list
- D. There is a Sequential entry that is one by one

[View Answer](#)

Ans : D

Explanation: It means There is a Sequential entry that is one by one.

8. The prefix form of $A-B / (C * D \wedge E)$ is?

- A. $-/*\wedge ACBDE$
- B. $-ABCD*\wedge DE$
- C. $-A/B*C\wedge DE$
- D. $-A/BC*\wedge DE$

[View Answer](#)

Ans : C

Explanation: The prefix form of $A-B / (C * D \wedge E)$ is $-A/B*C\wedge DE$.

9. Following is an incorrect pseudocode for the algorithm which is supposed to determine whether a sequence of parentheses is balanced: Which of these unbalanced sequences does the above code think is balanced?

```
declare a character stack
while ( more input is available)
{
  read a character
  if ( the character is a '(' )
    push it on the stack
  else if ( the character is a ')' and the stack is not empty )
    pop a character off the stack
  else
    print "unbalanced" and exit
}
print "balanced"
```

- A. ((()))
- B. ())()
- C. ((()))
- D. (())()

[View Answer](#)

Ans : A
Explanation: At the end of while loop, we must check whether the stack is empty or not. For input ((())), the stack doesn't remain empty after the loop.

10. What is the value of the postfix expression 6 3 2 4 + - *:

- A. Something between -5 and -15
- B. Something between 5 and -5
- C. Something between 5 and 15
- D. Something between 15 and 100

[View Answer](#)

Ans : D
Explanation: On solving the postfix expression the answer comes out to 18.

11. Process of removing an element from stack is called _____

- A. Create
- B. Push
- C. Evaluation
- D. Pod

[View Answer](#)

Ans : D
Explanation: No explanation.

12. Pushing an element into stack already having five elements and stack size of 5 , then stack becomes

- A. Overflow
- B. Crash

- C. Underflow
- D. User flow

[View Answer](#)

Ans : C
Explanation: None.

13. Which of the following applications may use a stack?

- A. A parentheses balancing program
- B. Tracking of local variables at run time
- C. Compiler Syntax Analyzer
- D. All of the mentioned

[View Answer](#)

Ans : D
Explanation: All are applications of stack.

14. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: $((()((())))$ are:

- A. 1
- B. 2
- C. 3
- D. 4 or more

[View Answer](#)

Ans : C
Explanation: Applying the postfix expression evaluation.

15. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. Suppose that you run the algorithm on a sequence that contains 2 left parentheses and 3 right parentheses (in some order). The maximum number of parentheses that appear on the stack AT ANY ONE TIME during the computation?

- A. 1
- B. 2
- C. 3
- D. 4 or more

[View Answer](#)

Ans : C
Explanation: Applying the postfix expression evaluation.

16. The data structure required to check whether an expression contains balanced parenthesis is?

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

[View Answer](#)

Ans : A
Explanation: None.

17. Here is an infix expression: $4 + 3 * (6 * 3 - 12)$. Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation.
The maximum number of symbols that will appear on the stack AT ONE TIME during the conversion of this expression?

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

[View Answer](#)

Ans : D
Explanation: None.

18. What data structure would you mostly likely see in a non recursive implementation of a recursive algorithm?

- A. Linked List
- B. Stack
- C. Queue
- D. Tree

[View Answer](#)

Ans : B
Explanation: None.

19. The process of accessing data stored in a serial access memory is similar to manipulating data on a _____

- A. Heap
- B. Binary Tree
- C. Array
- D. Stack

[View Answer](#)

Ans : D
Explanation: None.

20. The postfix form of $A * B + C / D$ is?

- A. $*AB/CD+$
- B. $AB*CD/+$
- C. $A*BC+/D$
- D. $ABCD+/*$

[View Answer](#)

Ans : B
Explanation: Applying the postfix expression evaluation.

21. Which data structure is needed to convert infix notation to postfix notation?

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

[View Answer](#)

Ans : D
Explanation: None.

22. The prefix form of $A-B / (C * D ^ E)$ is?

- A. $- / ^ A C B D E$
- B. $- A B C D ^ A D E$
- C. $- A / B ^ C ^ D E$
- D. $- A / B C ^ A D E$

[View Answer](#)

Ans : B
Explanation: Process of inserting an element in stack is called Push.

23. The postfix form of $A * B + C / D$ is?

- A. $* A B / C D +$
- B. $A B * C D / +$
- C. $A * B C + / D$
- D. $A B C D + / *$

[View Answer](#)

Ans : C
Explanation: Applying the prefix expression evaluation..

24. What is the result of the following operation
Top (Push (S, X))

- A. X
- B. Null
- C. S
- D. None of the mentioned

[View Answer](#)

Ans : A
Explanation: None.

25. The prefix form of an infix expression $p + q - r * t$ is?

- A. $+ p q - * r t$
- B. $- + p q r * t$
- C. $- + p q * r t$
- D. $- + * p q r t$

[View Answer](#)

Ans : C

Explanation: Applying the prefix expression evaluation.

codewitharrays.in 8007592194

1. The no of external nodes in a full binary tree with n internal nodes is?

- A. n
- B. n+1
- C. 2n
- D. 2n + 1

View Answer

Ans : B

Explanation: The no of external nodes in a full binary tree with n internal nodes is n+1.

2. Which of the following is a true about Binary Trees?

- A. Every binary tree is either complete or full.
- B. Every complete binary tree is also a full binary tree.
- C. Every full binary tree is also a complete binary tree.
- D. No binary tree is both complete and full.
- E. None of the above

View Answer

Ans : E

Explanation: 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, and all nodes are as far left as possible. A) is incorrect.

3. A Binary Tree can have

- A. Can have 2 children
- B. Can have 1 children
- C. Can have 0 children
- D. All of the above

View Answer

Ans : D

Explanation: A Binary Tree can have 0, 1, 2 children

4. Which of the following is not an advantage of trees?

- A. Hierarchical structure
- B. Faster search
- C. Router algorithms
- D. Undo/Redo operations in a notepad

View Answer

Ans : D

Explanation: This is an application of stack.

5. The difference between the external path length and the internal path length of a binary tree with n internal nodes is?

- A. 1
- B. n

- C. $n + 1$
- D. $2n$

[View Answer](#)

Ans : D

Explanation: The difference between the external path length and the internal path length of a binary tree with n internal nodes is $2n$.

6. In a complete k -ary tree, every internal node has exactly k children or no child. The number of leaves in such a tree with n internal nodes is:

- A. nk
- B. $(n - 1)k + 1$
- C. $n(k - 1) + 1$
- D. $n(k - 1)$

[View Answer](#)

Ans : C

Explanation: For an k -ary tree where each node has k children or no children, following relation holds $L = (k-1)n + 1$ Where L is the number of leaf nodes and n is the number of internal nodes.

7. Height of a binary tree is

- A. $\text{MAX}(\text{Height of left Subtree}, \text{Height of right subtree}) + 1$
- B. $\text{MAX}(\text{Height of left Subtree}, \text{Height of right subtree})$
- C. $\text{MAX}(\text{Height of left Subtree}, \text{Height of right subtree}) - 1$
- D. None of the above

[View Answer](#)

Ans : A

Explanation: Height of a binary tree is $\text{MAX}(\text{Height of left Subtree}, \text{Height of right subtree}) + 1$.

8. Which of the following options is an application of splay trees ?

- A. cache Implementation
- B. networks
- C. send values
- D. receive values

[View Answer](#)

Ans : A

Explanation: Splay trees can be used for faster access to recently accessed items and hence used for cache implementations.

9. Suppose a complete binary tree has height $h > 0$. The minimum no of leaf nodes possible in term of h is?

- A. $2h - 1$
- B. $2^{h-1} + 1$
- C. 2^{h-1}
- D. $2h + 1$

[View Answer](#)

Ans : C

Explanation: Suppose a complete binary tree has height $h > 0$. The minimum no of leaf nodes possible in term of h is 2^{h-1} .

10. 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 farthest leaf) of such a tree on n nodes is best described by which of the following?

- A. $\log_2 n$
- B. $\log_{4/3} n$
- C. $\log_3 n$
- D. $\log_{3/2} n$

[View Answer](#)

Ans : D

Explanation: No Explanation.

codewitharrays.in 8007592194



<https://www.youtube.com/@codewitharrays>



<https://www.instagram.com/codewitharrays/>



<https://t.me/codewitharrays> Group Link: <https://t.me/ccee2025notes>



[+91 8007592194](tel:+918007592194) [+91 9284926333](tel:+919284926333)



codewitharrays@gmail.com



<https://codewitharrays.in/project>