Placement handbook of

DATA STRUCTURES

TOP 40 MCQ'S WITH ANSWERS

www.mobiprep.com

TOP DATA STRUCURES MCQ's WITH ANSWERS

QUES 1). What the following code will do?

```
public Object function(int row_index, int col_index)

{

       if (row_index < 0 || col_index > N)

           {

          System.out.println("column index out of bounds");

                                return;

           }

       return (sparse_array[row_index].fetch(col_index));

}
```

a) Move the element to the specified position
b) Removes the element from the specified position and put it in next position.
c) delete the element from the specified position
d) Get the element from the specified position

Ans: d

QUES 2). The fetch method of SparseArray class is called , the row specified by row_index makes it an array and the col_index denotes the specified position. What is the worst case time complexity of deleting a node in Singly Linked list?

a) O (n)
b) O (n2)
c) O (nlogn)
d) O (1)

Ans; a

You have to traverse the whole list to find the key and in worst case it will found at last. So the time complexity will be O(n)

QUES 3). What does the following function do?

```
public Object FunctionA()throws emptyStackException

{

       if(isEmpty())
```

```
        throw new emptyStackException("underflow");

    return first.getElement();

}
```

a) retrieve the top-of-the-stack element
b) push operation
c) pop
d) delete the top-of-the-stack element

Ans; a

QUES 4). This code is only retrieving the top element, note that it is not equivalent to pop operation as you are not setting the 'next' pointer point to the next node in sequence.

What is the meaning of 'stackoverflow' ?

a) removing items from an empty stack
b) index out of bounds exception
c) accessing item from an undefined stack
d) adding items to a full stack

Ans: d

Adding items to a full stack is termed as stack underflow.

QUES 5). Spanning tree of a graph have.....

a) same number of vertices but not edges.
b) depends upon algorithm being used.
c) same number of edges and vertices.
d) same number of edges and but not vertices.

Ans:a

QUES 6). Spanning tree of a graph have same number of vertices and minimum possible number of edges.

Which data structure is used for parentheses matching?

a) b-tree
b)graph
c)Heap
d)Stack

Ans: d:

QUES 7). For every opening brace, push it into the stack, and for every closing brace, pop it off the stack. Do not take action for any other character. In the end, if the stack is empty, then the input has balanced parentheses.

If a queue is implemented using an array of size MAXIMUM, gets full when

a) Rear = MAXIMUM – 1
b) Front = (rear + 1)
c) Front = rear + 1-MAXIMUM
d) Rear = front

Ans: a:

QUES 8). When Rear = MAXIMUM – 1, there will be no space left for the elements to be added in queue. Thus queue becomes full.

What will be the output of the followind code?

```
public class question

{

    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

Ans: c

It is an 0-index array therefore index 5 gives ArrayIndexOutOfBoundsException.

QUES 9). What will be the output of the followind code?

```
public class question

{
        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

Ans: c

QUES 10). It is an 0-index array therefore index 5 gives ArrayIndexOutOfBoundsException.

What is the size of the array arr[100]?(Assuming int is of 4 bytes)

a) 100
b) 4
c) 400
d) 25

Ans: c:

Since there are 100 int elements and each int is of 4bytes, we get 100*4 = 400bytes.

QUES 11). What this program will do?

```
void push(struct Node** head_ref, int new_data)

{

    struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));

    new_node->data  = new_data;

    new_node->next = (*head_ref);

    (*head_ref)    = new_node;

}
```

a) add node at the front
b) add node at the end
c) add node somewhere in middle
d) Nothing

Ans: a

QUES 12). This program will add the new node at the front of the linkedlist by pointing next pointer of new node to the head and then moving the head to point to the new node.

A dynamic array is?

a) The memory to the array is allocated at runtime
b) An array which is reallocated everytime whenever new elements have to be added
c) A variable size data structure
d) An array which is created at runtime.

Ans:.c

It is a varying-size list data structure that allows items to be added or removed, it may use a fixed sized array at the back end.

QUES 13). Application of queue data structure is___

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

Ans: d:

QUES 13). What is the time complexity for inserting/deleting an element in the array?

a) O(1)

b) O(n)

c) O(logn)

d) O(nlogn)

Ans: b:

QUES 14). All the other elements will have to be moved, hence O(n).

Which of the following don't use matrices?

a) Graph theory
b) Sorting numbers
c) In solving linear equations
d) Image processing

Ans: b:

QUES 15). Numbers uses arrays(1-D) for sorting

What will be the resulting array after rotating arr[]={a, b, c, d, e} by 2?

a) a, b, c, d, e

b) c, d, e, a, b

c) b, a, c, e, d

d) c, e, d, a, b

Ans: b

Explanation: When the given array is rotated by 2 then the resulting array will be

QUES16). Rotation 1: {b,c,d,e,a}

Rotation 2: {c,d,e,a,b}.

Thus, the final array is {c,d,e,a,b}

________ data structure is used to solve Tower of Hanoi

a) Tree

b) Heap

c) Priority queue

d) Stack

Ans: d:

The Tower of Hanoi involves moving of disks 'stacked' at one peg to another peg with respect to the size constraint. It is conveniently done using stacks and priority queues. Stack approach is widely used to solve Tower of Hanoi

QUES 17). Minimum number of queues needed to implement the priority queue?

a)1

b)2

c)3

d)4

Ans: b:

QUES 18). One queue is used for actual storing of data and another for storing priorities.

What is the pre-order traversal of the tree given below.

a) DBHEIAFCG

b) ABDEHICFG

c) DHIEBFGCA

d) ABCDEFGHI

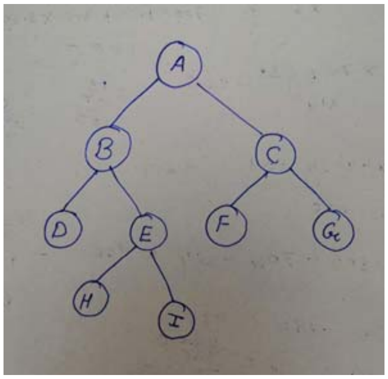
Ans: b:

pre-order steps:

1: Start from the Root.

2: Then, go to the Left Subtree.

3: Then, go to the Right Subtree.



QUES 19). Which of the following operation is faster in AVL trees?

a) Insertion

b) Deletion

c) Updation

d) Retrieval

Ans: d

QUES 20). Retrieval of element is fast in AVL tree. The number of steps required to find an item depends on the distance between the item and the root node. An AVL tree is balanced binary search tree in which the difference between the height of any node's left and right subtree is at most one. That's why retrieval is faster in AVL tree.

In a binary max heap containing n numbers, the smallest element can be found in time.

a)O(n)

b)O(logn)

c)O(nlogn)

d)O(1)

Ans: a:

QUES 21). The smallest element is always present at the leaf node. So the time complexity will be O(n).

In array representation of binary trees, what formula can be used to locate left child, if the index of the node is i?

a) 2i+3

b) 2i

c) 3i-1

d) 2i+1

Ans d:

QUES 21). If binary trees are represented in arrays, left children are located at indices 2i+1 and right children at 2i+2.

Which of the following traverals are used for prefix expression?

a) Post-order traversal

b) In-order traversal

c) Level-order traversal

d) Pre-order traversal

Ans: b.

QUES 22). The below code is for checking whether there will be a path from root to leaf with given sum. But something is missing in logic of the code, what is that?

```
checkSum(struct bin-treenode *root , int sum) :
 if(root==null)
   return sum as 0
 else :
   leftover_sum=sum-root_node-->value
   //missing
```

a) code for having recursive calls to either only left tree

b) code for having recursive calls to either only right tree

c) code for having recursive calls to either only left tree or right trees or to both subtrees depending on their existence

d) code for having recursive calls to either only left tree or right trees

Ans: c

QUES 23).if(left subtree and right subtree) then move to both subtrees

else if only left subtree then move to left subtree carrying leftover_sum parameter

else if only right subtree then move to right subtree carrying leftover_sum parameter.

Which traversal is used for sorting BST?

a)inorder

b)preorder

c)levelorder

d)postorder

Ans: a

QUES 24). Inorder traversal gives nodes in non-decreasing order. To get nodes of BST in non-increasing order, a variation of Inorder traversal where Inorder traversal's reversed can be used.

Consider the following data. The pre order traversal of a binary tree is 1, 2, 5, 3, 4. The in order traversal of the same binary tree is 2, 5, 1, 4, 3. The level order sequence for the binary tree is _________

a) 1, 3, 4, 2, 5

b) 1, 2, 3, 4, 5

c) 1, 2, 3, 5, 4

d) 4, 2, 5, 1, 3

Ans: C:

Steps:

1. First element in preorder will be the root of the tree, here it is 1

2. Now search the element 1 in inorder, all the elements left to 1 will construct the left subtree, and elements right to 1 will construct right subtree.

3. Now using step 1 and 2 ,recursively construct left sub tree and right sub tree and link both of them to the root.

4. Using the binary tree constructed above, find the level order traversal.

QUES 25). What should be the difference between left and right subtree of every node in a binary tree to make it balanced tree?

a) not more than 1

b) not more than 3

c) exactly 2

d) 0

Ans: a:

QUES 26). In a balanced binary tree the heights of two subtrees of every node never differ by more than 1.

5. Heap can be used as?

a) A decreasing order array

b) Normal Array

c) Priority queue

d) Stack

Ans: c

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.

QUES 27). What is a hash-map?

a) A structure used for storage

b) A structure used to implement stack and queue

c) A structure that maps values to keys

d) A structure that maps keys to values

Ans: d

QUES 28).A hash map is used to implement associative arrays which has a key-value pair, so the has table maps keys to values.

Assuming value of every weight to be greater than 20, in which of the following cases the shortest path of a directed weighted graph from 2 vertices will never change?

a) adding 20

b) multiplying by 20

c) subtracting 15

d) None

Ans: b:

In case of addition or subtraction the shortest path may change because the number of edges between different paths may be different, while in case of multiplication path wont change.

QUES 29).A connected planar graph having 12 vertices, 13 edges contains _____________ regions.

a) 1

b) 2

c) 3

d) 4

Ans: c:

QUES 30). By euler's formula the relation between vertices(n), edges(q) and regions(r) is given by n-q+r=2

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

a) Simple Graph

b) Complete Graph

c) Multi Graph

d) Regular Graph

Ans: b

QUES 31).a regular graph is a graph where each vertex has the same number of neighbors.

Let a graph G has n vertices, then the number of edges are_________

a)n

b)2n+1

c)2n-1/2

d)(n*(n-1))/2

Ans: d:

Number of ways in which every vertex can be connected to each other is nC2.

QUES 32). What is the maximum number of edges present in a simple directed graph with 1000 vertices if there exists no cycles in the graph?

a) 9999

b) 999

c) 99

d) 9

Ans: b)

If no cycle exists then the difference between the number of vertices and edges is 1.

What the function_graph will do?

QUES 33). void function_graph(int s)

```
{
    // Mark all the vertices as not visited(By default
    // set as false)
    boolean visited[] = new boolean[V];
    // Create a queue
    LinkedList<Integer> queue = new LinkedList<Integer>();
    // Mark the current node as visited and enqueue it
    visited[s]=true;
    queue.add(s);
    while (queue.size() != 0)
    {
        // Dequeue a vertex from queue and print it
        s = queue.poll();
        System.out.print(s+" ");
        // Get all adjacent vertices of the dequeued vertex s
        // If a adjacent has not been visited, then mark it
        // visited and enqueue it
        Iterator<Integer> i = adj[s].listIterator();
        while (i.hasNext())
        {
            int n = i.next();
            if (!visited[n])
            {
               visited[n] = true;

               queue.add(n);

          }

       }

    }

   }
```

a) DFS

b) Deleting a node in the given graph

c) BFS

d) None of these

Ans: c:

The given code is of BFS traversal of a graph.

source code: GeeksForGeeks

QUES 34). The time and space complexity of a recursive DFS algorithm is(E= number of edges and V= number of vertices)

a)O(V+E) and O(V)

b) O(V) and O(V+E)

c) O(V*E) and O(1)

d) O(logV) and O(log E)

Ans: a:

Time complexity: O(V + E), where V is the number of vertices and E is the number of edges in the graph.

Space Complexity: O(V). Since an extra visited array is needed of size V.

QUES 35) Suppose you cannot use arrays and linkedlist, then how many stacks is needed to implement a queue?

a) 1

b) 2

c) 3

d) 4

Ans: b:

A queue can be implemented using two stacks

Steps:

Keep 2 stacks, S1 and S2

Enqueue:

Push the new element onto S1

Dequeue:

If S2 is empty, refill it by popping each element from S1 and pushing it onto S2

Pop and return the top element from S2.