MSQs: Multiple Select Questions

Each question can have one or more correct answers. No negative marking.

1. What will be the output of the following code:

```
String s1=”DSA”;
int a=5;
int arr[]=new int[100];
String s2=”DSA”;
int b=5;
Int arr2[]=arr;

s2=”JAVA”;
b=100;
arr2[50]=75;

System.out.print(s1+” “+arr[51]+arr[50]+” “+s2+” “+a+” “+b);
```

A. DSA 0 75 JAVA 5 100
B. DSA 75 s2 5 100
C. JAVA 75 JAVA 5 100
D. JAVA 075 JAVA 5 100

2.

```
class Node{
        int a, b;
        Node(int a,int b) {
                this.a = a; this.b = b;
        }
}
public static void main(String[] args) {
        Node x=new Node(2,3);
        int arr[]=new arr[2];
}
```

Do arr and x occupy the same space ?

A. Yes B. No C. Can’t say

3.
T(n,m) = 2T(n/2,m/2) + O(n)
T(1,1)=T(1,0)=T(0,1) = 1
What is the runtime given by the above recurrence?

A. O(nlog(n*m)) B. O(nlog(min(n,m))) C O(nlog(max(n,m))) D O(nlog(n + m))

4.
matrix is 2d array of dimensions nxn

```
int fun(int[][] matrix, int x, int y){
         int X = matrix.length;
         int Y = matrix[0].length;
        if(x<0 || y<0 || x>=matrix.length || y>=matrix[0].length)
                return Integer.MAX_VALUE:
         if(x>=X || y>=Y) return Integer.MAX_VALUE;
        if(x==X-1 && y==Y-1) return 0;
        return matrix[x][y] + fun(matrix,x+1,y) + fun(matrix,x,y+1);
}
```

What is the time complexity of the given function ? (Assume X and Y are known to you)

A. O(n^2) B. O(n*log(n)) C. O(2^n) D. O(n*n!)

5.

What is the output of the given code:

```
int  fn(int n){
        if(n==0) System.out.println(n);
        int a= fn(n+1) + fn(n+2);
        return a ;
}
public static void main(String[] args){
System.out.println(fn(5));
}
```

A. Function will never stop executing
B. 5
C. Compilation Error
D. Runtime Error

6. What does the given code compute:

```
        int fun1(int n){
                if(n == 1)
                return 0;
                return 1 + fun1(n/2) + fun1(n/2);
        }
```

Note: n>1 when the function is called.

A. n   B. n-1   C.log(n)   D. 2log(n)

7. Is it possible to implement the First In First Out Principle using stack(s) only ?   (1)

A. Yes
B. No, FIFO can only be implemented using a queue
C. None of the above

8. Which sorting technique is the best to sort a partially sorted array?

A. Selection Sort   B. Insertion Sort   C.Bubble sort

9. What will be the running time of Quicksort when the input array is already sorted? (2)

A. O(1) B. O(nlogn) C.O(n) D.O(n^2)

10. What is the best case time complexity to sort an array of size containing repetitions of 0 ,1 and 2. (3)

A. O(n^2) B. O(nlogn) C. O(n) D.O(sqrt(n))

sqrt(n) is the square root of n.

11. Which of the following points is/are false about Linked List data structure when it is compared with array :- (3)

A. It is easier to delete and insert in Linked List
B. Memory usage in linked list is less for same size
C. Linked list are faster than arrays
D. Random access of elements could be possible in linked list

12. Which of the following operations is performed more efficiently by a doubly linked list than by a singly linked list?

A. Deleting a node whose location is given
B. Swapping 2 adjacent elements
C. Concatenating 2 sorted list to make resultant list sorted
D. Reversing a list

13. Which of the following options have asymptotic time complexity O(log(n))

```
A. for(int j = n; j> = 1; j /= 2) {
        for(int i = 0; i < j ; i *= 2){
        System.out.println("hello world");
      }
  }

B. int[] val = new int[n] //initialized as -1
  int fib(int n) {
      if(n == 0 || n == 1) val[n] = 1;
      if(val[n] != -1)
```

```
                val[n] = fib(n - 1) + fib(n - 2);
        return val[n];
  }

C. void met(int n) {
        if (n == 0) return;
        for(int i = 0; i < n; i += 2) {
                met(n-1);
        }
  }


D. int fun(int x, int y){
        if(y == 0) return x;
                return fun(y, x % y);
  }
```

Subjective Questions [70 Marks]

Question 1. [10 marks]

Your friend gave you two sorted stacks (Min on top) , when you asked for only one. You don't have enough space to store two stack pointers with you so you ask him to convert two stacks to one, one stack which contains all the elements of both stacks that too in sorted order (Min on top). Your friend is not very good with DSA problems, so you decide to help him by writing a function which does the job for him. But since he only knows stacks and nothing else, you are allowed to use only stack operations such as pop(), push(), size() and top(). No other data structures such as arrays are not allowed. You are allowed to use extra stacks. Complete the given function which does the same.

```
public Stack mergeStacks (Stack A , Stack B){
}
```

Question 2. [3+3+7 = 13 marks]

Anurag has grown so tired of studying DSA that he decides to drop off from college and follow his passion, which is to become a chef! He already has funds to open his restaurant and hire a crew. He wants to open a buffet restaurant so that he doesn't have to take so many orders in the beginning. Anurag wants to attract as many people as possible and so he keeps his restaurant for both vegetarians as well as non vegetarians. But he would also like to identify those people in the restaurant so that he could provide his services more efficiently. So he decides to have two types of plate- **veg** and **non-veg**. Obvious enough, vegetarians will take the former and non vegetarians will take the latter respectively.

Just like in a buffet, customers will stand in a line and take their plates and head towards their favourite dishes. Anurag has studied engineering way too much to make this process as efficient as possible but he's also weak in DSA. He asks for your help to solve his queries:

a) He wants to develop an efficient way of taking plates.
   Write the data structure required for this and why? **[1+2; correct answer and explanation]**
b) He wants to develop an efficient way of distributing those plates to the guests. The distribution algorithm should be fair for everyone on a first come first serve basis. Write the data structure required for this and why? **[1+2; correct answer and explanation]**
c) Now the plates as well as people are randomly set in your data structure you've selected. Anurag would want the people who are non-vegetarians to take non-veg plates and the number of non-vegetarians and non-veg plates is equal. Similarly for vegetarians.
   Explain in detail a system where everyone gets their plate, without breaking the queue. No code is required.

   {let suppose 2 DS are arrays then plates will be [n,v,v,n,v] and people will be [n,n,v,v,v]}

Question 3. [3+7 = 10 marks]

Studying in an online setting is very tiring.To make up for this, the TAs have offered you chocolates. Each chocolate has a sweet value represented by an integer. Naturally, you want to have as much sweet value as possible. The sweet values are stored in an array. The TAs too want to have chocolates, so they restrict you from choosing adjacent chocolates. Given the conditions

1. Write the recurrence relation to this problem.
2. Using the recurrence, write pseudo code for finding the maximum sweet value.

Question 4. [3+3 = 6 marks]

Abhinav thinks he is very smart . Being so smart, he starts to invent his own language.Since he is a DSA TA, all he does is think about Linked Lists the entire time. In his new language, an 'operation' is defined as rotating a linked list once. Like him, the list is also single (It is a singly linked list). Rotation means removing the last element and adding it before the first element. No need to write any code.

1) If you're given the head node as well as the last node, write the logic to achieve an "operation". What is the time complexity of the operation? **[1+2; for correct answer and explanation]**
2) Abhinav is busy studying for his internships, he has asked his students to perform an 'operation' a given linked list k times and return the head of the

modified list. What is the time complexity? **[1+2; for correct answer and explanation]**

Question 5. [(2+3)+3+3 = 11 marks]

Jasmine is really a big fan of wasting her time watching domino videos on youtube. She always gets fascinated by the amount of effort put in by people across the globe to come up with such interesting domino sets. One day, she decides to create her own domino. Since there's a lockdown, she cannot buy a domino set so she orders a chain reaction domino set **arr** of size **n** from amazon. Upon opening the set, she started to place them in random order in a straight line but then she realized that not all the dominoes were falling. She noticed the size of some **ith** domino was greater than some **jth** domino (where **j>i**). She wants to find how many such pairs **(arr[i],arr[j])** exist so that she could count the number of times she made an error while placing.
(arr is an array which stores the size of each domino)

1. Which sorting algorithm will help Jasmine find the pairs optimally and why?
2. Write the recurrence relation for sorting algorithm in part 1.
3. Solve the recurrence relation in part 2 and find the time complexity.

Question 6. [10 marks]

Abhi is very competitive in nature. So when his professor gave him circular singly linked lists he wanted to make a one concatenated circular singly linked list with it, that too in O(1) time. He wrote a function which took two circular singly linked lists as arguments and returned the result. Your task is to complete the function he wrote.

```
Class ListNode{
        int data;
        ListNode nxt;
}
ListNode concatenate( ListNode A, ListNode B){
}
```

Question 7. [10 marks]

Alice and Bob are very good friends. But after studying recursion for 4 nights straight Bob has developed a phobia for it. He wrote the following code to find whether a path from (i, j) to (n-1, n-1) exists or not in a square matrix with obstacles. (From (i, j) you can only go to (i+1, j) and (i, j+1)). Obstacles are represented by mat[i][j] =0. Normal cells are mat[i][j] =1.

```
boolean find_path (int mat[][], int i, int j, int n){
        if( i >= n || j >= n){    return false;
                }
        if( mat[i][j] == 0) {     return false;   }
        if( i == n-1 && j ==n-1) {     return true;    }
        if( find_path(mat, i+1, j, n)) {         return true;}
        if( find_path(mat, i, j+1, n)) {         return true;}
        return false;
}
```

But Bob can't face a recursive code. And Alice is desperate to help Bob, so she decides to rewrite this without using recursion. Kindly help Alice by completing the function she wrote to perform the same task. Also Name the data structure you would use.

```
boolean find_path( int mat[][], int i, int j, int n){
        Stack<int[]> stack = new Stack<>();
        _________________                     // Line 1
        while(stack.isEmpty() == false){
                int ele[] = stack.pop();
                // Blank 1 and Blank 2
                if( ele[0] == ____  && ele[1] == ____ )
                        { return true; }
                if(ele[0]+1 < n && mat [ ele[0] +1] [ele[1]] == ___ )                //Blank 3
                        { ________________________}               // Line 2
                if(ele[1]+1 < n && mat[ele[0]][ele[1]+1]== ____ )          // Blank 4
                        { ________________________ }               // Line 3

        }
        return false;
}
```