Name: Anvil Gupta Roll No. 22114009. SET D

Indian Institute of Technology Roorkee (IIT-R)

Spring Semester (2022-2023), Mid Term Examination
Sub: Data Structures (CSN/DA-102)
Class: B.Tech./B.Sc. I year
Time: 1 hours 30 minutes (1:45 PM to 03:15 PM)
Date: 24/04/2023.

Max. Marks: 25

Instructions:

1. **Both Sections A and B are compulsory to attempt.**
2. **There is NO NEGATIVE marking for Section A.**
3. **In Section A, Questions 1 - 11 are multiple-choice questions. Each question has four options out of which one or more options are correct. Full marks will be awarded only if you mark all the correct options.**

Section A [Max. Marks: 15]

Q1. An implementation of a queue Q, using two stacks, S1 and S2, is given below:

```
void insert(Q, X){
  push(S1, X);}
void delete(Q){
 if(stack-empty(S1)) then {
   print("Q is empty");
   return; }
else while (!(stack-empty(S1))){
   X = pop(S1);
   push(S2, X); }
 X = pop(S2);}
```

Let n insert and m(≤ n) delete operations be performed in an arbitrary order on an empty queue. Let x and y be the number of push and pop operations performed, respectively in the process. Which one of the following is/are true for all m and n? **[1 Mark]**

a) $n+m \le x \le 2n$ and $2m \le y \le 2n$
b) $n+m \le x \le 2n$ and $2m \le y \le n+m$
c) $2m \le x \le 2n$ and $2m \le y \le 2n$
d) $2m \le x \le 2n$ and $2m \le y \le n+m$

Q2. A single array A[1...MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (top1 < top2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for "stack full" is/are _____. **[1 Mark]**

a) top 1 + top 2 = MAXSIZE
b) (top 1 = MAXSIZE/2) AND (top 2 = MAXSIZE/2 + 1)
c) top 1 = top 2 – 1
d) (top 1 = MAXSIZE/2) or (top 2 = MAXSIZE)

Q3. Match the following and select the correct answer. **[2 Marks]**

Column A	Column B

1) `int a = 0, b = 0;`
`for (i = 0; i < N; i++) {`
`  a = a + rand();}`
`for (j = 0; j < N; j++) {`
`  b = b + rand();}`
	P) O(N*N)

2) `int a = 0;`	Q) O(nLogn)

Name: Roll No.

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

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

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

a) 1 – R, 2 – P, 3 – S, 4 – Q
b) 1 – P, 2 – R, 3 – Q, 4 – S
c) 1 – P, 2 – R, 3 – S, 4 – Q
d) 1 – R, 2 – P, 3 – Q, 4 – S

Q4. Suppose we need to sort an array of eight integers using the quicksort algorithm, and we just finished the partitioning with the array: {2, 5, 1, 7, 9, 12, 11, 10}. Which of the following is/are TRUE? **[1 Mark]**

a) The pivot could be 9.
b) The pivot could be 7.
c) The pivot could be 5.
d) The pivot could be 11.

Q5. The result evaluating the postfix expression 10 10 20 + * 5 30 15 / + 8 - 2 + ^ is: _____? **[1 Mark]**

a) 300 b) 301 c) 308 d) 302

Q6. What is/are the correct output/s of the following code? **[2 Marks]**

```
#include<stdio.h>
void mte(int n, int a, int b)
{
  if (n <= 0) return;
  mte(n-1, a, b+n);
  printf("%d %d %d\n",n,a,b);
  mte(n-1, b, a+n); }
int main()
{
  mte(3,4,5);
  return 0; }
```

a)
1 4 10
2 4 8
1 8 6
1 5 9
2 5 7
1 7 7

b)
1 4 10
2 4 8
1 8 6
3 4 5
1 5 9
2 5 7
1 7 7
3 4 5

c)
1 4 10
2 4 8
1 8 6
3 4 5
3 4 5

d)
1 4 10
1 5 9
2 5 7
1 7 7

Q7. Suppose T(n) = 2T (n/2) + n, T(0) = T(1) = 1 Which of the following is/are FALSE? **[1 Mark]**

a) T(n) = o(nlog n)
b) T(n) = $O(n^2)$

c) $T(n) = \theta(n \log n)$

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

Q8. Let A be a square matrix of size n x n. Consider the following program. What is/are the expected output/s? **[1 Mark]**

```
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) Adding 100 to the upper diagonal elements and subtracting 100 from diagonal elements of A
b) Transpose of matrix A
c) The matrix A itself
d) None of the above

Q9. An array A consists of n integers in locations A[0], A[1] ....A[n-1]. It is required to shift the elements of the array cyclically to the left by k places, where 1 <= k <= (n-1). An incomplete algorithm for doing this in linear time, without using another array is given below. Complete the algorithm by filling in the blanks. Assume all the variables are suitably declared. **[2 Marks]**

```
min = n; i = 0;
while (_____) {
     temp = A[i]; j = i;
     while (_____) {
     A[j] = _____
     j= (j + k) mod n ;
     If ( j< min ) then
         min = j;
}}
A[(n + i – k) mod n] = _____
i = _____
```

a) i < min; j!= (n+i)mod n; A[j + k]; temp; i + 1;
b) i > min; j!= (n+i)mod n; A[j + k]; temp; i + 1 ;
c) i < min; j!= (n+i-k)mod n; A[(j + k)mod n]; temp; i + 1;
d) i > min; j!= (n+i+k)mod n; A[(j + k)]; temp; i + 1;

Q10. Consider the following function that takes reference to the head of a Doubly Linked List as a parameter. Assume that a node of doubly linked list has the previous pointer as prev and next pointer as next.

```
void fun(struct node **head_ref)
{
    struct node *temp = NULL;
    struct node *current = *head_ref;

    while (current !=  NULL)
    {
        temp = current->prev;
        current->prev = current->next;
        current->next = temp;
        current = current->prev;
```

Name: Roll No.

```
    }

    if(temp != NULL )
        *head_ref = temp->prev;
}
```

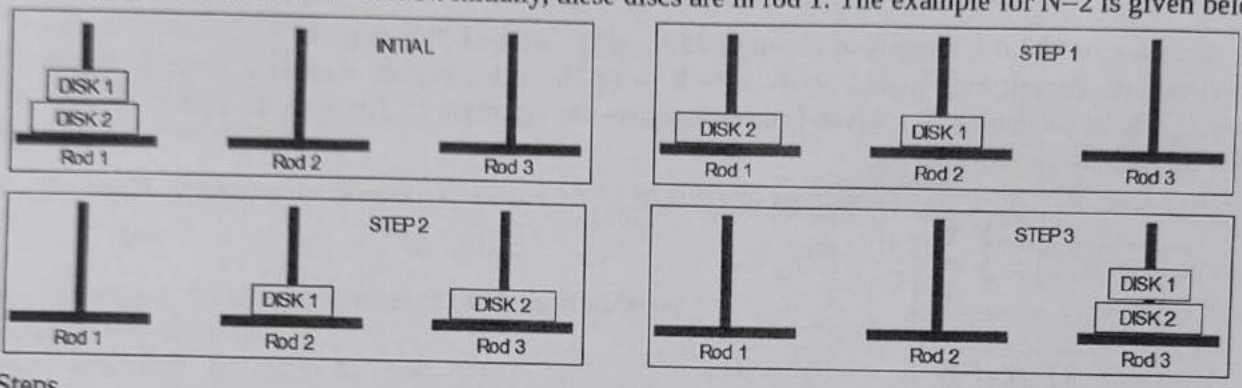
Assume that the reference of the head of the following doubly linked list is passed to above function
1 <--> 2 <--> 3 <--> 4 <--> 5 <-->6.
What option/s should be the modified linked list after the function call? **[1 Mark]**

a) 5 <--> 4 <--> 3 <--> 2 <--> 1 <-->6
b) 2 <--> 1 <--> 4 <--> 3 <--> 6 <-->5
c) 6 <--> 5 <--> 4 <--> 3 <--> 1 <--> 2
d) 6 <--> 5 <--> 4 <--> 3 <--> 2 <--> 1

Q11. The tower of Hanoi is a famous puzzle where we have three rods and N disks. The objective of the puzzle is to move the entire stack to another rod. The rules to transfer the disks are as follows:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack. In other words, a disk can only be moved if it is the uppermost disk on a stack.
- No larger disk may be placed on top of a smaller disk.

You are given the number of discs N. Initially, these discs are in rod 1. The example for N=2 is given below:



Steps

1. move disk 1 from rod 1 to rod 2
2. move disk 2 from rod 1 to rod 3
3. move disk 1 from rod 2 to rod 3

Which of the following is/are the correct pseudocode/s to transfer the disks from rod 1 to rod 3? **[2 Marks]**

a)

```
#include <bits/stdc++.h>
using namespace std;
void towerOfHanoi(int n, char from_rod, char to_rod,cha  aux_rod)
{
  if (n == 0) {
       return;}
  towerOfHanoi(n/2, from_rod, aux_rod, to_rod);
  cout << "Move disk " << n << " from rod " << from_rod
       << " to rod " << to_rod << endl;
  towerOfHanoi(n/2, from_rod, aux_rod, to_rod);
}
int main()
{
  int N = 3;          // A, B and C are names of rods
  towerOfHanoi(N, 'A', 'C', 'B');
  return 0;}
```

b)

```
#include <bits/stdc++.h>
using namespace std;
void towerOfHanoi(int n, char from_rod, char to_rod,cha  aux_rod)
{
  if (n == 0) {
       return;}
  towerOfHanoi(n - 1, from_rod, aux_rod, to_rod);
  cout << "Move disk " << n << " from rod " << from_rod
       << " to rod " << to_rod << endl;
  towerOfHanoi(n - 1, aux_rod, to_rod, from_rod);
}
int main()
{
  int N = 3;           // A, B and C are names of rods
  towerOfHanoi(N, 'A', 'C', 'B');
  return 0;}
```

c)

```
#include <bits/stdc++.h>
using namespace std;
void towerOfHanoi(int n, char from_rod, char to_rod,cha  aux_rod)
{
  if (n == 0) {
       return;}
  towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
  cout << "Move disk " << n << " from rod " << from_rod
       << " to rod " << to_rod << endl;
  towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
}
int main()
{
  int N = 3;        // A, B and C are names of rods
  towerOfHanoi(N, 'A', 'C', 'B');
  return 0;}
```

d)

```
#include <bits/stdc++.h>
using namespace std;
void towerOfHanoi(int n, char from_rod, char to_rod,cha  aux_rod)
{
  if (n == 0) {
       return;}
  towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
  cout << "Move disk " << n << " from rod " << from_rod
       << " to rod " << to_rod << endl;
  towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
}
int main()
{
  int N = 3;        // A, B and C are names of rods
  towerOfHanoi(N, 'A', 'C', 'B');
  return 0;}
```

Please turn over for Section B

Name: Roll No. SET D

Section B

[Max. Marks: 10]

Q1. Refer to Question 11(from Section A), If N=3, then what is the optimal number of steps required to transfer the disk from rod 1 to rod 3. Illustrate all the steps in an order. **[Marks: 02]**

Q2. Find the multiplication of the following sparse matrix using Sparse Matrix Multiplication algorithm.

[Marks: 03]

Matrix 1: (4x4)

Row	Column	Value
1	2	10
1	4	12
3	3	5
4	1	15
4	2	12

Matrix 2: (4X4)

Row	Column	Value
1	3	8
2	4	23
3	3	9
4	1	20
4	2	25

Q3. What is the output of the following? **[Marks: 03]**

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 4
int main()
{
    int q = -1, inp_array[SIZE],element;
    void U(int x)
{    if (q == SIZE - 1)
    { printf("Operation cannot be performed");}
    else
    {   q = q + 1;
        inp_array[q] = x;}}
void P()
{ if (q == -1)
    {printf("Operation cannot be performed");    }
    else
    {  element = inp_array[q];
       q = q - 1;
        printf("%d", element);
    }}
    U(1);U(2);P();U(1);U(2);P();P();P();U(2);P();
}
```

Q4. Consider a singular linked list of N nodes (N is an odd number). Write a pseudo code to find out the middle element of the linked list without using any counter variable. Time complexity of the pseudocode is strictly bound to O(N). **[Marks: 02]**

[***Hint:*** *you may use at most two pointers except head.*]

Good Luck!