Birla Institute of Technology & Science, Pilani
Work-Integrated Learning Programmes Division
Second Semester 2020-2021

EC-3 Regular Comprehensive Examination

Course Title	:	Data Structures and Algorithms Design
Course No	:	SS ZG519
Total	:	50 marks
Nature of Exam	:	Open Book
Duration	:	2 hours
Date	:	01/05/2021 (FN)

No. of Pages = 2
No. of Questions = 11

Note:

1. **Please follow all the Instructions to Candidates given on the cover page of the answer book.**
2. **All parts of a question should be answered consecutively. Each answer should start from a fresh page.**
3. **Assumptions made if any, should be stated clearly at the beginning of your answer.**

1. Find out what is the output of the following algorithm if the inputs are two positive natural numbers. Prove that indeed algorithm is correctly computes that. What is the running time of the algorithm? $(5 Marks)$

```
Input: M and N
$L = 0$
repeat {
   if M is odd then L = L +N;
   M= M div 2;
   N= N+N;
} until (M==1)
return $L$
```

2. Suppose $f(n) \in O(g(n))$. Prove or disprove $2^{f(n)} \in O(2^{g(n)})$. $(2 Marks)$

3. Prove that inorder traversal of binary search tree returns elements stored in it's internal nodes in ascending order. $(4 Marks)$

4. What is the probability of the following sorting algorithm terminates? What is the expected running time of the algorithm? Justify your answer. $(5 Marks)$

```
Input: Array $A$ consisting $n$ distinct integers
Output: Sorted Array $A$
while Sorted(A) is not true
   Shuffle(A)
endwhile
```

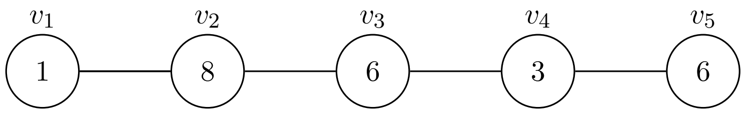
Sorted method will return true if the given array is sorted in $O(n)$ time and *Shuffle* method will give a random permutation of the given array in $O(n)$ time.

5. Find the minimum number of scalar multiplications and an optimal parenthesization of a matrix-chain product whose sequence of dimensions is $< 5, 10, 3, 12, 5, 50, 6 >$. There are 6 matrices in this chain and the first matrix has dimension 5×10, second matrix has dimension 10×3, ... and the sixth matrix has dimension 50×6. $(4 Marks)$

6. Let A be an array of n elements. An element e is a majority element in A if it appears strictly more than n/2 times in A. Write a divide and conquer algorithm to check whether

a given array has majority element or not. Running time of your algorithm should be O(nlogn) in the worst case. There is no order relation between elements and so equality is the only comparison operator allowed. $(7 Marks)$

7. A set S of nodes in a graph is independent if no two nodes in S are joined by an edge in that graph. Consider a path $P = (V, E)$ and each vertex v_i in the path is associated with a positive integer w_i. We want to find an independent set in P whose total weight is as large as possible. For example, the path given below has an independent set with weight 14 (vertices v_2 and v_5).



Give examples to show that the following two algorithms do not always find optimal solutions. $(4 Marks)$

Algo: "heaviest-first" greedy algorithm

```
Start with S equal to empty set
While some node remains in G
   Pick a node v of maximum weight
   Add v to S
   Delete v and its neighbors from G
EndWhile
Return S
```

Algo: "Odd-Even" greedy algorithm

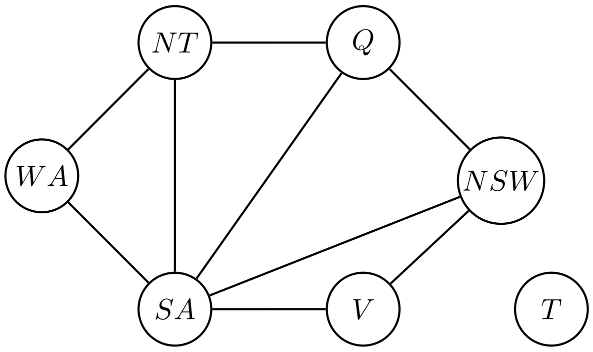
```
Let S1 be the set of all vertices
     with odd indices
Let S2 be the set of all vertices
     with even indices
Determine which of S1 or S2 has greater
     total weight and return that set
```

8. Suppose that you are given an $n \times n$ chess board and a token. You must move the token from the bottom edge of the board to the top edge of the board according to the following rule. At each step you may move the token to one of three squares:

- the square immediately above,
- the square that is one up and one to the left (but only if the token is not already in the leftmost column),
- the square that is one up and one to the right (but only if the token is not already in the rightmost column).

Each time you move from square x to square y, you receive p(x, y) dollars. You are given p(x, y) for all pairs (x, y) for which a move from x to y is legal. Do not assume that p(x, y) is positive. Give an algorithm that figures out the set of moves that will move the token from somewhere along the bottom edge to somewhere along the top edge while gathering as many dollars as possible. Your algorithm is free to pick any square along the bottom edge as a starting point and any square along the top edge as a destination in order to maximize the number of dollars gathered along the way. What is the running time of your algorithm? $(8 Marks)$

9. Construct the adjacency matrix and adjacency list for the following graph. $(4 Marks)$



10. Give an example of an n-vertex simple graph G that causes Dijkstra's algorithm to run in $\Omega(n^2 logn)$ time when its implemented with a heap for the priority queue. $(7 Marks)$