LOYOLA COLLEGE (AUTONOMOUS), CHENNAI – 600 034

M.Sc. DEGREE EXAMINATION – **MATHEMATICS**

FIRST SEMESTER – **NOVEMBER 2019**

PMT 1504 – COMPUTER ALGORITHMS

Date: 07-11-2019 Dept. No. Max. : 100 Marks

Time: 01:00-04:00

Answer ALL the Questions:

1. a) Write an algorithm to find the trace and transpose for an $n \times n$ matrix. Also find the time complexity. **(5)**

OR

b) Find the time complexity for Algorithm Heapify. **(5)**

c) Define a stack and a circular queue. State procedures to add and delete items from a stack and a circular queue. **(15)**

OR

d) State Algorithm HeapSort to sort numbers in an array. Simulate it on A(1 : 7) = (16, 9, 23, 6, 11, 18, 5). **(15)**

2. a) Give the recurrence relation of complexity for divide-and-conquer algorithm. Solve the recurrence relation when $a = 5$, $b = 4$, $f(n) = cn^2$. **(5)**

OR

b) Find the worst-case time for binary search algorithm. **(5)**

c) State Algorithm QuickSort. Simulate it on $A(1 : 8) = (40, 60, 100, 20, 70, 120, 10, 50)$. **(15)**

OR

d) State algorithm MergeSort. Simulate it on $A(1 : 8) = (45, 32, 56, 78, 81, 3, 23, 11)$. Draw tree of calls of MergeSort and Merge when $n = 8$. **(15)**

3. a) Explain the knapsack problem with an example. **(5)**

OR

b) Let J be a set of k jobs and $\pm = i_1 i_2 \dots i_k$ a permutation of jobs in J such that $d_{i_1} \leq d_{i_1} \leq \cdots \leq d_{i_k}$. Prove that J is a feasible solution if and only if the jobs in J can be processed in the order $\pm$ without violating the deadline. **(5)**

c) With usual notations, prove that if $l_1 \leq l_2 \leq \cdots \leq l_n$, the ordering $i_j = j$, $1 \leq j \leq n$ minimizes $\sum_{i=1}^{n} \sum_{j=1}^{n} l_{i_j}$ overall possible permutations of i_j. State Algorithm Store and simulate it when $n = 15$, $m = 3$ and the length of the programs are 42, 23, 11, 5, 33, 15, 8, 45, 20, 35, 40, 2, 13, 25, 6. **(15)**

OR

d) State an algorithm which generates a two-way merge tree for n files with weight values $(q_1, q_2 \dots q_n)$, $n \geq 1$ and prove that it generates an optimal two-way merge tree. Simulate the algorithm on 11 files of length 80, 70, 56, 90, 5, 44, 35, 63, 15, 25, 40 and find the weighted external path length of the two-way merge tree representing a merge pattern. **(15)**

4. a) State an algorithm that present a recursive formulation of backtracking technique. **(5)**

OR

b) Write an algorithm to find all *m*-colorings of a graph. **(5)**

c) Explain *n*-queen's problem and give an algorithm to place the k^{th} queen in column *i* of a $n \times n$ chessboard and a backtracking algorithm to solve the *n*-queens problem. **(15)**

OR

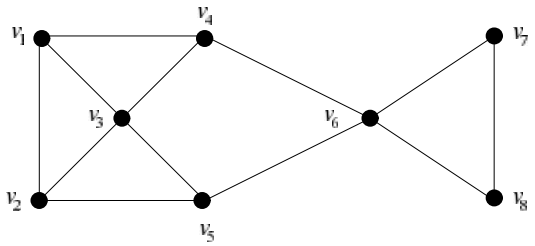
d) State algorithm SumOfSub. Let $w = \{4, 6, 8, 9, 10, 15\}$ and $m = 25$. Find all possible subsets of *w* that sums to *m* using SumOfSub and draw the portion of state space tree generated by SumOfSub. **(15)**

5. a) Write a nondeterministic pseudocode for clique decision problem. **(5)**

OR

b) Explain the terms 'Polynomially solvable' and 'NP-complete'. State Cook's theorem. **(5)**

c) Define node cover for a graph. Find the size of node cover for the following graph.



Prove that the node cover decision problem is NP-Complete. **(15)**

OR

d) Explain the maximum clique problem with an example. Prove that clique decision problem is NP-complete. **(15)**

@@@@@@