NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSPC42- DESIGN AND ANALYSIS OF ALGORITHMS

Programme: B.TECH	Final Assessment	Session: JAN/2024
Date: 21.05.2024	Duration: 3 Hours	Total Marks: 40

Answer all the questions

1. a) Find the time complexities of the following recurrences using substitution method or recursion tree method. (4 M) (CO1)

(i) $T(n) = 2\,T(\frac{n}{2}) + n\log n$ (ii) $T(n) = T(\frac{n}{2}) + T(\frac{2n}{3}) + n^2$

b) Write an algorithm for finding maximum and minimum elements in an array using Tournament method. Analyze its time complexity. How many minimum number of comparisons are required to find minimum and maximum elements in the given array with size 100 using Tournament method. (4 M) (CO1)

2. a) Consider the making change problem. The input to this problem is an integer M. The output should be the minimum number of coins to make M rupees of change. Assume the available coins are 1,5,10,20,25,50 rupees. Assume that we have an unlimited number of coins of each type. Provide a greedy algorithm with time complexity analysis for finding the minimum number of coins. Does this greedy algorithm produce the optimal solution or not? Justify your answer with suitable examples. (3 M) (CO1, CO2)

b) Write an algorithm for job sequence scheduling and analyze its time complexity. Find the sequence of jobs for the following example. (3 M) (CO1, CO2)

Jobs: A, B, C, D, E, F, G, H, I, J. **Deadlines:** 4, 2, 3, 2, 3, 4, 2, 4, 2, 1

Profits: 110, 10, 80, 20, 130, 160, 80, 130, 150, 100

3. a) Consider the following puzzle. There is a row of n chairs and two types of people: M for mathematicians and P for poets. You want to assign one person to each seat but you can never seat two mathematicians together or they will start talking about mathematics and everyone else in the room will get bored. For example, if $n = 3$, the following are some valid seatings: PPP, MPM, and PPM. However, the following is an invalid seating: MMP. Let $f(n)$ be the number of valid seatings when there are n chairs in a row.

i. Write a recurrence relation for f(n). Analyze the time complexity. (4 M) (CO1, CO4)

ii. Design an efficient algorithm to compute f(n).

b) Why do we use an optimal binary search tree? Determine the cost and structure of an optimal binary search tree for a set of n = 5 keys with the following probabilities: (4 M) (CO4)

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

c) You are given an exam with questions numbered 1, 2, 3, ..., n. Each question i is worth p_i points. You must answer the questions in order, but you may choose to skip some questions. The reason you might choose to do this is that even though you can solve any individual question i and obtain the p_i points, some questions are so frustrating that after solving them you will be unable to solve any of the following f_i questions. Suppose that you are given the p_i and f_i values for all the questions as input. Devise an efficient algorithm for choosing set of questions to answer that maximizes your total points, and compute its asymptotic worst case running time as a function of n. (4 M) (CO1, CO4)

d) Given an array of n integers, finding the length of the longest subsequence in a given array of integers such that all elements of the subsequence are sorted in strictly ascending order is called the Longest Increasing Subsequence (LIS) problem. Give a recursive formula for optimal value of LIS. Is it following overlapping subproblems property, justify your answer. Write optimal algorithm for the problem and analyze its time complexity. (4 M) (CO1, CO4)

4. a) Discuss Randomized algorithms. Write algorithm for Randomized quick sort and prove that its time complexity is O(nlogn) (4 M) (CO1, CO4)

5. a) How to prove the following statements (3 M) (CO4)
(i) $P = NP$ (ii) $P \neq NP$

b) State whether the following statements are True or False with proper explanations.
(Note: NPC means NP Complete) (3 M) (CO4)

(i) If $P = NP$, then $NPC \subseteq P$

(ii) If $NPC \subseteq P$, then $P = NP$.

(iii) If $P \neq NP$, then $NPC \cap P = \emptyset$

(iv) If $P \neq NP$, then NP-Hard = NP

(v) Let A be a problem that belongs to the class NP, then there is no polynomial time algorithm for problem A

(vi) Let A be a problem that belongs to class NP. If problem A can be solved deterministically in polynomial time, then $P = NP$

*** Best Wishes***