Roll No.: _

Amrita Vishwa Vidyapeetham

Amrita School of Computing, Coimbatore

B.Tech Degree Examinations – January 2023

Fifth Semester

Computer Science and Engineering

19CSE302 Design and Analysis of Algorithms

Duration: Three hours Maximum: 100 Marks

Course Outcomes (COs):

CO	Course Outcomes
CO01	Evaluate the correctness and analyze complexity of algorithms.
CO02	Understand and implement various algorithmic design techniques and solve classical problems.
CO03	Design solutions for real world problems by identifying, applying and implementing appropriate design techniques..
CO04	Design solutions for real world problem by mapping to classical problems.
CO05	Analyze the impact of various implementation choices on the algorithm complexity .

Answer all questions

1) Consider the brute-force algorithm for searching a pattern in the given text. How many comparisons does the algorithm make to find all occurrences of the pattern in the text?
   A. ‘ana’ in the text ‘bananas’?
   B. ‘ana’ the text ‘anaeatsabanana’?
   C. ‘ana’ the text ‘anabanananananann’?
   D. How many comparisons does it take for the Boyer- Moore algorithm for all the test cases above [10][CO02][BTL Level 2]
2) Let's suppose algebra, calculus, operating system, data structures, statistics, computer architecture, and physics are the six courses offered in our college. And let’s say that the following pairs have common students: “algebra and statistics”, "algebra and calculus", “statistics and physics”, “computer architecture and operating systems”, and “operating systems and data structures”. If the exam office schedules the above pair of exams in the same slot, then students taking both courses have to miss at least one exam. Using the backtracking approach, how can we schedule exams in a

R

minimum no of days so that courses having common students are not held on the same day? Show the supporting state space tree. [10][CO03][BTL Level 3][Hint: Graph colouring Problem]

3) Alex is trading in the share market; he is concerned with a product. In advance, he knows the prices of this product over the next 'n' days. The price of the i^{th} day is denoted as P_i. Assume that the price of the product Alex is interested in is stored in an array P of size n. You have to devise an algorithm which outputs two days from this array. On the first day, Alex can buy the product(which should be the minimum price) and on the second day, outputted by the algorithm, Alex will sell it(which should be the maximum price) so that Alex makes maximum profit when he buys and sells in these two days. In other words, we need to find the maximum value of $(P_j - P_i)$ such that $j >= i$.
   A. Write a brute force algorithm for finding two days such that profit maximises and analyse the time complexity of the algorithm.
   B. Can you do better than the brute force algorithm? Explain the algorithm and give the time complexity of the algorithm.
   [10][CO04][BTL Level 4]

4) Answer the following
   A. Given two sets A and B of integers and an integer x, give an efficient algorithm and analyse the time complexity to determine whether x = a + b for some 'a' belongs to A, and 'b' belongs to B
      Input: Integer Set **A**, Integer Set B and Integer x
      Output: Pair of two numbers from each of the sets, such that the sum is x
   B. Suppose we have an 'N' array of 'N' elements. You are told that each $i^{th}(i < N)$ *array* is sorted in increasing order. Also, each k^{th} element across the 'N' array is sorted in increasing order. Design an O(N) algorithm to search for a given number in the above arrangement of elements.[10][CO01][BTL Level 2]

5) Answer the following questions:
   A. The following algorithm is used for finding the shortest path from node s to node t in a directed graph with some negative edges: add a large constant to each edge weight so that all the weights become positive, then run Dijkstra's algorithm starting at nodes, and return the shortest path found to node t. Is this a valid method? Either prove that it works correctly or give a counter-example. [4]
   B. Develop a graph G containing both positive and negative edges but does not contain negative cycles, and specify some source s is an element of V where Dijkstra(G; s) does not correctly compute the shortest paths from s. Your graph should have the minimum number of vertices possible. Show which paths are computed incorrectly and explain why Dijkstra fails.[6][CO05][BTL Level 2]

6) Assign an individual (object) to carry out a task (job) where exactly one person corresponds to one job according to that person's ability level. Assigning tasks for each result in different costs. For example, person P, if doing job 1, requires a cost (cost) of c (P, 1), person Q if doing job 2, requires a cost (cost) of c (Q, 2), and so on. So that a job mapping is formed for each person in each job, and the costs, from the results of the mapping, there will be a minimum cost. Using Branch and Bound strategy that solves the problem. Explain the conditions for promising nodes and non-promising

R

nodes. Represent your solution for the following data with a state space tree. [10][CO03][BTL Level 3]

	Job 1	Job 2	Job 3	Job 4
Person 1 (A)	**11**	**4**	**9**	**10**
Person 2 (B)	**8**	**6**	**5**	**9**
Person 3 (C)	**7**	**10**	**3**	**10**
Person 4 (D)	**9**	**8**	**11**	**6**

7) Suppose that we have one machine and a set of n tasks $a_1, a_2 \ldots, a_n$, each of which requires time on the machine. Each task a_j requires t_j time units on the machine (its processing time), yields a profit of p_j, and has a deadline d_j. The machine can process only one task at a time, and task a_j must run without interruption for t_j consecutive time units. If we complete task a_j by its deadline d_j, we receive a profit p_j, but if we complete it after its deadline, we receive no profit. As an optimisation problem, we are given the processing times, profits, and deadlines for a set of n tasks, and we wish to find a schedule that completes all the tasks and returns the greatest amount of profit. The processing times, profits, and deadlines are all nonnegative numbers.
   A. State this problem as a decision problem.
   B. Show that the decision problem is NP-complete and give a polynomial-time algorithm for the decision problem, assuming that all processing times are integers from 1 to n.

   [10 marks] [CO04] [BTL4]

8) Consider the incomplete pseudocode, which takes the following input and produces the output as mentioned below.

   Input: An array A of non-zero floating-point numbers and the ' n' - the size of the array.

   Expected Output: Let P be the product of all elements of array A. An array B storing $b_0, b_1, b_2, \ldots, b_{n-1}$ and $b_i = P/a_i = a_0 \cdots a_i\, a_{i+1} \cdots a_{n-1}$.

   Fill in the following pseudocode to solve the problem mentioned using the Divide and Conquer approach. Also, find the recurrence relation for the completed function [10][CO03][BTL Level 4]

```
void function ( float *A, int n, float *B)
{
        int i, m, M;
        float s, t;
        if (_______)
           { _____________ } //Base case
        m = M = 0;
        while (M <= n - 2)
           { ++m; ++M; ++M;
        }
        function(_______);    //Recursive call
        function(_____________); //Recursive call
        s = A[0] * B[0]; /* We have s = a0a1 · · · am−1 */
        t = A[m] * B[m]; /* We have t = amam+1 · · · an−1 */
        Statement 1: _______
        Statement 2: _______
}
```


R

9) A scattered substring of a string $y = b_1b_2 \ldots b_n$ is a string $x = a_1a_2 \ldots a_m$ with $m \leq n$ such that $a_i = b_j$, for some $1 \leq j_1 < j_2 < \ldots < j_m \leq n$. For example, 12345 is a scattered substring of the string 1ds2j34muy5dy. (a) Give an algorithm that decides whether or not x is a scattered substring of y in time O(n). Explain the working of your algorithm, and prove that it runs in O(n) time.[10][CO02][BTL Level 3]

10) The input to this problem consists of an ordered list of *n* words. The length of the i^{th} word is w_i; that is, the i^{th} word takes up w_i spaces. (For simplicity, assume that there are no spaces between words.) The goal is to break this ordered list of words into lines; this is called a *layout*. Note that you can *not* be allowed to reorder the words. The length of a line is the sum of the lengths of the words on that line. The ideal line length is L. No line may be longer than L, although it may be shorter. The penalty for having a line of length *k* is L− *k*. The total *penalty* is the maximum of the line penalties. The problem is to find a layout that minimises the total penalty. Is there a greedy solution to this problem? If so, give a greedy algorithm and proves its correctness. Else argue that it is not possible using both counter-examples and formal explanations. [10][CO04][BTL Level 3]

****

CO	**Marks**	**BTL**	**Marks**
CO01	**10**	BTL 1	
CO02	**20**	BTL 2	**40**
CO03	**30**	BTL 3	**40**
CO04	**30**	BTL 4	**20**
CO05	**10**	BTL 5	
CO06		BTL 6	

Course Outcome /Bloom's Taxonomy Level (BTL) Mark Distribution Table

R