SSN College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
Department of Computer Science and Engineering
Continuous Assessment Test – II

Degree & Branch:	**B.E. Computer Science and Engineering**	Semester:	**4**
Subject Code & Name:	**UCS1403 DESIGN AND ANALYSIS OF ALGORITHMS**		
Academic Year:	**2020-2021** Batch: 2019-2023	Date:	**10-04-2021**
Time: 10:00 am-11:55 am	Answer All Questions	Maximum: 50 Marks	

Part A (30 min, $10 \times 2 = 20$ marks)

1. Though there may be an exponential number of subproblems, Dynamic Programming (DP) may achieve polynomial running time. What are the reasons? CO3, K1

   A. There is only a polynomial number of distinct subproblems

   B. DP computes the solution to each subproblem once and stores it for subsequent look-up

   C. DP uses iteration instead of recursion

   D. DP solves the problem bottom-up

2. Compared to Dynamic Programming, memoization CO3, K2

   A. solves exactly the same number of subproblems

   B. solves at most the same number of subproblems

   C. solves at least the same number of subproblems

   D. solves different number of subproblems

3. In how many different orders can the chain of matrix multiplications $A \times B \times C \times D$ be evaluated? CO4, K3

   A. 2

   B. 3

   C. 4

   D. 5

4. Suppose binomial coefficient $\binom{n}{k}$ is calculated using dynamic programming, and $C[k, n]$ is stored in a table where k is the row index and n is the column index. Then the table is filled CO4, K2

   A. row by row, from left to right in each row

   B. row by row, from right to left in each row

   C. column by column, from top to bottom in each column

D. column by column, from bottom to top in each column

5. The input is $x_1, \ldots, x_n$, and $y_1, \ldots, y_m$. A subproblem is $x_1, \ldots, x_i$ and $y_1, \ldots, y_j$. What is the total number of subproblems? CO2, K3

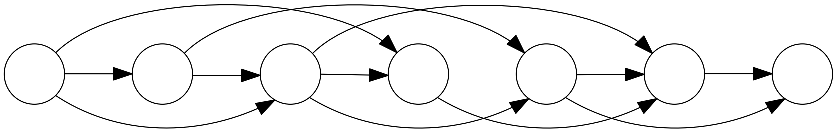
    A. $O(mn)$

    B. $O(m^2)$

    C. $O(n^2)$

    D. $O(m + n)$

6. If the dependency among the subproblems is as shown in the figure, how many subproblems do exhaustive search and DP solve? CO3, K3



    A. 12, 7

    B. 7, 12

    C. 12, 12

    D. 7, 7

7. Floyd-Warshall algorithm recurses on CO4, K2

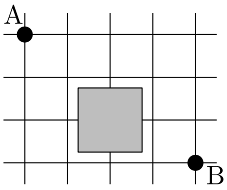
    A. the maximum number of intermediate vertices permitted in the path

    B. the set of intermediate vertices permitted in the path

    C. the ordered set of intermediate vertices permitted in the path

    D. the neighbors of the end vertex of the path

8. Find the number of different shortest paths from point A to point B in a city with perfectly horizontal and vertical streets as shown in the figure. No path can cross the fenced off area shown in grey in the figure. CO1, K3



    A. 31

    B. 17

    C. 9

    D. 8

9. Four people wish to cross a bridge. It is dark, and it is necessary to use a torch when crossing the bridge, but they have only one torch between them. The bridge is narrow, and only two people can be on it at anyone time. The four people take different amounts of time to cross the bridge; when two cross together they proceed at the speed of the slowest. The first person takes 1 minute to cross, the second 2 minutes, the third 5 minutes and the fourth 10 minutes. The torch must be ferried back and forth across the bridge, so that it is always carried when the bridge is crossed. What is the optimal time and the time taken by greedy approach that chooses the fastest persons? CO3, K3

   A. 17 minutes, 19 minutes

   B. 19 minutes, 22 minutes

   C. 17 minutes, 22 minutes

   D. 19 minutes, 19 minutes

10. What is the strategy that extends a partial structure by a sequence of steps, keeping it optimal at each step, to make a globally optimal structure? CO3, K2

   A. Optimization

   B. Greedy

   C. Dynamic Programming

   D. Exhaustive search

11. Dijkstra's algorithm for Single Source Shortest Path greedily adds to the Shortest Path (SP) tree CO3, K2

   A. the fringe vertex nearest to the SP tree

   B. the fringe vertex nearest to the source

   C. the next vertex in the graph nearest to the source

   D. the next shortest edge in the graph

12. Prim's algorithm for MST greedily adds to the partial tree CO3, K2

   A. the fringe vertex nearest to the partial tree

   B. the fringe vertex nearest to the source

   C. the next vertex in the graph nearest to the source

   D. the next shortest edge in the graph that does not form a cycle

13. Removing an edge from an MST (Minimum Spanning Tree) T of a graph and adding a new edge from the graph produces a tree T' that costs CO2, K2

   A. more than T

   B. less than T

C. the same as T

D. depends on the weight of the new edge added

14. Shortest path tree in Dijsktra's algorithm for SSSP can be represented in an array using CO4, K2

A. child pointers from each node

B. child pointer and next sibling pointer

C. parent pointer from each node

D. array of vertices in the order they are selected

15. It is possible to implement Union-Find data structure to achieve CO4, K2

A. $O(1)$ for union and $O(\log n)$ for find

B. $O(1)$ for find and $O(\log n)$ for union

C. $O(1)$ for both union and find

D. $O(n)$ for find and $O(\log n)$ for union

Part B (40 min, 20 marks)

16. An executive in a bank has n customers to serve. The service time required by each customer is known in advance: customer i will take time $t_i, 1 \leq i \leq n$. The bank wants to *minimize the average time* that a customer spends in the bank (Since n, the number of customers, is fixed, this is the same as *minimizing the total time* spent in the bank by all the customers). Design and analyse a greedy algorithm to optimally order the customers. Prove that your algorithm finds the optimal order. CO3, K2

Example: Three customers, with $t_1 = 5, t_2 = 10, t_3 = 3$. There are six possible orders of service. 312 is the optimal order, with the minimum total waiting time.

1.4.1, 2.3.1

Order	Total time	
123	5 + (5 + 10) + (5 + 10 + 3)	= 38
132	5 + (5 + 3) + (5 + 3 + 10)	= 31
213	10 + (10 + 5) + (10 + 5 + 3)	= 43
231	10 + (10 + 3) + (10 + 3 + 5)	= 41
312	**3 + (3 + 5) + (3 + 5 + 10)**	**= 29**
321	3 + (3 + 10) + (3 + 10 + 5)	= 34

17. Suppose you are given a sequence of integers separated by $+$ and $-$ signs; for example: CO3, K3

$$1 + 3 - 2 - 5 + 1 - 6 + 7$$

1.4.1, 2.3.1

You can change the value of this expression by completely parenthesizing it in different ways. For example:

$$((((((1+3)-2)-5)+1)-6)+7)=-1$$
$$((1+3-(2-5))+((1-6)+7))=9$$
$$(((1+(3-2))-(5+1))-(6+7))=-17$$

Given a list of integers separated by $+$ and $-$ signs, we want to compute the maximum possible value the expression can take by adding parentheses. Design an algorithm using DP. What is the structure in which you store the solutions to subproblems and in which order you store them? Analyse the algorithm.

Hint: Store the integers in an array `A[1..N]` and the operators in an array `S[1..N-1]`. A recurrence relation for the maximum value of an expression using `A[i..j]` and `S[i..j-1]` is given by

$$V(i,j)=\max_{i\leq k<j} V(i,k)\; S[k]\; V(k+1,j)$$

Part C (20 min, 10 marks)

18. Consider the problem of making change for n rupees using the least number of coins. CO3, K3

   a) Describe a greedy algorithm to make change, assuming there are four denominations of coins with values 25, 10, 5, 1 rupees. Prove that your algorithm yields an optimal solution.

   b) Suppose that the available coins have the values $c^0, c^1, \ldots, c^k$ for some integers $c > 1$ and $k \geq 1$. Show that the obvious greedy algorithm always yields an optimal solution.

   c) Describe a dynamic programming algorithm that yields an optimal solution for an arbitrary set of coin values.

2.1.3, 2.4.1, 2.4.3