F - 301

END-SEMESTER EXAMINATION, January-2024
Algorithms Design 1 (CSE 3131)

Programme: B.Tech (CSE/CSIT/CIOT/CAIML/CCS/CDS) **Sem:3rd**

Full Marks: 60 **Time: 3 Hours**

Subject/Course Learning Outcome	*Taxonomy Level	Ques. Nos.	Marks
to apply **knowledge of computing and mathematics** to algorithm design. (i) to understand computational tractability considering polynomial time as a definition of efficiency of an algorithm; (ii) to analyze worst-case **running times of algorithms (both recursive and iterative)** using asymptotic analysis;	L2, L3, L4, L5, L6	1(a), 1(b), 1(c), 2(a), 2(b), 2(c)	2+2+2 +2+2 +2
to understand various types and aspects of basic data structures (array, linked list, stack, queue, binary tree) and **advanced data structures like priority queue (implementation using heap)**.	L4, L5, L6	3(a), 3(b), 3(c)	2+2+2
to explain the major **graph algorithms** and their analyses. Employ graphs to model engineering problems, when appropriate	L3, L4, L6	4(a), 4(b), 4(c), 5(a)	2+2+2 +2
to describe the **greedy paradigm** and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms. Derive and describe the performance of greedy algorithms.	L3, L4, L5, L6	5(b), 5(c), 6(a), 6(b), 6(c),	2+2+2 +2+2
to describe the **divide-and-conquer paradigm** and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.	L3, L4, L5, L6	7(a), 7(b), 7(c), 8(a), 8(b), 8(c),	2+2+2 +2+2 +2
to describe the **dynamic programming paradigm** and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic programming algorithms. Derive and solve recurrences describing the performance of dynamic programming algorithms.	L2, L3, L4	9(a), 9(b), 9(c), 10(a), 10(b), 10(c)	2+2+2 +2+2 +2

*Bloom's taxonomy levels: Remembering (L1), Understanding (L2), Application (L3), Analysis (L4), Evaluation (L5), Creation (L6)

Answer all questions. Each question carries equal mark.

1. (a) Given a number, design a recursive algorithm to find the multiplication of the digits of the number. (Eg. Input: 5131 and Output: 5 * 1 * 3 * 1 i.e. 15.) 2

   (b) Find the recurrence relation of the algorithm designed. Explain each part of the recurrence relation. 2

   (c) Estimate the time complexity of the algorithm designed above. 2

2. (a) Draw the recurrence tree for the given recurrence relation: $T(n)=T(n/5) + T(4n/5) + cn$. 2

   (b) Estimate the lower bounding and upper bounding function of the above recurrence relation. 2

   (c) Let $f(n)=3n^2+2n+2$ and $g(n)=4n^2+6$. Can we say $f(n)= \Omega(g(n))$? Justify your answer. 2

3. (a) Given a list of numbers as {16, 15, ?, 12, 14, 8, 6, ?, 9, 11, ?} from 6 to 16 with no repetition. How many MAX-HEAP can be constructed by putting the missing numbers? Depict all. 2

   (b) Construct the MIN-HEAP out of the given HEAP consisting of numbers 11, 7, 19, 2, 16, 8, 28, 15, 17, 12. How many calls are made to MIN-HEAPIFY, and how many swap operations are performed to find the MIN-HEAP? 2

   (c) Given a Min-Priority Queue of elements having their keys as 27, 82, 65, 104, 92, 80, 105, 212, 106, 314, 109. Let an operation **Increase_Key(Q,x,k)** be allowed on the Min-Priority Queue in contrast to the normal Decrease_Key(Q,x,k) operation. If the element with priority 82 is increased to 115, what will be the updated Min-Priority Queue? Design the algorithm for **Increase_Key(Q,x,k)** where Q is the Min-Priority Queue, x is the existing key, and k is the new key. 2

4. (a) Given an undirected graph, G:(V, E) where V denotes a finite set of vertices and E denotes a finite set of edges. How will you verify that the given graph is connected or not? Design an algorithm for it. 2

   (b) Given an undirected graph G where V={A, B, C, D, E, F, G, H} and E={B-A, C-A, B-D, E-B, D-E, E-F, B-C, G-E, G-C, C-H, H-G, F-H, C-F}. Apply BFS at node E and generate a BFS tree. 2

   (c) What must be the size of the QUEUE, which will be sufficient to construct the BFS tree above? Explain why? 2

5. (a) Given a directed graph G(V, E). Design an algorithm to find the vertices in G from which a given node u (where $u \in V$) is reachable. Analyse its time complexity. 2

(b) Conventionally, the PRIM algorithm uses a MIN-priority queue for constructing a minimum weight spanning tree. Design a variant of the PRIM algorithm, finding a suitable substitute in place of the MIN-Priority queue. 2

(c) Compare the above variant of the PRIM algorithm with the original PRIM algorithm. 2

6. (a) There is a tender about installing insulated electricity supply across 'n' locations in a city. The tender contains a map showing the possible connectivity between locations and the length of the required insulated wire. Company "X-Corporation" got the tender and wished to use as minimum wire as possible to complete the project. Suggest an algorithm for it. 2

(b) Given a CACHE with a capacity of 4 blocks and the sequence of blocks being referred as 3, 5, 2, 4, 8, 0, 6, 3, 9, 6, 0, 1, 2, 1, 3, 2, 2, 3, 5, 8, 1, 4. Considering the policy, "The block that last brought in must be evicted first". Produce a cache eviction schedule. 2

(c) Compare the eviction schedule with the one achieved with Optimal Caching, which works with the policy "evict the block that is needed the farthest into the future". 2

7. (a) Given two lists, A and B, whose elements are ordered in non-decreasing order. Design an algorithm to merge these two lists and produce a list C whose elements are ordered in non-increasing order. 2

(b) Compare Merge-Sort and Quick-Sort. Suggest some scenarios where one is preferred over the other. 2

(c) Given a list of numbers 12,3,20,7,5,16,4,10,8. Find how far this list is from being arranged in ascending order. (Illustration: When the given list is {5,1,3,2}, the arrangement in ascending order is {1,2,3,5}. These are the instances (5,1),(5,3),(5,2),(3,2) where the expected order is deviated. Hence, the given list is at a distance of 4 from the list being arranged in ascending order). 2

8. (a) Propose a divide and conquer approach to find the number of 0s in a given binary string of n bits. 2

(b) Find the recurrence relation of the above algorithm and deduce the time complexity of it. 2

(c) Compare its performance with the Naive (or brute-foce) algorithm to find the number of 0s in a given binary string of n bits. 2

9. (a) Compare Dynamic Programming with Greedy Paradigm. 2

(b) Given a weighted directed graph with five vertices {A, B, C, D, E} and edges with their weights as {A->B (2), B->D (3), A->C (2),
C->D (6), C->E (4), E->D (-7), D->C (2)}. Check whether Bellman-Ford can estimate the shortest path from A to all other nodes. 2

(c) In Q9(b), If the answer is "YES", show the shortest path estimate from A to all other nodes. If the answer is "NO", mention the reason for it. 2

10 (a) Given a set of Intervals with their (start-time, finish-time, weight) as S = {(1,3,4), (3,6,5), (6,9,4), (6,7,2), (1,4,3), (2,7,5)}. Use Memoization with Recursion to find a set of non-conflicting intervals with the maximum weight. 2

(b) Estimate the time complexity of finding the set of non-conflicting intervals with the maximum weight considering the algorithm used to solve Q10(a). 2

(c) Given a set of items $S=\{i_1,i_2,i_3,i_4\}$ with their weights w[]={2,3,4,5} and profits p[]={3,4,5,6} and a knapsack with capacity W=9, find an optimal selection of items to fill the knapsack using dynamic programming approach so that the total profit of the selected items will be maximum. 2

End of Questions