

[This question paper contains 10 printed pages.]
Your Roll No.....

A

Sr. No. of Question Paper : 1143

Unique Paper Code : 32341401

Name of the Paper : Design and Analysis of Algorithms

Name of the Course : B.Sc. (H) Computer Science

Semester : IV

Duration : 3 Hours Maximum Marks : 75

Instructions for Candidates

1. Write your Roll No. on the top immediately on receipt of this question paper.
 2. Question No. 1 is compulsory.
 3. Attempt any four of Questions Nos. 2 to 8.
-
1. (a) Consider the recursive version of Insertion sort algorithm as follows :

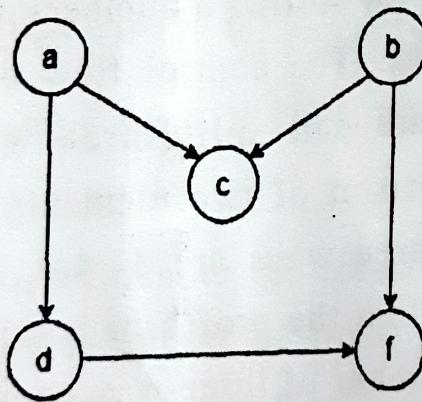
In order to sort $A[1:n]$, we recursively sort $A[1:n-1]$ and then insert $A[n]$ into the sorted array $A[1:n-1]$. Write a recurrence for the running time of this algorithm. (3)

- (b) What are the minimum and maximum number of elements in a binary heap of height h ? (2)
- (c) Which of the following algorithms are stable: Insertion sort, Quicksort? Justify with the help of an example. (3)
- (d) What are the two key factors that decide whether Dynamic Programming is applicable for an optimization problem or not? (2)
- (e) "The minimum spanning tree in a graph is not always unique." Justify. Give a graph with 5 nodes that has two different minimum spanning trees. (2)
- (f) Would you use BFS to find the shortest path between two nodes in a weighted graph with arbitrary edge weights? Justify your answer with the help of a graph having at least 5 Vertices and at least 7 edges. (3)
- (g) "Counting sort is a comparison sort algorithm." Yes or No. Justify your answer using the input $A = < 4, 3, 2, 4, 1, 5, 2, 4, 3 >$ (3)
- (h) A priority queue can be implemented in two different ways using min-heap and using singly linked list in which elements are stored in sorted

order (Smaller values indicates higher priority). Compare the time complexity of following operations when performed on two different implementations of priority queue.

- (1) Finding the highest priority element
 - (2) Deleting the highest priority element
 - (3) Increase the priority of a certain element
- (6)

- (i) Find possible Topological sorts of the given directed acyclic graph. Give any four : (2)



- (j) Give the worst case for the merge algorithm to merge two sorted arrays $A[1 \dots k]$, $B[1 \dots m]$ (where $n = k + m$) and give the total number of comparisons in the worst case (in terms of n). Merge the two sorted arrays $A = [3, 7, 9, 12, 14]$ and $B = [2, 5, 6, 10]$ using merge algorithm. How many number of comparisons are done by the algorithm in the above example? (6)

(k) Consider the Interval Scheduling problem wherein we are given a resource and a set of requests each having a start time and a finish time. The goal is to maximize the number of requests scheduled. Show that the following greedy strategy does not give an optimal solution for the above problem.

Greedy strategy: Select the request with fewest number of incompatible requests. (3)

2. (a) Professor William claims that the $\Omega(n \lg n)$ lower bound for sorting n numbers does not apply to his machine. The control flow of a program on his machine can split three ways after a single comparison of two elements of the array $a_i : a_j$. The three ways are $a_i < a_j$, $a_i = a_j$, or $a_i > a_j$. Show that the professor is wrong by proving that the number of three-way comparisons required to sort n elements is $\Omega(n \lg n)$. (5)

- (b) Suppose you have an algorithm to find median of n elements of an unsorted array in $O(n)$ time in the worst case. Now consider an implementation of Quicksort where you first find median using the above algorithm, then use median as pivot. What will be the time complexity of this modified

Quicksort? Write down the recurrence relation to justify your time complexity (Median of n elements is the element whose rank is $n/2$ if n is even and it is $(n+1)/2$ if n is odd). (5)

3. (a) Illustrate the operation of BUILD -MAX -H EAP on the array A <4, 3, 17, 10, 28, 19, 6, 12, 7>. Write down the total number of comparisons done by BUILD -MAX -H EAP. (5)
- (b) Consider the following recursive relation for 0-1 knapsack problem.

If $w < w_i$, then $\text{OPT}(i, w) = \text{OPT}(i-1, w)$
 else $\text{OPT}(i, w) = \max(\text{OPT}(i-1, w), v_i + \text{OPT}(i-1, \underline{\hspace{2cm}}))$

where $\text{OPT}(i, w)$ denote the value of the optimal solution using a subset of items $\{1, 2, \dots, i\}$ with maximum allowed weight w .

v_i is the cost of i th item

Fill the missing value.

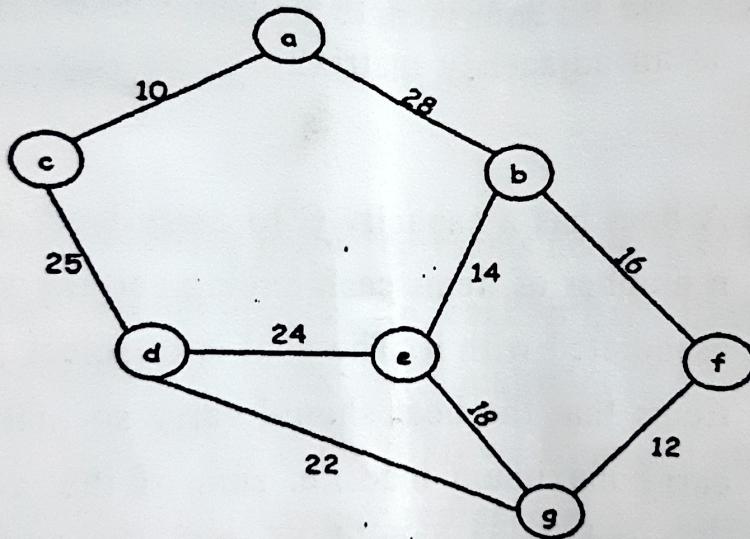
What is the running time of the recursive implementation of the above recurrence? Justify. Give memoized recursive algorithm for the above problem. Explain how does it improve the running time?

(5)

4. (a) Give an algorithm to sort n integers in the range 0 to $n^3 - 1$ in $O(n)$ time. Justify its time complexity. (4)
- (b) Suppose we use randomized select to select the minimum element of the array $A = \{4, 2, 1, 7, 8, 12, 3, 0, 9, 5, 10\}$. Describe a sequence of partitions that result in a worst case performance of randomized select. (4)
- (c) Suppose an input to the bucket sort algorithm is not uniformly distributed. What will be the effect of this condition on the running time of the algorithm? Justify. (2)
5. (a) A Shopkeeper has n empty boxes and M number of balls. Let $\{K_1, K_2, \dots, K_n\}$ denote the number of balls that each box can store:
 Given M and $\{K_1, K_2, \dots, K_n\}$, describe a greedy algorithm which determines the minimum number of boxes needed to store the balls. Give time complexity of the algorithm. (4)
- (b) Design a $O(|V| + |E|)$ time algorithm to find whether a given undirected graph is bipartite (where V is the set of vertices, E is the set of edges of the graph). (4)

(c) Give space requirements of adjacency matrix and adjacency list representation having m edges and n vertices. (2)

6. (a) "Prim's algorithm only include an edge in the Minimum Spanning tree when it is justified by the Cut property." State the Cut Property. Justify the above statement on the given graph showing cuts in all the intermediate steps.



(5)

- (b) Suppose divide and conquer approach is used by an algorithm to solve a problem on the input of size n . The algorithm divides the problem into k number of smaller instances, each of which is $1/b$ the size of the original problem. It solves the problem recursively on these smaller instances and combine their solutions to construct the final

solution of the original problem. Let $G(n)$ denotes the cost of dividing the problem into smaller instances and $F(n)$ denotes the cost of combining the solutions.

Write the recurrence relation to find the running time of the algorithm. Give $G(n)$ and $F(n)$ for Both Quicksort and Mergesort algorithm. (3)

(c) Let G be a graph with n vertices and m edges. What is the upper bound on the running time of Depth First Search on G , where G is represented as an adjacency matrix? (2)

7. (a) A boat has a capacity C to carry load. There are n number of items each having certain weight W_i associated with it. The goal is to select the set of items that the boat should carry so that it could carry maximum load(i.e. sum of the weights of selected items should be maximum) within its capacity C .

Design a Polynomial time Dynamic Programming algorithm for the problem. Derive the time complexity of the above algorithm.

Run this algorithm on the sample instance given below to find the optimal solution. (6)

Capacity of the boat = 4kg

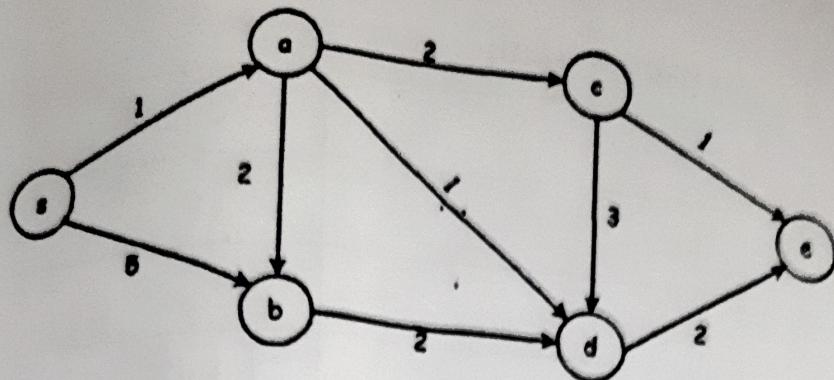
Items	Weights
Item1	3 kg
Item2	2 kg
Item3	1 kg

- (b) Suppose we perform a sequence of n operations on a data structure in which the i th operation costs i if i is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation. (4)

8. (a) The following graph represents network of airports that are connected to each other. Each edge represents the distance (in multiples of 1000 miles) covered by the flight to travel from one airport to another and vertices represents the airports. A flight starts from airport s and has to reach destinations c , d and e . Run an efficient algorithm to find the route taken by the flight to each to its destinations in minimum possible time. Show all the intermediate steps taken by the algorithm.

Also derive its time complexity.

(5)



- (b) There are n Jobs where each job starts at time s_i and finishes at time f_i . There is a profit associated with each job. The goal is to find a subset of non-overlapping jobs such that the sum of their profits is maximum.

Given below is the instance of the problem :

Job Number	Start Time (s_i)	Finish time (f_i)	Profit (p_i)
Job1	0	6	60
Job2	1	4	30
Job3	3	5	10
Job4	5	7	30
Job5	5	9	50
Job6	7	8	10

Give a Dynamic programming iterative solution for the above problem. Explain the recurrence relation used in the solution.

Show that the "Optimal solution to the above problem contains within it optimal solution to its subproblems." With reference to the above example. (5)

(1500)