[This question paper contains 8+2 printed pages]

Roll No.

S. No. of Question Paper : 6512

Unique Paper Code : 32341401 HC

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

(Write your Roll No. on the top immediately on receipt of this question paper.)

Question No 1 of 35 marks is compulsory.

Attempt any *four* questions from Q. No. 2 to Q. No. 7.

1. (*a*) Arrange the following functions in the increasing order of their rate of growth : $n^2 \log(n)$, 2^n, 2^{2^n}, $n^{\log(n)}$ 2

(*b*) Consider a variation of the merge sort algorithm that solves a problem of size n by dividing it into two subproblems of sizes 2n/3 and n/3, and then merging the solutions. Find the recurrence for the running time of the above algorithm and solve it. 4

P.T.O.

P. T. O.

(c) A thief wants to steal all the gold dust from a st having W kg of it. The thief has n sacks havi different capacities. Give an efficient algorithm for thief to fill his sacks with dust so that the number sacks used is minimized.

(d) Consider an instance of the weighted interval schedul problem with 6 intervals as specified below :

Interval number	Start time (s_i)	Finish time (f_i)	Weight (
1	0	2	2
2	1	3	4
3	2	4	4
4	1	5	7
5	4	5	2
6	4	6	1

With the help of the above example argue the memorized recursive algorithm solves lesser nun of subproblems than the corresponding itera algorithm.

(e) Can a red-black tree have

(i) a black node without any sibling ? Justify.

(ii) a red node without any sibling ? Justify. 4

(f) Consider an algorithm A with run time O(n). What is the condition on A for it to be usable as the intermediate sort in Radix sort ? Explain. 3

(g) Let G = (V,E) be an undirected path graph with n nodes. We call a graph a 'path' if its nodes can be written as $v_1, v_2, \ldots\ldots, v_n$ with an edge between v_i and v_j if and only if the numbers i and j differ by exactly 1. With each node v_i, we associate a positive integer weight denoted by w_i. A subset of the nodes is called an independent set if no two of them are joined by an edge. Consider the following greedy algorithm for finding an independent set of maximum total weight in a path graph :

Start with S = empty set

While some node remains in G

P. T. O.

Pick a node v_i of maximum weight

Add v_i to S

Delete v_i and its neighbors from G .

Endwhile

Return S

Give an example to show that the above algorithm does not always find an optimal solution. 4

(*h*) Discuss the run time complexity of the naïve string matching algorithm. 2

(*i*) Consider a directed graph G = (V, E). Given two vertices *s* and *t* in V, mention the name of an algorithm that can be used to determine if there exists at least one *s*-*t* path in G. What is the running time of the algorithm ? 3

(*j*) Give an efficient algorithm to find the maximum element in a min-heap. Give the exact running time of the algorithm. 3

(*k*) Would you use DFS to find the shortest-path distance between two nodes in a graph ? Justify your answer. 3

(*a*) Consider an adjacency list representation of a directed graph wherein an array of the outgoing edges (and not the incoming edges) for each vertex is maintained. Give an algorithm to compute the indegree for each vertex and discuss the time complexity of the algorithm. 5

(*b*) For each of the following operations does a Red Black Tree work faster than a Binary Search Tree ? Elaborate your answer.

(*i*) Insertion

(*ii*) Preorder traversal. 5

(*a*) Consider the following algorithm that takes as input an array of n integers and an integer S. It finds whether there exist two elements in the array that sum up to S and returns 1 on success and 0 on failure.

P. T. O.

```
FindPair (Arr, n, S)

    Quicksort (Arr, 1, n)
    for i = 1 to n
        flag = BinarySearch(Arr, i+1, n, | S-Arr[i] | )
        if (flag)
            return 1
        endif
    endFor
    return 0
```

FindPair uses the following algorithms :

Quicksort (Array, First, Last).

BinarySearch (Array, First, last, element)

Analyze the worst case running time of FindPair. 4

(*b*) Can a graph G in which edge weights are not necessarily distinct, have more than one minimum spanning trees (MST). If yes, give an example; if no, justify. 4

(*c*) Is Merge sort (*i*) in place (*ii*) stable ? Explain. 2

4. (*a*) Consider a stack S that supports the following operations :

Push(S, x) : push element x onto stack S

Pop(S) : pop the top element from stack S and return it

Multipop(S, k) : remove top k elements of stack S

Using the aggregate method of analysis, determine the amortised cost per operation when a sequence of n operations is performed on an empty stack. 3

(*b*) For each of the following sorting algorithms give an input of size five for which it shows worst case behaviour :

(*i*) Merge sort

(*ii*) Quick sort. 4

(*c*) For the variant of interval scheduling problem that minimizes lateness, give an instance with two different optimal solutions, neither of which has any inversions or idle time. 3

5. (a) Give an example graph having five nodes that has two different topological orderings. Also, show the topological orderings. 4

(b) Observe that selection sort and heap sort function in a similar fashion in the following sense - in any iteration both the sorting techniques find the maximum amongst the elements yet to be processed and place it appropriately. However, they have different running times. Give the asymptotic time complexity of both and comment on what makes their running times different. 4

(c) Create a Red-black tree with successive insertions for the following sequence of numbers: 10,8,14,12,13. 2

6. (a) Suppose we are given an instance of the Shortest s-t path problem on a directed graph G. We assume all the edge costs are positive and distinct. Let P be a minimum cost s-t path for this instance. Now suppose

we replace each edge cost c_e by its square, c_e^2, thereby creating a new instance of the problem with the same graph but different costs. Is it necessary for P to still be a minimum cost s-t path for this new instance ? Explain. 4

(b) Why is Bucket sort considered to be a non-comparison based sorting algorithm (where no comparison of keys is performed for sorting the list) despite the fact that insertion sort, which is used to sort individual buckets, is comparison based ? 2

(c) What kind of inputs will lead to (*i*) best case (*ii*) worst case performance for insertion sort algorithm ? Give the running time for both the cases. 4

7. (a) Show that at most 3*floor(n/2) comparisons are sufficient to find both the minimum and maximum in a given array of size n. 3

P. T. O.

(*b*) Consider the following recursive algorithm to find an optimal solution to the subset sum problem :

```
Compute_opt(i,w)
If i = 0 or w = 0 then
    Return 0
Else
    If w < w_i then:
        Return Compute_opt(i-1, w)
    else
        Return max(w_i + Compute_opt(i-1, w - w_i),
        Compute_opt(i-1,w))
```

(*i*) Explain why does this algorithm take exponential time to run in the worst case.

(*ii*) What changes should be made to the above algorithm to make it run in polynomial time.

(*iii*) Consider an optimal solution to the above problem. Is it possible for this solution to contain a sub-optimal solution to a subproblem ? Explain. 2+2+3