11/5/18 (morning)

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, $2^{2\wedge n}$, $n^{\log(n)}$ 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.

(*c*) A thief wants to steal all the gold dust from a store having W kg of it. The thief has n sacks having different capacities. Give an efficient algorithm for the thief to fill his sacks with dust so that the number of sacks used is minimized. 3

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

Interval number	Start time (s_i)	Finish time (f_i)	Weight (v_i)
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 that the memorized recursive algorithm solves lesser number of subproblems than the corresponding iterative algorithm. 4

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

(*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