No. of Printed Pages : **3**

Sl. No.

C4-R4 : ADVANCED ALGORITHMS

NOTE :

1. Answer question 1 and any FOUR from questions 2 to 7.

2. Parts of the same question should be answered together and in the same sequence.

Time : 3 Hours **Total Marks : 100**

1. (a) Define: Big-Oh, Omega & Theta notations with proper graphs.

(b) What is the disadvantage of Naïve String Matching ? Explain with an example.

(c) Compare Kruskal's and Prim's algorithms for finding the minimum spanning tree.

(d) Define NP, NP-Complete and NP-Hard Problems.

(e) With proper justification arrange the following asymptotic complexities in the increasing order : 2^n, $n^2 \log n$, $n^{3/2}$, $\sqrt{n}$, $n \log n$, n^2, $n^{\log n}$.

(f) How does Dynamic Programming differ from the Divide and Conquer method ?

(g) "Though the best case and average case time complexities of Quick Sort are $\Omega(n\log(n))$ and $\theta(n \log(n))$ respectively, but the worst case time complexity is $O(n^2)$." Justify with an example.

(7x4)

2. (a) Following table presents the weights and associated profits of seven items (n = 7) in the Knapsack problem. Assuming the capacity of the Knapsack as 15, i.e. m = 15, find the optimal solution for the given instance of the Knapsack problem.

Item	Total Weight	Total Profit
1	2	10
2	3	5
3	5	15
4	7	7
5	1	6
6	4	18
7	1	3

(b) What do you mean by amortized analysis ? What is the usefulness of it ?

(c) Why do we analyze the expected running time of a randomized algorithm and not its worst-case running time ?

(7+5+6)

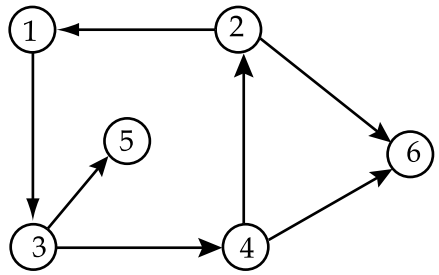
3. (a) Prove that any comparison-based sorting algorithm in an array A of n numbers must take $\Omega(n \log n)$ running time in the worst case.

(b) Sort the following list of English words in increasing order using radix sort technique: COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX.

(c) Suppose that all edge weights in a graph are integers in the range from 1 to $|V|$. How fast can you make Prim's algorithm run ? What if the edge weights are integers in the range from 1 to W for some constant W ?

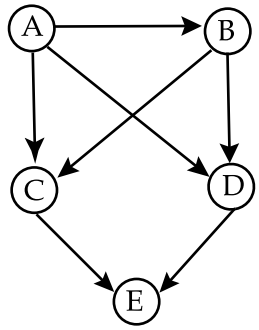
(6+6+6)

4. (a) Find the order in which the vertices will be traversed in depth first search of the graph shown below, starting at vertex 1. Show the steps clearly.



(b) Prove or disprove: "A directed graph G is acyclic if and only if a depth-first search of G yields no back edges".

(c) Find the number of different topological orderings possible for the given graph :



(6+6+6)

5. (a) Create Huffman Tree from given input characters a, b, c, d, e and f whose frequencies in a given text are respectively, 7, 5, 3, 2, 12 and 9.

(b) In context of Huffman code, when a code is said to be optimal ? What is the time complexity for obtaining an optimal Huffman Coding Tree ?

(c) Multiply the given matrices P and Q using Strassen's Matrix Multiplication :

$$P = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \qquad Q = \begin{bmatrix} 4 & 1 \\ 2 & 5 \end{bmatrix}$$

(6+6+6)

6. (a) Compute the prefix function π for the pattern ababbabbabbababbabb when the alphabet is $\sum$={a,b}.

(b) Argue for the correctness of Heap Sort using the following loop invariant :
At the start of the iteration with an i of the for loop (a) the subarray A[1..i] is a max-heap containing the i smallest elements of A[1.. n], and (b) the subarray A[i+1..n] contains the n-i largest elements of A[1..n] in correctly sorted order (ascending order).

(c) Solve the following recurrence relation using recurrence tree method $T(n) = 3T(n/4) + cn^2$.

(5+7+6)

7. (a) Let us modify the traditional binary search algorithm, BinarySearch (low, high, key) as follows: instead of having the search interval in each iteration, select one of the remaining positions at random. Assume that every position between low and high is equally likely to be chosen by the modified algorithm. Compare the performance of this modified binary search algorithm with the traditional binary search algorithm.

(b) What is a flow network ? Also explain Residual Network.

(c) Derive recurrence equation for Merge Sort algorithm. Comment on the optimality of the Merge Sort algorithm.

(7+4+7)

- o O o -