B.E. COMPUTER SCIENCE AND ENGINEERING THIRD YEAR SECOND SEMESTER SUPPLEMENTARY EXAM 2023

Subject-**Design and Analysis of Algorithms**

Time: **3 hours** Full Marks: **100**

*Answer Question No. 1(**compulsory**) and question no. (**2 or 3**) and question no.(**4 or 5**) and question no. (**6 or 7**) and question no.(**8 or 9**)*

1. State whether the following statements is TRUE or FALSE with proper justifications.

a) $O(n/100) = O(n) = O(n^2)$

b) An algorithm with time complexity of O(n!) is better than the algorithm with time complexity of $O(2^n)$

c) The combination step of Divide and Conquer Algorithm for a problem should take running time of $O(n^k)$, where k is a constant

d) Greedy algorithm can always give the correct solution to a problem.

e) In Dynamic programming approach, a problem is solved by solving a collection of overlapping sub-problems.

f) If a problem is in NP, it may not be NP-hard.

g) Prim's algorithm is a greedy algorithm

h) Master Theorem can be applied to solving recurrence equation for the running time of any recursive algorithm.

i) For any connected undirected weighted graph G=(V,E), the Prim's algorithm always finds minimum spanning tree .

j) Dijktra's algorithm can only find the shortest path from source node to the farthest node from the source. 4 x 10 = **40 marks**

Group A

2. a)Suppose you are choosing between the following two algorithms

I) Algorithm A solves problems of size *n* by recursively solving two sub-problems of size *n-1* and then combining the solutions in constant time

II) Algorithm B solves problem of size *n* by dividing them into nine sub-problems of size n/3, recursively solving each sub-problem , and then combining the solutions in $O(n^2)$ time

Compute the running time for the algorithms A & B. Then show which one is better in terms of running time.

b) State Master Theorem. Can we solve the following recurrence equation using Master Theorem ?-explain. $T(n) = 2T(\sqrt{n}) + 1$, T(1)= 1. Solve this recurrence equation

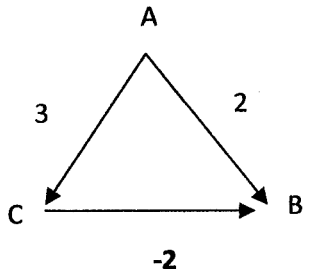
c) If $f(n) = n^3/2$, and $g(n) = 37n^2 + 120n + 17$, using the limit method , show that $g(n) \in O(f(n))$, but $f(n) \notin O(g(n))$ 5+5+5=15 marks

OR

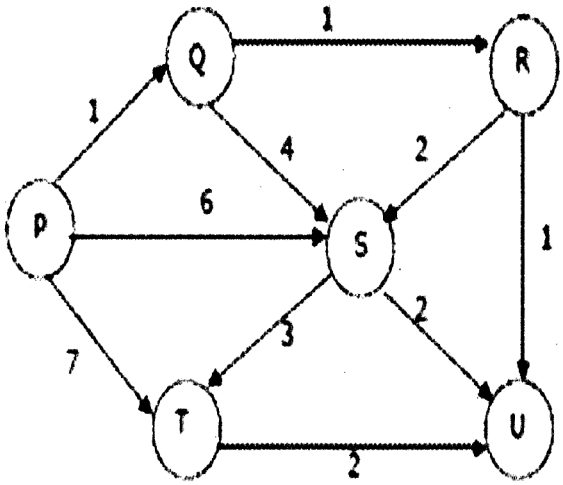
3. Write the school level algorithm for multiplying two large n-digit numbers. Compute its time complexity. Write an efficient Divide and Conquer algorithm for multiplying two large n-digit numbers. Compute its time complexity. 5 + 10 =15 marks

Group B

4. a) Consider the following directed graph and source node A, and explain whether Dijkstra's algorithm will give the correct solution or not.

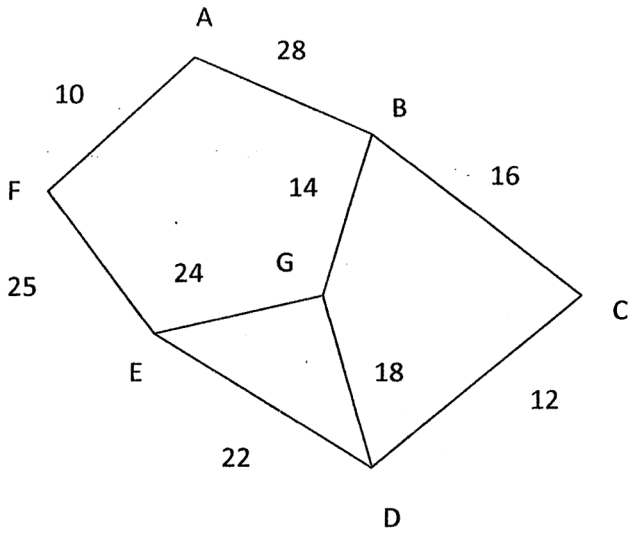


b) Consider the directed graph given below and use Dijkstra's single source shortest-path algorithm on this graph starting with **node p** (show step by step progress of the algorithm with the updates in the priority queue and node values).



5+10=15 marks

5. Considering A as the source node , find out minimum spanning tree for the following graph using Prim's algorithm. Show the steps clearly.



Compute the time complexity of the Prim's algorithm if priority queue is used in its implementation process. 10+5=15 marks

GROUP C

6. a) In the longest increasing subsequence problem, the input is a sequence of numbers: a_1, a_2..a_n. A subsequence is any subset of these numbers taken in order, of the form $a_{i_1}, a_{i_2} \ldots a_{i_k}$, where $1 \leq i_1 < i_2 < \cdots i_k \leq n$, and an increasing sequence is one in which numbers are getting strictly larger. The task is to find the increasing subsequence of greatest length. For instance, the longest increasing subsequence of 5, 2, 8, 6, 3, 6, 9, 7 is 2, 3, 6, 9.

Explain the dynamic programming algorithm with DAG based input representation for finding the increasing subsequence of greatest length. Compute the time complexity of the algorithm.

b) Use the dynamic programming approach to find minimum edit distance and the optimal alignment between the strings the strings "E X P O N E N T I A L" and "P O L Y N O M I A L". Show only the table of computations, min edit distance and final optimal alignment.

7+8=15 marks

OR

7. Describe recursive solution and dynamic programming solution to finding n-th Fibonacci number. Compare time complexity for both the versions.

What is time complexity of the brute force method and dynamic programming method for solving string alignment problem. 10 +5=15 marks

GROUP D

8. a) Does the sorting problem belong to the class NP? - Give reasons.

b) What does NP stands for? Is *Boolean Satisfiability Problem* (SAT) NP-complete? - explain.

c) Explain the easier method to check whether a given problem is NP-complete or not 5 x 3=15 marks

OR

9. a) Consider that a minimum cost communication network will be set up among the major cities of a country in such a way that there will a communication path between any two cites. You may consider each city as a vertex and each communication link as the edge between two vertices. The weight of an edge is the cost of constructing the link represented by the edge. Every connected sub -graph that includes all the vertices represents a feasible solution. Under the assumption that all weights are non-negative, suggest an efficient algorithm for finding the optimal solution with minimum-cost. Formulate the problem as the graph problem first and discuss which and how the efficient graph algorithms can be used to solve this problem. How the running time of your algorithm differs from the brute force algorithm?

b) Define (i) the class NP, (ii) the NP-completeness 10 + 5 = 15 marks