Full Marks: 70
Time: 3 hours

Question No. 1 is compulsory. Answer any five from the rest.
The figures in the right hand margin indicate marks.

1. Answer all questions. **[2 ×10]**

[a] Let A be an algorithm that finds optimal solution for problem P? Differentiate between performance analysis and performance measurement of the algorithm A?

[b] Show that $n^3 \log n$ is $\Omega(n^3)$?

[c] What are basic principles of DYNAMIC PROGRAMMING?

[d] What are the two different approach to GREEDY ALGORITHM design?

[e] Define "***optimization Problem***" with an example showing objective function and constraints?

[f] Explain, why "spanning tree" is always unique if constructed using Kruskal's algorithm?

[g] Solve the following recurrence relation assuming N as an integer power of 2

$T(n) = 1$; $n = 1$

$T(n) = 4T(n/2) + 4n^4$; $n \geq 2$ and n is power of 2

[h] What is the need of the approximation algorithms for NP-hard problem?

[i] When is a divide-and-conquer algorithm very efficient and when is it not?

[j] What data structure would you use to keep track of live nodes in a best-first branch-and-bound algorithm?

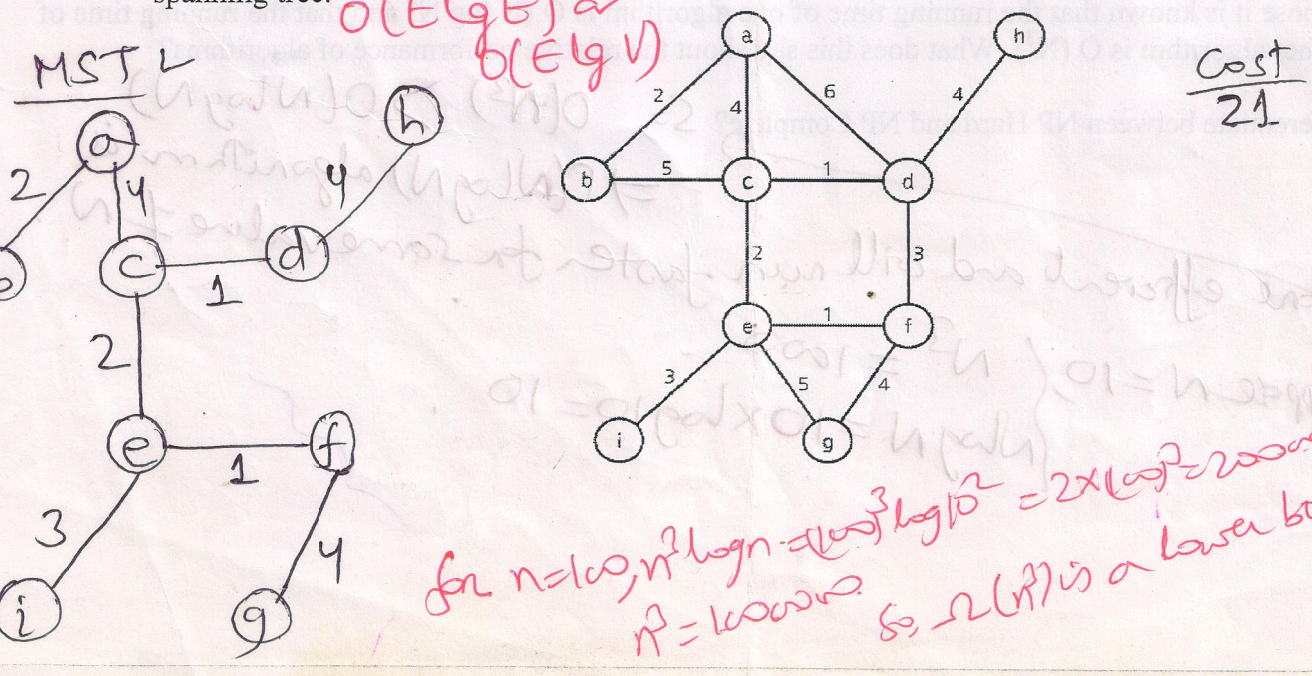
2[a] Write a recursive divide and conquer algorithm to perform binary search in a sorted array? Show that the worst case complexity of your algorithm is O(log n)? Is your algorithm optimal? **[5]**

[b] Construct a min-heap with the following list **{13, 15, 41, 11, 17, 19, 26, 77, 32, 34, 14}.** What is the time complexity of the whole process? Comment about time and space complexity of heap sort. **[5]**

3[a] Give an analysis of merge sort process in the light of "Divide and Conquer paradigm". Find out the time complexity of the merge sort algorithm from recurrence relation by the method of telescoping. **[5]**

[b] Discuss the concept of pattern matching algorithm? What is the advantage of Boyer-Moore method over BruteForce method to match a pattern P in a given string T? **[5]**

4[a] Find the minimum spanning tree for the following graph using Kruskal's algorithms? What is the time complexity of this algorithm? Explain the reason, why kruskal's algorithm always yields optimal spanning tree. **[5]**

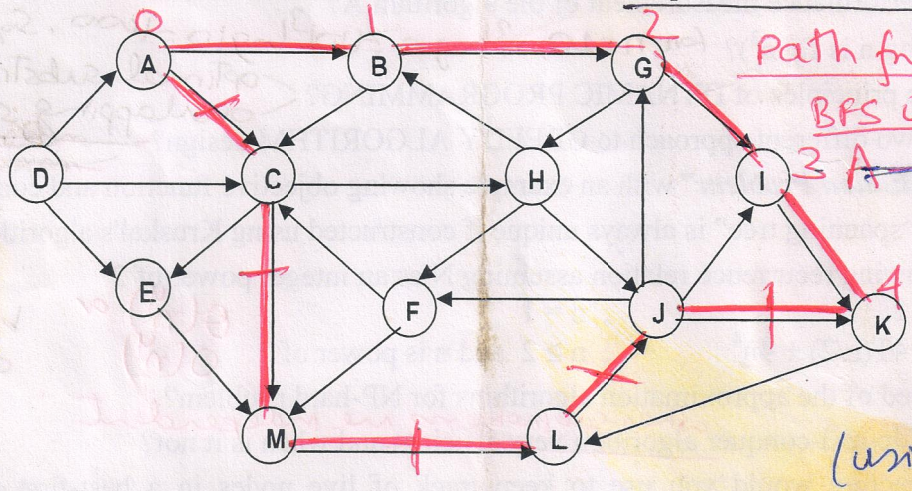


P.T.O.

4[b] State an algorithm to find a minimum-cost path from source to ~~shrink~~ sink in a multistage graph, using backward approach using ***dynamic programming*** paradigm. Bellman-ford algorithm [5]

5[a] Discuss the mechanism of Branch and bound algorithm as *"all state space search method"*. Considering 0/1 Knapsack problem, explain the application of ***Least cost Branch and bound search and FIFO branch and Bound search*** to found an optimal solution. [5]

[b] Write an algorithm to perform BFS on a graph? Find out the path from A to K using BFS ? [5]



Path from A to K using BFS is
A → C → M → L
→ J → K
A → B → G
→ I → K
(using queue)

6[a] Define *approximation algorithm* for a problem P? How do you characterize approximation algorithms? [5]

[b] What are operations supported by disjoint set data structure? Discuss the Union find algorithm for any two elements belong to disjoint set? [5]

7[a] Explain and analyze an algorithm for finding all-pairs maximum-lengths (of simple paths) in a graph? [5]

[b] State the reasons for selecting Backtracking algorithm? Explain how the efficiency of the Backtracking is estimated considering 8-queen problem. [5]

8. Answer all questions. [2.5×4]

[a] Explain the *backtracking algorithm* with 4 Queens problem on a 4*4 chess board?

[b] What are the advantages of dynamic programming over greedy method? always get an optimal solution, bottom up approa

[c] Suppose it is known that the running time of one algorithm is O (N log N) and that the running time of another algorithm is O (N^3). What does this say about the relative performance of algorithms?

[d] Differentiate between NP Hard and NP Complete? So, $O(N^3) > O(N \log N)$
$\Rightarrow O(N \log N)$ algorithm is more efficient and will run faster for some value of N.

Suppose $N=10$, $N^3 = 1000$
$N \log N = 10 \times \log 10 = 10$.