MAHINDRA UNIVERSITY HYDERABAD

École Centrale School of Engineering

End Semester Examination

Program: B.Tech
Branch: CSE/AI/ECM/CM/CB
Subject: Design and Analysis of Algorithms (CS/AI 2102)
Start Time: 10:00 AM

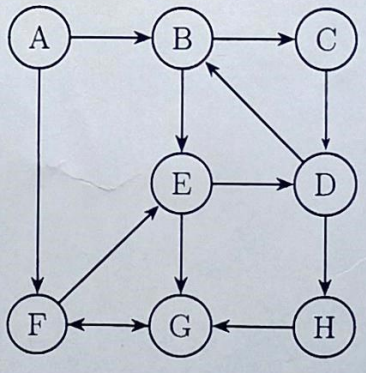
Year: Second/Third
Semester: First
Date: 16-12-2024

Please read the following instructions before answering questions.

1. Answer **all** questions. There are **12** questions in total. The first **8** questions are worth **7.5** marks each, while the last **4** questions are worth **10** marks each.
2. Provide concise and focused responses – include only essential details.
3. Answer all parts of a question in one place. If not, you may risk losing marks.
4. If a question seems unclear, clearly state your assumptions before answering.

---

1. Prove that $\log(n!) \in \Theta(n \log n)$.
2. Design an algorithm for computing $p(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ at a given point x using exactly n additions and n multiplications.
3. Design an algorithm to sort n comparable elements in $O(n \log n)$ time (in the worst case, too) using the divide-and-conquer approach.
4. Perform Breadth-First-Search(BFS) on the following graph; whenever there is a choice of vertices, pick the one that is alphabetically first, and provide the list of vertices in the order they are visited during the BFS.



5. What is the time complexity of Dijkstra's algorithm in the following scenarios?
   (a) When a simple array implementation is used.
   (b) When a heap-based implementation is used.

6. Solve the interval scheduling problem for the following set of intervals by selecting the maximum number of non-overlapping intervals: $[1,3]$, $[2,5]$, $[4,6]$, $[5,9]$, $[7,9]$, $[8,10]$.

7. Explain the Naive String Matching algorithm and analyze its time complexity.

8. State the master theorem and answer the following question. Suppose you are choosing between the following three algorithms:

   (a) Algorithm A solves problems of size n by dividing them into eight subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.

   (b) Algorithm B solves problems of size n by recursively solving two subproblems of size $n-1$ and then combining the solutions in constant time.

   (c) Algorithm C solves problems of size n by dividing them into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.

   What are the running times of each of these algorithms (in big-O notation), and which one would you choose and why?

9. A k-way merge operation. Suppose you have k sorted arrays, each with n elements, and you want to combine them into a single sorted array of kn elements.

   (a) Here's one strategy: Merge the first two arrays, then merge in the third, then merge in the fourth, and so on. What is the time complexity of this algorithm, in terms of k and n?

   (b) Give a more efficient solution to this problem, using divide-and-conquer.

10. Given two strings $x = x_1x_2\ldots x_n$ and $y = y_1y_2\ldots y_m$, we wish to find the length of their longest common subsequence, that is, the largest k for which there are indices $i_1 < i_2 < \ldots < i_k$ and $j_1 < j_2 < \ldots < j_k$ with $x_{i_1}x_{i_2}\ldots x_{i_k} = y_{j_1}y_{j_2}\ldots y_{j_k}$. Show how to do this in time $O(mn)$.

11. Design a deterministic finite automaton (DFA) to recognize the pattern **ABAB** and demonstrate its step-by-step matching process on the text **ABABABAB**.

12. Define the following concepts:

   (a) Problems belonging to the class P.

   (b) Problems belonging to the class NP.

   (c) $NP-Complete$ problems.

   (d) $NP-Hard$ problems.

   (e) Approximation Algorithm.