INDIAN STATISTICAL INSTITUTE

Second Class Test

M.Tech CS 2024-2025

*Design and Analysis of Algorithms*

Date: November 9, 2024 Maximum Marks: 20 Duration: 1.5 Hours

1. (a) Which of the following is used to find a shortest path in an unweighted graph: BFS or DFS? Given an example to show why the other search technique fails? 5

   (b) If you are given a weighted graph, present an algorithm to find a shortest path in the graph. You may assume that all the weights are positive. 8

   (c) Analyze the running time of your algorithm. 4

   (d) Explain the correctness of your algorithm. 8

**OR**

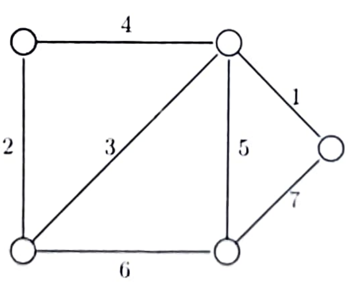
2. (a) What is the difference between dynamic programming and greedy algorithms? 3

   (b) Let $G$ be a weighted graph that has a cut $(S, V\backslash S)$. We say $(u, v)$ is a crossing edge *crossing edge* w.r.t. this cut if $u \in S$ and $v \in V\backslash S$. Let $e$ be a least weight crossing edge. Prove that, there exists a minimum spanning tree that contains $e$. 6

   (c) How can this property be used to design an algorithm for finding a minimum spanning tree? 4

   (d) What is the cut used in the Prim's algorithm? Present the algorithm. 7

   (e) Describe each step of Kruskal's algorithm for the following graph: 5



INDIAN STATISTICAL INSTITUTE

Semester Examination

M.Tech CS 2024-2025

Design and Analysis of Algorithms

Date: 29.11.2024 | Maximum Marks: 100 | Duration: 3.5 Hours

## Group A

*Note: You may answer all the questions in Group A, but the maximum you can score is 75.*

1. (a) For some constants $a_1, a_2, \ldots, a_\ell > 1$, consider the following recurrence:

$$T(n) \leq \begin{cases} O(1) & \text{if } n < C, \\ T(\frac{n}{a_1}) + T(\frac{n}{a_2}) + \ldots + T(\frac{n}{a_\ell}) + O(n) & \text{if } n \geq C, \end{cases}$$

for some sufficiently large constant $C$. Show that, if $\frac{1}{a_1} + \frac{1}{a_2} + \ldots + \frac{1}{a_\ell} < 1$, then $T(n) = O(n)$.

(b) Given an array $A$ of $n$ distinct elements, for an element $x$ in $A$, define $L_x$ as the set of elements in $A$ smaller than $x$ and $R_x$ as the set of elements in $A$ larger than $x$. We say, $x$ is an *approximate median* if $|L_x| \geq \frac{n}{4}$ and $|R_x| \geq \frac{n}{4}$. Let $T(n)$ be the time to compute the median of $n$ distinct elements. Suppose, you are given a subroutine that outputs an approximate median of $n$ distinct elements in $T(n/5) + O(n)$ time.

- Describe an algorithm to find the median of $A$ in linear time.
- Analyze the time complexity of your algorithm.
- Prove the correctness of it.

[6 + (3 + 3 + 3) = 15]

2. Given an array $A$ of distinct elements and an element $x$ in $A$, `partition`$(A, x)$ is a subroutine that rearranges the elements of $A$ in-place such that all the elements smaller than $x$ are to the left of $x$ and all the elements larger than $x$ are to the right of $x$.

- Design a linear (in the size of the array) time algorithm to implement `partition` that can use only $O(1)$ additional space.
- Prove the correctness of your algorithm.

[7 + 8 = 15]

3. The *longest increasing subsequence* problem aims to find a substring of a given string in which the elements of the substring are sorted in alphabetical order. For example, the longest increasing subsequence of ALGORITHM is A-L-O-R-T.

- Given a sequence of size $n$, design a dynamic programming algorithm to find a longest increasing subsequence in time that is polynomial in $n$.

  *Note: You are asked to output a longest sequence, not the length of it.*
- Prove the correctness of your algorithm.

[10 + 5 = 15]

4. (a) Let $G$ be a simple weighted graph where $G$ can have negative weight edges but does not have any negative weight cycles. Why is Dijkstra's algorithm not guaranteed to succeed to find a shortest path in such a graph $G$?

   (b) Describe an efficient algorithm to find a shortest path in such a graph $G$.

   (c) Prove the correctness of your algorithm.

[4 + 6 + 5 = 15]

5. (a) Describe the Prim's algorithm to find a minimum spanning tree in a weighted graph.

   (b) Prove the correctness of the algorithm.

[6 + 9 = 15]

6. (a) What is an NP-complete problem?

   (b) A *k-clique* in a graph is a subset of vertices of size $k$ where every pair of vertices is connected. Given a graph and an integer $1 \leq k \leq n$ as input, prove that the problem of deciding if the graph has a $k$-clique is NP-complete assuming 3SAT is NP-complete.

   (c) A *k-independent set* of a graph is a subset of vertices of size $k$ such that no two vertices share an edge. Given a graph and an integer $1 \leq k \leq n$ as input, prove that the problem of deciding if the graph has a $k$-independent set is NP-complete.

[4 + 7 + 4 = 15]

## Group B

*Note: Answer any one of questions 1 and 2 of 10 marks, and any one of questions 3 and 4 of 15 marks.*

1. Consider the following recurrence:

$$T(n) = \begin{cases} 1 & n < 2, \\ 2T(n/2) + O(\log n) & n \geq 2. \end{cases}$$

Construct the recursion tree for this recurrence and use it to show that $T(n) = O(n)$.

**OR,**

2. Design a polynomial-time algorithm for finding a Hamiltonian path in a directed acyclic graph. Comment about the time complexity and the correctness of your algorithm.

[10]

3. Design a dynamic programming algorithm to find an independent set in a *tree* in time that is polynomial in the size of the tree. Comment about the time complexity and the correctness of your algorithm.

**OR,**

4. Consider a graph $G = (V, E)$ where $|V| = n$. Fix an integer $1 \leq k \leq n$. Let $C$ be a set of colors of size $k$ and $c : V \to C$ be a coloring of the vertices of $G$. We say $G$ has a *colorful* $k$-path if $G$ has a path of length $k$ (number of vertices) where no two vertices on the path have the same color. Design a $O(2^k \cdot n)$ time algorithm to decide whether $G$ has a colorful $k$-path. Discuss the correctness of your algorithm.

[15]