

Problem Set Sample

Due on: Never

Instructions

- For your proofs, you may use any lower bound, algorithm or data structure from the text or in class, and their correctness and analysis, but please cite the result that you use.
- Each problem is worth 10 points. Remember that for Problems 2-4, you will not get full credit unless your solution is the most efficient one in terms of running time.
- If you do not prove that your algorithm is correct, we will assume that it is incorrect. If you do not provide an analysis of the running time, we will assume you do not know what the running time is.

Problem 1

You are given a set of n variables x_1, x_2, \dots, x_n and a set of m_1 equality constraints of the form $x_i = x_j$ and a set of m_2 inequality constraints of the form $x_i \neq x_j$. Is it possible to satisfy all of them? For example, it is impossible to satisfy the constraints: $x_1 = x_2, x_2 = x_3, x_1 \neq x_3$.

Design an algorithm that takes as input the $m_1 + m_2$ constraints and decides whether the constraints can be satisfied. Prove that your algorithm is correct, and provide an analysis of its running time.

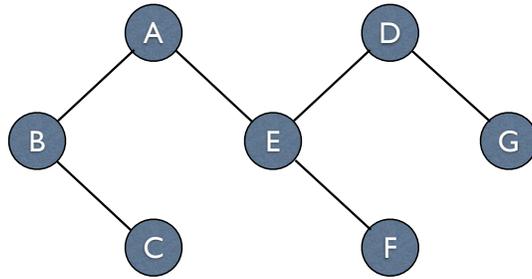
Problem 2

A traveling salesman is getting ready for a big tour. Starting at his hometown he will conduct a journey in which each of his target cities is visited exactly once before he returns home. His problem is as follows: given the pairwise distances between all pairs of cities he will visit, what is the best order in which to visit them, so as to minimize the overall distance traveled?

Describe and analyze an algorithm to solve the traveling salesman's problem in $O(2^n \text{poly}(n))$ time. You are given an undirected n -vertex graph G with weighted edges, where each node in G is a city, and the weight of an edge (u, v) is the distance between cities u and v . Your algorithm should return the weight of the lightest traveling salesman tour in G . [Hint: The obvious recursive algorithm takes $O(n!)$ time.]

Problem 3

A vertex cover of a graph $G = (V, E)$ is a subset of vertices $S \subseteq V$ that includes at least one endpoint of every edge in E . For instance, in the following tree, possible vertex covers include $\{A, B, C, D, E, F, G\}$ and $\{A, C, D, F\}$, but not $\{C, E, F\}$. The smallest vertex cover has size 3: $\{B, E, G\}$.



Give a linear-time algorithm for the following task: given an input an undirected tree $T = (V, E)$, find the smallest vertex cover of T .