**CSCI8080 Final Exam Spring 2020**

**Problem 1 (10 points)**

Company TR’s employees are organized in a strict hierarchy with the CEO as the root of the tree. The children of a node N are all supervised by N.

Each employee E in TR is assigned a positive number, EV[E], that measures how valued he/she is, in TR. We want to find a set S of employees that with the total maximum T value with the following conditions: (i) The CEO is always in the set regardless of her value, and (ii) If an employee is in the set, then her immediate supervisor is not in the set.

Design an algorithm that computes S and T given the employee hierarchy of TR using a dynamic programming based approach.

Notes

* Probably going to start with only CEO and one other employee, build from there?
* Optimal substructure
  + Every subtree solution (which employees I take from this subtree) must also be the total max T value for that subtree. Otherwise you could

**Problem 2 (10 points)**

Let T be the minimum spanning tree of a graph G. Prove or disprove the following two statements.

1. T will not contain the maximum weighted edge on any cycle in G.
2. T will contain the minimum weighted edge of every cycle in G.

Notes

* MST = all vertices connected, without cycles, using minimum weight
* Cycle in G = all vertice
* A

Answers

1. True, assuming this is an undirected graph with a cycle that means that there are 2 ways (edges) to get to every node. Given that every node will be included in the minimum spanning tree in the end and there are 2 ways (edges) to get to every node the MST must always use the lighter edge, by definition. This can be proven by contradiction. Let’s assume we have a MST that includes the max weighted edge on a cycle in the graph. We use this edge to connect to some node N. If instead we were able to connect to node N using a lighter edge (the other edge that connects node N to the graph that we know is there because a cycle exists and is lighter because the other edge is the maximum weight edge in the cycle) then we would end up with a spanning tree that is lighter than the MST we assumed before. This obviously cannot happen meaning that any MST will not contain the maximum weighted edge on any cycle in G.

**Problem 3 (5 points)**

Rewrite the Faster-APSP algorithm to include the predecessor matrix computation. Explain the modifications you made.

Notes

* Predecessor matrix
  + For every vertex pair a, b, mark what vertex came right before b on the shortest path from a to b

Answers

**Problem 4 (5 points)**

Suppose you are given a magic black box that can determine in polynomial time, given an arbitrary Boolean formula α, whether α is satisfiable.Describe and analyze a polynomial-timealgorithm that either computesa satisfying assignment for a given Boolean formula or correctly reportsthat no such assignment exists, using the magic black box as a subroutine.

**Problem 5 (10 points)**

We define the 2Sol-SAT problem as follows.

Input: A, an instance of a SAT formula (A is a conjunction of disjunctive clauses).

Output: 1 if A has at least two satisfying solutions, otherwise, 0.

(a) Show that SAT ≤p 2Sol-SAT.

(b) Show that the 2Sol-SAT problem is in NP.

Notes

* A
  + Map inputs, verify outputs match
* B
  + Show it can be verified in p-time

Answers

1. Slkd
2. In order to show that the 2Sol-SAT problem is in NP we need to show that it can be verified in polynomial time.

**Problem 6 (10 points)**

A Hamiltonian Cycle in a graph is a cycle that visits every vertex exactly once. DIRECTED-HAMILTONIANC problem checks to see if a directed graph contains a Hamiltonian cycle. UNDIRECTED-HAMILTONIANC problem does the same for undirected graphs.

1. Describe a polynomial-time reduction from UNDIRECTED-HAMILTONIANC to DIRECTED-HAMILTONIANC.
2. Describe a polynomial-time reduction from DIRECTED-HAMILTONIANC to UNDIRECTED-HAMILTONIAN PATH.

Notes

* Hamiltonian cycle = visits each vertex exactly once
* B
  + Take directed graph, represent it as undirected graph

Answers

1. A polynomial time reduction from UNDIRECTED-HAMILTONIANC to DIRECTED-HAMILTONIANC involves mapping the inputs of UNDIRECTED-HAMILTONIANC to the inputs of DIRECTED-HAMILTONIANC in polynomial time and then showing that the outputs match for both problems given the original input and the mapping function to the other problem.
   1. You can map the input of UNDIRECTED-HAMILTONIANC (a graph, G, with edges, E, and vertices, V) to UNDIRECTED-HAMILTONIANC (a graph, G’, with edges, E’, and vertices, V’) by simply taking every undirected edge (u, v) in E and creating a directed edge in both directions such that you end up with (u, v) and (v, u). The set of vertices would remain the same: G’ = {E’, V} such that E’ contains (u, v) and (v, u) for every edge (u, v) in the undirected edges in E. This can be done in polynomial time since you only need to iterate over the edges in E.
   2. DIRECTED-HAMILTONIANC will only return true given graph G’, as defined above, IFF UNDIRECTED-HAMILTONIANC will return true given graph G. This is true because the resulting directed graph, G’, is equivalent to the undirected graph G since an undirected edge can be represented as 2 directed edges, one in both directions. So saying there is a Hamiltonian cycle in an undirected graph G is the same thing as saying there is a Hamiltonian cycle in another directed graph where we represent each undirected edge as 2 directed edges, one in both directions.
2. A polynomial time reduction from DIRECTED-HAMILTONIANC to UNDIRECTED-HAMILTONIANC will involve the same steps as above but going in the opposite direction. We will need to map the inputs from DIRECTED-HAMILTONIANC (a directed graph) to UNDIRECTED-HAMILTONIANC (an undirected graph) and then show that UNDIRECTED-HAMILTONIANC returns true given a graph G’ IFF DIRECTED-HAMILTONIANC returns true given graph G.
   1. One way to map