Problem Set 13

Data Structures and Algorithms, Fall 2020

Due: Dec 24, in class.

Problem 1

(a) A language L is *complete* for a language class $\mathcal{C}$ with respect to polynomial-time reductions if $L \in \mathcal{C}$ and $L' \leq_P L$ for all $L' \in \mathcal{C}$. Show that $\emptyset$ and $\{0,1\}^*$ are the only languages in $\mathbf{P}$ that are not complete for $\mathbf{P}$ with respect to polynomial-time reductions.

(b) Show that, with respect to polynomial-time reductions, L is complete for $\mathbf{NP}$ if and only if $\overline{L}$ is complete for $\mathbf{coNP}$.

Problem 2

(a) Suppose that someone gives you a polynomial-time algorithm to decide boolean formula satisfiability. Describe how to use this algorithm to find a satisfying assignment in polynomial time. You need to argue your algorithm indeed runs in polynomial time.

(b) We have shown in class that 3-SAT is NP-complete. However, it is known that 2-SAT $\in \mathbf{P}$, where 2-SAT is the language containing all satisfiable boolean formulas in CNF with exactly 2 literals per clause. Prove that 2-SAT $\in \mathbf{P}$ by giving a polynomial time algorithm. You need to argue your algorithm indeed runs in polynomial time.

Problem 3

Given an integer $m \times n$ matrix A and an integer m-vector b, the 0-1 integer programming problem asks whether there exists an integer n-vector x with elements in the set $\{0,1\}$ such that $Ax \leq b$. Prove that 0-1 integer programming is NP-complete. *(Hint: reduce from* 3-SAT.*)*

Problem 4

(a) Suppose that, in addition to edge capacities, a flow network has vertex capacities too. That is each vertex v has a limit $l(v)$ on how much flow can pass though v. Show how to transform a flow network $G = (V, E)$ with vertex capacities into an equivalent flow network $G' = (V', E')$ without vertex capacities, such that a maximum flow in G' has the same value as a maximum flow in G. How many vertices and edges does G' have?

(b) Suppose that you are given a flow network G, and G has edges entering the source s. Let f be a flow in G in which one of the edges (v, s) entering the source has $f(v, s) = 1$. Prove that there must exist another flow f' with $f'(v, s) = 0$ such that $|f| = |f'|$. Give an $O(|E|)$-time algorithm to compute f', given f, and assuming that all edge capacities are integers.

(c) Let $G = (V, E)$ be a bipartite graph with vertex partition $V = L \cup R$, and let G' be its corresponding flow network. Give a good upper bound (that is, as large as possible) on the length of any augmenting path found in G' during the execution of the Ford-Fulkerson algorithm. You need to give an example to justify your bound can be attained. *(Hint: a constant is not the desired answer.)*

Problem 5

The edge connectivity of an undirected graph is the minimum number of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show how to determine the edge connectivity of an undirected graph $G = (V, E)$ by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(|V|)$ vertices and $O(|E|)$ edges. You need to argue the correctness of your algorithm.

Problem 6

You are helping arranging the work schedules of doctors in a hospital. In particular, you need to ensure there is at least one doctor covering each vacation day.

More specifically, there are k vacation periods (e.g., the week of Christmas, the July 4$^{\text{th}}$ weekend, the Thanksgiving weekend), each spanning several contiguous days. Let D_j be the set of days included in the j^{th} vacation period. We refer to the union of all these days, $\bigcup_j D_j$, as the set of all vacation days.

There are n doctors at the hospital, and doctor i has a set of vacation days S_i when he or she is available to work. (This may include certain days from a given vacation period but not others; so, for example, a doctor may be able to work the Friday, Saturday, or Sunday of Thanksgiving weekend, but not the Thursday.)

Design and analyze an algorithm that takes this information and determines whether it is possible to select a single doctor to work on each vacation day, subject to the following two constraints:

- For a given parameter c, each doctor should be assigned to work at most c vacation days total, and only days when he or she is available.
- For each vacation period j, each doctor should be assigned to work at most one of the days in the set D_j. (In other words, although a particular doctor may work on several vacation days over the course of a year, he or she should not be assigned to work two or more days of the Thanksgiving weekend, or two or more days of the July 4$^{\text{th}}$ weekend, etc.)

Problem 7

Suppose you are given a flow network $G = (V, E)$ with integer edge capacities and an integer-valued maximum flow f in G. Describe and analyze algorithms for each of the following two operations:

(a) `Increment`(e): Increase the capacity of edge e by 1 and update the maximum flow.

(b) `Decrement`(e): Decrease the capacity of edge e by 1 and update the maximum flow.

Both algorithms should modify f so that it is still a maximum flow, more quickly than recomputing a maximum flow from scratch. In particular, each algorithm should run in $O(|V| + |E|)$ time.