Due April 4, 11:59pm

1. (20 pts.) Minimum Cost Flows

In the max flow problem, we just wanted to see how much flow we could send between a source and a sink. But in general, we would like to model the fact that shipping flow takes money. More precisely, we are given a directed graph G with source s, sink t, costs l_e, capacities c_e, and a flow value F. We want to find a nonnegative flow f with minimum cost, that is $\sum_e l_e f_e$ that respects the capacities and ships F units of flow from s to t.

(a) Show that the minimum cost flow problem can be solved in polynomial time by reducing it to LP.

(b) Solve the shortest path problem using the minimum cost flow problem.

(c) Solve the maximum flow problem using the minimum cost flow problem.

2. (20 pts.) Linear program for shortest path, using flows

Suppose we want to compute the shortest path from node s to node t in a directed graph with edge lengths $l_e > 0$.

(a) Show that this is equivalent to finding an $s-t$ flow f that minimizes $\sum_e l_e f_e$ subject to $size(f) = 1$. There are no capacity constraints.

(b) Write the shortest path problem as a linear program. Explain your variables, your constraints, and your objective function.

(c) Show that the dual LP can be written as

$$\begin{aligned} &\max x_s - x_t \\ &\text{s.t. } x_u - x_v \le l_{uv}, \forall (u,v) \in E \end{aligned}$$

3. (20 pts.) Circular Reductions

Given a set S of non-negative integers $[a_1, a_2, \ldots, a_n]$, consider the following problems:

- **Partition**: Determine whether there is a subset $P \subseteq S$ such that $\sum_{i \in P} a_i = \sum_{j \in S \setminus P} a_j$
- **Subset Sum**: Given some integer K, determine whether there is a subset $P \subseteq S$ such that $\sum_{i \in P} a_i = K$

(a) Find a linear time reduction from Partition to Subset Sum. Then prove your reduction is correct.

(b) Find a linear time reduction from Partition to Knapsack (for Knapsack, refer to pages 164-168 of textbook). Then, prove your reduction is correct.

(c) Find a linear time reduction Subset Sum to Partition. Then prove your reduction is correct.
(Hint: think about adding certain elements to the set S)

4. **(20 pts.) Decision vs. Search vs. Optimization**
The following are three formulations of the VERTEX COVER problem:

- As a *decision problem*: Given a graph G, return TRUE if it has a vertex cover of size at most b, and FALSE otherwise.
- As a *search problem*: Given a graph G, find a vertex cover of size at most b (that is, return the actual vertices), or report that none exists.
- As an *optimization problem*: Given a graph G, find a minimum vertex cover.

At first glance, it may seem that search should be harder than decision, and that optimization should be even harder. We will show that, up to polynomial factors, they actually have the same difficulty:

Describe your algorithms precisely; justify correctness and running time. No pseudocode.

Hint for both parts: Call the black box more than once.

(a) Suppose you are handed a black box that solves VERTEX COVER (DECISION) in polynomial time. Give an algorithm that solves VERTEX COVER (SEARCH) in polynomial time.

(b) Similarly, suppose we know how to solve VERTEX COVER (SEARCH) in polynomial time. Give an algorithm that solves VERTEX COVER (OPTIMIZATION) in polynomial time.

5. **(15 pts.) Bipartite Vertex Cover**
A vertex cover of an undirected graph $G = (V, E)$ is a subset of the vertices which touches every edge. In other words, a subset $S \subset V$ such that for each edge $\{u, v\} \in E$, one or both of u, v are in S.

Show that the problem of finding the minimum vertex cover in a bipartite graph reduces to maximum flow. Prove that your reduction is correct.