CS 170 Homework 10 (OPTIONAL)

Due **11/6/2023, at 10:00 pm (grace period until 11:59pm)**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write "none".

2 A Cohort of Secret Agents

A cohort of k secret agents residing in a certain country needs escape routes in case of an emergency. They will be travelling using the railway system which we can think of as a directed graph $G = (V, E)$ with V being the cities and E being the railways. Each secret agent i has a starting point $s_i \in V$, and all s_i's are distinct. Every secret agent needs to reach the consulate of a friendly nation; these consulates are in a known set of cities $T \subseteq V$. In order to move undetected, the secret agents agree that at most c of them should ever pass through any one city. Our goal is to find a set of paths, one for each of the secret agents (or detect that the requirements cannot be met).

Model this problem as a flow network. Specify the vertices, edges, and capacities; show that a maximum flow in your network can be transformed into an optimal solution for the original problem. You do not need to explain how to solve the max-flow instance itself.

3 Running a Gym

You are running a gym and need to hire trainers for it. Here are the constraints:

- There are D days in total.
- There are n instructors for hire. The i^{th} instructor is available for hire from day a_i to day b_i. The numbers $\{a_i, b_i\}$ for $i = 1 \ldots n$ are given as input.
- Each instructor can be hired for their entire period of availability, i.e. if we hire the i^{th} instructor, then they would be at the gym every day from day a_i to day b_i.
- There should be **exactly** k instructors at the gym every day.

Your goal is to determine which instructors to hire so that the above constraints are met. Devise an efficient algorithm that solves the problem by constructing a directed graph G with a source s and sink t, and computing the maximum flow from s to t. Proof of Correctness is not required.

4 Weighted Rock-Paper-Scissors

You and your friend used to play rock-paper-scissors, and have the loser pay the winner 1 dollar. However, you then learned in CS170 that the best strategy is to pick each move uniformly at random, which took all the fun out of the game.

Your friend, trying to make the game interesting again, suggests playing the following variant: If you win by beating rock with paper, you get 2 dollars from your opponent. If you win by beating scissors with rock, you get 1 dollars. If you win by beating paper with scissors, you get 4 dollar.

Feel free to use an online LP solver to solve your LPs in this problem. Here is an example of an online solver that you can use: `https://online-optimizer.appspot.com/`.

(a) Draw the payoff matrix for this game. Assume that you are the maximizer, and your friend is the minimizer.

(b) Write an LP to find the optimal strategy in your perspective.

Your friend now wants to make the game even more interesting and suggests that you assign points based on the following payoff matrix:

		Your friend: rock	paper	scissors
You:	rock	-10	3	3
	paper	4	-1	-3
	scissors	6	-9	2

(c) Write an LP to find the optimal strategy for yourself. What is the optimal strategy and expected payoff?

(d) Now do the same for your friend. What is the optimal strategy and expected payoff? How does the expected payoff compare to the answer you get in part (c)?

5 Domination

In this problem, we explore a concept called *dominated strategies*. Consider a zero-sum game with the following payoff matrix for the row player:

Column:

		A	B	C
	D	1	2	-3
Row:	E	3	2	-2
	F	-1	-2	2

(a) If the row player plays optimally, can you find the probability that they pick D without directly solving for the optimal strategy? Justify your answer.

Hint: How do the payoffs for the row player picking D compare to their payoffs for picking E?

(b) Given the answer to part a, if the both players play optimally, what is the probability that the column player picks A? Justify your answer.

(c) Given the answers to part a and b, what are both players' optimal strategies?

Note: All parts of this problem can be solved without using an LP solver or solving a system of linear equations.

6 Decision vs. Search vs. Optimization

Recall that a vertex cover is a set of vertices in a graph such that every edge is adjacent to at least one vertex in this set.

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 if any one can be solved in polynomial time, so can the others.

(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.

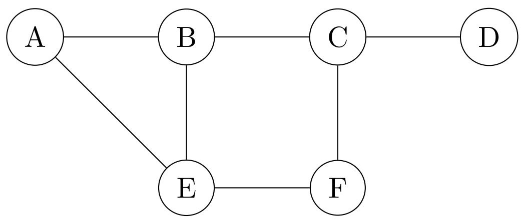
(b) Similarly, suppose we know how to solve VERTEX COVER (SEARCH) in polynomial time. Give an algorithm that solves VERTEX COVER (OPTIMIZATION) in polynomial time.

7 Vertex Cover to Set Cover

Vertex Cover: given an undirected unweighted graph $G = (V, E)$, a vertex cover C_V of G is a subset of vertices such that for every edge $e = (u, v) \in E$, at least one of u or v must be in the vertex cover C_V.

Set Cover: given a universe of elements U and a collection $\mathcal{S} = \{S_1, \dots, S_m\}$, a set cover is any (sub)collection of sets C_S whose union equals U.

In the *minimum vertex cover problem*, we are given an undirected unweighted graph $G = (V, E)$, and are asked to find the smallest vertex cover. For example, in the following graph, $\{A, E, C, D\}$ is a vertex cover, but not a minimum vertex cover. The minimum vertex covers are $\{B, E, C\}$ and $\{A, E, C\}$.



Then, recall in the *minimum set cover problem*, we are given a set U and a collection $S_1, \dots, S_m$ of subsets of U, and are asked to find the smallest set cover. For example, given $U := \{a, b, c, d\}$, $S_1 := \{a, b, c\}$, $S_2 := \{b, c\}$, and $S_3 := \{c, d\}$, a solution to the problem is $C_S = \{S_1, S_3\}$.

Give an efficient reduction from the minimum vertex cover problem to the minimum set cover problem. Proof of correctness is not required.

8 Min Cost Flow

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.

(b) Describe a reduction from the shortest path problem to the minimum cost flow problem. Proof of correctness is not required.

(c) Describe a reduction from the maximum flow problem to the minimum cost flow problem. Proof of correctness is not required.