CS 170 Homework 9

Due **Saturday 11/1/2025, at 10:00 pm (grace period until 11:59pm)**

1 Study Group (1 point)

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

2 Gaps in proof of Max-Flow = Min-Cut Theorem (12 points)

In lecture, we saw a proof of the *Max-Flow = Min-Cut Theorem*, but two of the steps were somewhat hand-wavy. In this problem, we will fill in these gaps and formalize these steps.

Let $G = (V, E)$ be a directed graph with capacities $c_e \in \mathbb{Z}^+$ for each edge $e \in E$. Let $s, t \in V$ denote the source and sink vertices of G, respectively. Assume that s has no incoming edges and t has no outgoing edges. Recall that a flow $f = \{f_e\}_{e \in E}$ in G is a collection of nonnegative values satisfying the equation

$$\sum_{v \in V} f_{u,v} = \sum_{v \in V} f_{v,u}$$

for every vertex $u \in V \setminus \{s, t\}$. Moreover, the flow also satisfies the inequality $f_e \leq c_e$ for every edge $e \in E$. Recall also from lecture that the size of the flow is defined to be $\text{size}(f) = \sum_{v \in V} f_{s,v}$.

Given an s–t cut (L, R) in G, define the *net flow from L to R* to be the total outgoing flow from L to R minus the reverse flow from R to L. Namely, the net flow from L to R is defined to be the value

$$\text{flow}(L, R) = \sum_{u \in L} \sum_{v \in R} f_{u,v} - \sum_{u \in L} \sum_{v \in R} f_{v,u}.$$

(a) (5 points) Show that for any s–t cut (L, R) and flow f, we have $\text{size}(f) = \text{flow}(L, R)$.

Hint: consider inducting on $|L|$. What is the base case? How does $flow(L, R)$ change when moving a vertex from L to R?

(b) (3 points) Show that for any s–t cut (L, R) and flow f, we have $\text{size}(f) \leq \text{capacity}(L, R)$.

(c) (4 points) Let (L, R) be the s–t cut constructed in the proof of the Max-Flow = Min-Cut Theorem from lecture, and let f be the flow outputted by the Ford-Fulkerson algorithm. Show that $\text{size}(f) = \text{capacity}(L, R)$.

3 Flow vs. LP (11 points)

You play a middleman in a market of m suppliers and n purchasers. The i-th supplier can supply up to $s[i]$ products, and the j-th purchaser would like to buy up to $b[j]$ products.

However, due to legislation, supplier i can only sell to a purchaser j if they are situated at most 1000 miles apart. Assume that you're given a list L of all the pairs (i, j) such that supplier i is within 1000 miles of purchaser j. Given m, n, $s[1, \ldots, m]$, $b[1, \ldots, n]$, and L as input, your job is to compute the maximum number of products that can be sold. The runtime of your algorithm must be polynomial in m and n.

For parts (a) and (b), assume the product is divisible. That is, it is fine to sell a fraction of a product.

(a) (4 points) Show how to solve this problem using a network flow algorithm as a subroutine. Describe the graph and explain why the output from running the network flow algorithm on your graph gives a valid solution to this problem.

(b) (4 points) Formulate this problem as a linear program. Explain why this linear program correctly solves the problem and can be solved in polynomial time.

(c) (3 points) Now let us assume you *cannot* sell a fraction of a product. In other words, the number of products sold by each supplier to each purchaser must be an integer. Which formulation is more amenable to finding integral solutions: network flow or linear programming? Explain your answer.

4 A cohort of secret agents (9 points)

A cohort of k secret agents residing in a certain country needs escape routes in case of an emergency. They will be traveling 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.

5 Domination (10 points)

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:

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

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

(a) (4 points) 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.

(b) (3 points) Given the answer to part (a), if both players play optimally, what is the probability that the column player picks A? Justify your answer.

(c) (3 points) Given the answers to parts (a) and (b), what are both players' optimal strategies?

6 Weighted Rock-Paper-Scissors (9 points)

For this problem only, you are allowed to use code or online tools to automatically solve your LPs. Feel free to use an LP solver such as: `https://online-optimizer.appspot.com/`. However, **make sure to cite your LP solver** if you use one.

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 dollar. If you win by beating paper with scissors, you get 4 dollars.

(a) (2 points) Draw the payoff matrix for this game. Assume that you are the maximizer, and your friend is the minimizer.

(b) (3 points) Write an LP to find the optimal strategy in your perspective. You do not need to solve the LP.

The following subparts are independent of the previous subparts.

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) (2 points) Write an LP to find the optimal strategy for yourself. What is the optimal strategy and expected payoff? Feel free to use an online LP solver.

(d) (2 points) 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)?