CS 170 Homework 6

Due **Saturday 10/11/2025, 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 Arbitrage (11 points)

Shortest-path algorithms can also be applied to currency trading. Suppose we have n currencies $C = \{c_1, c_2, \ldots, c_n\}$: e.g., dollars, Euros, bitcoins, dogecoins, etc. For any pair of currencies c_i, c_j, there is an exchange rate $r_{i,j}$: you can buy $r_{i,j}$ units of currency c_j at the price of one unit of currency c_i. Assume that $r_{i,i} = 1$ and $r_{i,j} \geq 0$ for all i, j.

The Foreign Exchange Market Organization (FEMO) has hired Oski, a CS170 alumnus, to make sure that it is not possible to generate a profit through a cycle of exchanges; that is, for any currency $i \in C$, it is not possible to start with one unit of currency i, perform a series of exchanges, and end with more than one unit of currency i. (That is called *arbitrage.*)

More precisely, arbitrage is possible when there is a sequence of currencies $c_{i_1}, \ldots, c_{i_k}$ such that $r_{i_1,i_2} \cdot r_{i_2,i_3} \cdot \cdots \cdot r_{i_{k-1},i_k} \cdot r_{i_k,i_1} > 1$. This means that by starting with one unit of currency c_{i_1} and then successively converting it to currencies $c_{i_2}, c_{i_3}, \ldots, c_{i_k}$ and finally back to c_{i_1}, you would end up with more than one unit of currency c_{i_1}. Such anomalies last only a fraction of a minute on the currency exchange, but they provide an opportunity for profit.

We say that a set of exchange rates is arbitrage-free when there is no such sequence, i.e., it is not possible to profit by a series of exchanges.

(a) (7 points) Give an efficient algorithm for the following problem: given a set of exchange rates $(r_{i,j})_{i,j \in n}$ which is *arbitrage-free*, and two specific currencies a, b, find the most profitable sequence of currency exchanges for converting currency a into currency b. That is, if you have a fixed amount of currency a, output a sequence of exchanges that gets you the maximum amount of currency b.

Hint: $\log(xy) = \log(x) + \log(y)$.

(i) Describe your algorithm.

(ii) Prove the correctness of your algorithm.

(iii) Analyze the runtime of your algorithm.

(b) (4 points) Oski is fed up with manually checking exchange rates and has asked you for help to write a computer program to do his job for him. Give an efficient algorithm for detecting the possibility of arbitrage.

(i) Describe your algorithm.

(ii) Analyze the runtime of your algorithm.

3 Happy trees (13 points)

Let $G = (V, E)$ be an undirected graph with edge weights $\{w_e\}_{e \in E}$ **that are all distinct**. Call an edge $e \in E$ *happy* if all cycles containing e have an edge with weight greater than w_e.

(a) (5 points) We first show that the MST of G is unique and, in particular, is made up of all the happy edges of G.

(i) Given an MST T, prove that all its edges are happy.

(ii) Given an MST T, prove that all happy edges in G are in T.

(b) (3 points) True or False: The minimum spanning tree of G includes the minimum-weight edge in every cycle in G. If true, write a proof. If false, draw a counterexample with at most 5 vertices, and clearly label (i) the MST T and (ii) a cycle in which the minimum-weight edge is not included in T.

(c) (5 points) Suppose we compute the MST T on G, and then afterwards realized that we had the incorrect weight w_e for an edge $e \in T$ and that its true weight is $w'_e > w_e$. You should assume this new w'_e is distinct from all the edge weights in $\{w_e\}_{e \in E}$.

(i) Give an algorithm for computing the MST T' under the true weights by only removing at most one edge and adding at most one edge to T.

(ii) Prove that your algorithm is correct.

Hint: First show that all other edges in G that were previously happy will still be happy.

4 Jordan's picnic planning fiasco (10 points)

The CS 170 course staff is planning a trip to Ginger Island. Originally, Jordan was in charge of seating assignments in cars for all members of the course staff. He arranged p people in n cars where car i has capacity $c_i \forall i \in 1, \ldots, n$.

However, given Jordans poor planning skills, he assigned people in cars in a terribly inefficient manner, leaving a lot of empty space in many cars. Given the number of people p, and the capacities of each of the n cars $c_1, \ldots, c_n$, come up with an efficient greedy algorithm to fit people in cars such that the CS 170 course staff can take the least number of cars possible on the road trip.

Prove the optimality of your algorithm using an exchange argument. No runtime analysis required.

5 Nina's ribbon (10 points)

Nina has a roll of ribbon that is n feet long and an array of nondecreasing nonnegative integers that contains prices of all lengths of size at most n feet. She wants to find the maximum value she can make by cutting up the roll of ribbon and selling the pieces. For example, if the length of the ribbon roll is 8 feet and the values of different pieces are given as follows, then the maximum obtainable value is 22 (by cutting into two pieces of lengths 2 and 6).

length	1	2	3	4	5	6	7	8
price	1	5	8	9	10	17	17	20

Give an efficient dynamic programming algorithm so Nina can find the maximum obtainable value given any ribbon roll length and set of prices.

(a) (7 points) Describe your algorithm.

(b) (3 points) Analyze the runtime of your algorithm.