CS 170 HW 5 (Optional)

Due 2020-02-04, at 10:00 pm
You may submit your solutions if you wish them to be graded, but they will be worth no points

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 Arbitrage

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 \( i, j \) of currencies, 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 \( c_{i_1}, c_{i_2}, \ldots, c_{i_k} \) such that \( r_{i_1,i_2} \cdot r_{i_2,i_3} \cdot \cdots \cdot r_{i_k,i_1} > 1 \). 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) Give an efficient algorithm for the following problem: given a set of exchange rates \( r_{i,j} \) which is arbitrage-free, and two specific currencies \( s, t \), find the most advantageous sequence of currency exchanges for converting currency \( s \) into currency \( t \).

Hint: represent the currencies and rates by a graph whose edge weights are real numbers.

(b) Oski is fed up of 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. You may use the same graph representation as for part (a).

3 Bounded Bellman-Ford

Modify the Bellman-Ford algorithm to find the weight of the lowest-weight path from \( s \) to \( t \) with the restriction that the path must have at most \( k \) edges.
4 Money Changing.

Fix a set of positive integers called denominations $x_1, x_2, \ldots, x_n$ (think of them as the integers 1, 5, 10, and 25). The problem you want to solve for these denominations is the following: Given an integer $A$, express it as

$$A = \sum_{i=1}^{n} a_i x_i$$

for some nonnegative integers $a_1, \ldots, a_n \geq 0$.

1. Under which conditions on the denominations $x_i$ are you able to do this for all integers $A > 0$?

2. Suppose that you want, given $A$, to find the nonnegative $a_i$’s that satisfy $A = \sum_{i=1}^{n} a_i x_i$, and such that the sum of all $a_i$’s is minimal —that is, you use the smallest possible number of coins. Define a greedy algorithm for this problem. (Your greedy algorithm may not necessarily solve the problem, i.e., it may fail on some inputs)

3. Show that the greedy algorithm finds the optimum $a_i$’s in the case of the denominations 1, 5, 10, and 25, and for any amount $A$.

4. Give an example of a denomination where the greedy algorithm fails to find the optimum $a_i$’s for some $A$. Do you know of an actual country where such a set of denominations exists?

5. How far from the optimum number of coins can the output of the greedy algorithm be, as a function of the denominations?