CS 161 W19: Recitation 7 Problems

February 2019

Exercise 0

Suppose the various economies of the world use a set of currencies $C_1, \ldots, C_n$ — think of these as dollars, pounds, bitcoins, etc. Your bank allows you to trade each currency C_i for any other currency C_j at an exchange rate r_{ij}, that is, you can exchange each unit of C_i for $r_{ij} > 0$ units of C_j. Due to fluctuations in the markets, it is occasionally possible to find a sequence of exchanges that lets you start with currency A, change into currencies, B, C, D, etc., and then end up changing back to currency A in such a way that you end up with more money than you started with. That is, there are currencies $C_{i_1}, \ldots, C_{i_k}$ such that

$$r_{i_1 i_2} \times r_{i_2 i_3} \times \cdots \times r_{i_{k-1} i_k} \times r_{i_k i_1} > 1.$$

This is called an arbitrage opportunity, but to take advantage of it you need to be able to identify it quickly (before other investors leverage it and the exchange rates balance out again)! Devise an efficient algorithm to determine whether an arbitrage opportunity exists. Justify the correctness of your algorithm and its runtime.

Exercise 1

Suppose we have a rod of length k, where k is a positive integer. We would like to cut the rod into integer-length segments such that we maximize the *product* of the resulting segments' lengths. Multiple cuts may be made. Write an algorithm to determine the maximum product possible.

Exercise 2

Given an 8×8 chessboard and a knight that starts at position $a1$, devise an algorithm that returns how many ways the knight can end up at position xy after k moves. Knights move ± 1 squares in one direction and ± 2 squares in the other direction.

Exercise 3

Pepper and Plucky have been talking. They're working out whether they can modify Dijkstra's algorithm to deal with negative edge weights. Here's what they came up with:

Let $G = (V, E)$ be a weighted graph with negative edge weights, and let w^* be the smallest (most negative) weight that appears in G. Consider a graph $G' = (V, E')$ with the same vertices as G. Then to construct the edges E', we do the following: for every edge $e \in E$ with weight w, we add an edge $e' \in E'$ with weight $w - w^*$. Now all of the weights in G' are non-negative, so we can apply Dijkstra's algorithm such that:

recitation$_7/modifiedDijkstras.png$

Does this suggestion work?

(That is, does it always return a shortest path from s to t in G if it exists?) Either prove that it is correct (that is, prove that this algorithm correctly finds shortest paths in weighted directed graphs), or give a counter-example.