CS 180 Homework 4

March 4, 2015

1 Dynamic Coin Grabbing

(a) Show a sequence of $n \geq 6$ coins for which it is not optimal for the first player to start by picking up the available coin of larger value. That is, give an example for which the natural greedy strategy is suboptimal.

> One such sequence of six coins in which the greedy approach fails is as follows:
>
> $$1 \ 2 \ 3 \ 4 \ 1000 \ 6$$
>
> In this case, the coin grabbing goes as follows:
>
> | First Player | Second Player |
> |:---:|:---:|
> | 6 | 1000 |
> | 4 | 3 |
> | 2 | 1 |
>
> With this sequence of greedy grabbing, the first player ends up with a total of 12 while the second player has a total of 1004.

(b) Give an $O(n^2)$ algorithm to compute an optimal strategy for the first player. Given the initial sequence, your algorithm should precompute some information in $O(n^2)$ time, and then the first player should be able to make each move optimally in $O(1)$ time by looking up the precomputed information.

(c) Find a sequence of $n \geq 6$ coins for which the dynamic programming strategy guarantees the first player a higher total value than the simple strategy

2 Box Stacking

(a) Give an algorithm for box stacking that uses as few split and join steps as possible, when given two stacks of boxes which are represented as ascendingly sorted sequences of numbers (box sizes). Duplicate numbers are permitted in the input sequences, and the two stacks can contain identical numbers. Hint: the goal is to find the optimal split of one of two stacks.

(b) Determine the time complexity of your algorithm.

3 Longest Ascending Subsequence

(a) Give a Dynamic Programming recursion equation for $Len(x, i)$ in terms of $Len(x, 1)$, ... , $Len(x, (i1))$.

(b) Show how to use these recursion equations to obtain a polynomial-time algorithm for solving this problem.

(c) What is the time complexity of this algorithm?.

4 Currency Exchange

(a) Because each entry in the matrix R is a rate, if we take the log of entries in the matrix, then sums of matrix entries are equal to logs of corresponding products of rates. Prove that the weight of a path using the log-transformed matrix $L = \log R$ is the log of the product of rates along that path. Also show the weight of the same path in the matrix $C = L = \log R$ corresponds to the inverse of the rate in L.

(b) Give an efficient algorithm (in pseudocode) to compute the shortest paths from each currency to each other currency using the matrix $C = L = \log R$.

(c) The matrix C can have negative entries, and if we view it as the adjacency matrix of a graph, it can have negative cycles. Any negative cycle in the matrix C will correspond to a gain of money by currency trades and an opportunity for arbitrage. Give an efficient algorithm for finding the maximally negative cycle in the matrix C. (If there is no negative cycle, it should determine that.)