Assignment 3. CS341, Winter 2011

Distributed Thursday, Feb. 3, due Feb. 17, 1pm, 2011. Hand in to the assignment boxes on the 3rd floor of MC, before 1pm.

1. (10 marks) Greedy algorithms for coin systems.

   (a) For Canadian coin system (see lecture notes), prove that the greedy algrorithm always uses the fewest number of coins to make change.

   (b) Give a coin system where the greedy algorithm does not always give the fewest number of coins. (Your coin system should allow change to be made for any amount. And please do not use the example (12,5,1) in the lecture notes)

   (c) For any coin system with coin denomination $c^0, c^1, \ldots, c^k$ for some constants $c, k \in N$, prove that the greedy algorithm uses the fewest number of coins.

2. (10 marks) If A is an $a \times b$ matrix and B is a $b \times c$ matrix then the cost of multiplying A and B, counting arithmetic steps, is abc. Consider the problem of multiplying matrices $M_1, M_2, \ldots, M_n$ where matrix M_i has k_i rows and k_{i+1} columns. Because matrix multiplication is associative, we may choose the order in which we multiply pairs of matrices. Suppose our goal is to minimize the total cost. Find examples to show that the following greedy strategies do not minimize the cost:

   (a) First multiply the adjacent pair of minimum cost.

   (b) First multiply the adjacent pair of maximum cost.

   (c) First multiply the pair M_i and M_{i+1} of maximum k_{i+1}.

3. (10 marks) We are given an undirected tree T of arbitrary degree with n vertices. We would like to find a largest subset S of edges such that no two edges in S are incident to the same vertex. Describe a greedy algorithm to solve this problem in time $O(n^2)$ or better. Carefully prove that your algorithm is correct.

   Hint: start from the leaf level.

4. (10 marks). [Building Bridges] Consider a 2-D map with a horizontal river passing through its center. There are n cities, named $1, \ldots, n$ on the southern bank with x-coordinates $a(1), \ldots, a(n)$, respectively, and n cities, also named $1, \ldots, n$, on the northern bank with x-coordinates $b(1), \ldots, b(n)$, respectively. You want to connect as many north-south pairs of cities as possible with bridges such that no two bridges cross. When connecting cities, you can only connect city i on the northern bank to city i on the southern bank, see Figure 1. Give a dynamic programming formulation to this problem.

5. (10 marks) Consider a row of n coins of values $v(1), \ldots, v(n)$, respectively, where n is even. We play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently,

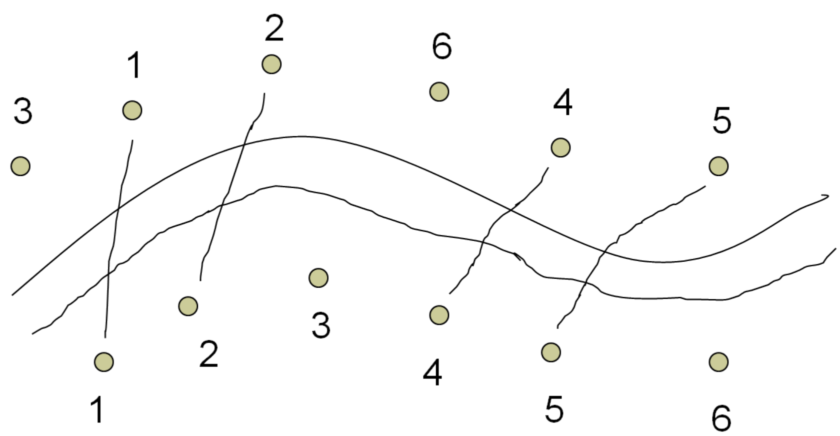

Figure 1: The Building Bridges Problem

and receives the value of the coin. Determine the maximum possible amount of money we can definitely win if we move first. Give the dynamic programming formulation of this problem.