CIT 596 Online **Spring 2020**

Module 9–10

Name: NAME HERE

Questions requiring written answers.

1. Two players play a game where they start with a row of n piles of varied amounts of money. The players take turns and in each turn a player can pocket either the money in the first pile or the last pile in the row of piles that remains. Design an efficient algorithm, which on any given sequence of amounts, determines the maximum amount of money that player 1 can win.

   If n is even, prove that player 1 wins at least half the money available. If n is odd, player 1 actually gets one more pile than player 2. In spite of that, show with a simple example that player 1 can be left with far less than half the total amount.

2. Given an array of integers A, $a_1, a_2, \ldots, a_n$, design an efficient algorithm that finds the length of the longest increasing subsequence of the array. A subsequence of A is a (not necessarily consecutive) sequence of elements from A that can be obtained by deleting, but not rearranging, elements of A. **Hint:** For each valid i and j find the 'best' increasing subsequence of length i among the integers $a_1, a_2, \ldots, a_j$, where the notion of 'best' is for you to define. These will be your subproblems.

3. Consider an assignment with n questions. Each question has marks m and time t associated with it. Find an efficient algorithm to compute the minimum time required to obtain a target of M marks. (Assume that you either get full marks or 0 on each question. (Of course, this is not true in our course!)

4. Give the run-time of the algorithm taught in the lecture video to find a piece-wise linear regression fit for n points using dynamic programming so as to minimize the cost defined in the video:

$$Ck + \sum_{j=1}^{k} \sum_{i=i_j+1}^{i_{j+1}} (y_i - m_j x_i - b_j)^2$$

5. (Chapter 6, problem 13 on page 324)
   The problem of searching for cycles in graphs arises naturally in financial trading applications. Consider a firm that trades shares in n different companies. For each pair $i \neq j$, they maintain a trade ratio r_{ij}, meaning that one share of i trades for r_{ij} shares of j. Here we allow the rate r to be fractional; that is, $r_{ij} = \frac{2}{3}$ means that you can trade three shares of i to get two shares of j.
   A trading cycle for a sequence of shares $i_1, i_2, \cdots, i_k$ consists of successively trading shares in company i_1 for shares in company i_2, then shares in company i_2 for shares i_3, and so on, finally trading shares in i_k back to shares in company i_1. After such a sequence of trades, one ends up with shares in the same company i_1 that one starts with. Trading around a cycle is usually a bad idea, as you tend to end up with fewer shares than you started with. But occasionally, for short periods of time, there are opportunities to increase shares. We will call such a cycle an opportunity cycle, if trading along the cycle increases the number of shares. This happens exactly if the product of the ratios along the cycle is above 1. In analyzing the state of the market, a firm engaged in trading would like to know if there are any opportunity cycles.
   Give a $O(n^3)$ algorithm that finds such an opportunity cycle, if one exists.

6. You want to go from station 1 to station n by rail. The train fare from station i to station j is $F(i, j)$. You want to break your trip up into a series of segments in order to minimize the cost, but each segment must go in the forward direction, i.e., from station i to station j for $i < j$. Design an efficient dynamic programming algorithm for doing this and analyze its running time.