Department of Computer Science **University of San Francisco**

Computer Science 673
Fall 2016
Homework 6:
Dynamic Programming & Greedy Algorithms
Due Friday, October 14th

For dynmamic programming algorithms, you should:

- Define the table – What does each element of the table hold?
- Give a formula for filling in table locations – this should include both a base case and a recursive case.
- Describe the order in which the table will be filled in (a picture is a good idea here)
- Give pseudocode for your algorithm

For greedy algoritms, you describe the algorithm in english, give pseudocode, and the prove the algorithm, proving both optimal substructure and greedy choice.

1. Let $p_1, p_2, ...p_n$ be n programs to be stored on a disk. Program p_i requires s_i kilobytes of storage, and the capacity of the disk is D, where $s_1 + s_2 + \ldots + s_n > D$.

   (a) (6 points) Give an efficient algorithm to determine which programs to store on the disk so that the total number of programs on the disk is maximized. If you give a greedy algorithm, be sure to prove that your algorithm is correct by proving both greedy choice and optimal program substructure. What is the running time of your algorithm, in terms of n?

   (b) (6 points) Give an efficient algorithm to determine which programs to store on the disk so that as much of the disk as possible is utilized. If you give a greedy algorithm, be sure to prove that your algorithm is correct by proving both greedy choice and optimal program substructure. What is the running time of your algorithm, in terms of n?

2. (8 points) Suppose that you are given an n x n checkerboard and a checker. You must move the checker from the bottom edge of the board to the top edge of the board according to the following rule. At each step you may move the checker to one of 3 squares

   1 The square immediately above

   2 The square that is one up and one to the left (but only if the checker is not already in the leftmost column)

3 The square that is one up and one to the right (but only if the checker is not already in the rightmost column)

Each time you move from square x to square y, you receive $p(x, y)$ dollars. You are given $p(x, y)$ for all pairs (x, y) for which a move from x to y is legal. Do not assume that $p(x, y)$ is positive. Give an algorithm that figures out the set of moves that will move the checker from somewhere along the bottom edge to somewhere along the top edge while gathering as many dollars as possible. You algorithm is free to pick any square along the bottom edge as a starting point, and any square along the top edge as a destination in order to maximize the number of dollars gathered along the way. What is the running time of your algorithm?

3. (8 points) Problem 15-6 *Planning a Company Party*

   Professor Stewart is consulting for the president of a corporation that is planning a company party. The company has a hierarchical structure; that is, the supervisor relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality ranking, which is a real number. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend.

   Professor Stewart is given a tree that describes the structure of the corporation, using the left-child, right-sibling representation described in section 10.4. Each node of the tree holds, in addition to the pointers, the name of an employee and that employees conviviality ranking. Describe an algorithm to make up a guest list that maximizes the sum of the conviviality rankings of the guests. Analyze the running time of your algorithm.

4. (8 points) Problem 15-4 **Printing Neatly** *HINT:* This problem is a lot like matrix multiply.