CSE 830: Homework #3: Dynamic Programming

Due Oct. 20th 2011, 10:20am

1. *Making Change*: The natural greedy algorithm for making change of n units using the smallest number of coins is: Include one unit of the highest denomination coin of at most n units, say d units. Repeat to make change of the remaining $n - d$ units. For each type of coinage below, establish whether or not this greedy algorithm always minimizes the number of coins returned in change. Prove or give a counter example.

- **a.** English coinage before the decimalization, which consisted of half-crowns (30 pence), florins (24 pence), shillings (12 pence), sixpence (6 pence), threepence (3 pence), pennies (1 pence), half pennies (1/2 pence), and farthings (1/4 pence).
- **b.** United States coinage, which consists of half dollars (50 cents), quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies.
- **c.** Prehistoric coinage, which includes cowrie shells worth 1, 2, 5, 10, 20, 40, and 50 units.
- **d.** Martian coinage, where the available denominations are 1, p, p^2,..., p^k, where both p and k are positive integers.

2. *String Matching*: Implement the dynamic programming algorithm for approximate string matching (in a language of your choice) and use it to find the alignment with the fewest insertions, deletions, and substitutions between the following pairs of strings:

- **a.** "I have written over his word files" and "My favorite novel is Lord of the Flies"
- **b.** "This is what a high fidelity transmission might look like" and "Th3s is vhat a lowy ffidlity tranZmissi8n m1ght looook lke"
- **c.** For Lewis Carroll fans: "Twas brillig, and the slithy toves did gyre and gimble in the wabe" and "All mimsy were the borogoves and the mome raths outgrabe"

Turn in only the final alignments, but your source code must be available upon request. You may choose to ignore case or not, but be consistent.

3. *Shelving Books*: Consider the problem of storing n books on the shelves of a library. The order of the books is fixed by the cataloging system and so cannot be rearranged. Therefore, we can speak of a book b_i, where $1 \leq i \leq n$, that has a thickness t_i and a height h_i. The length of each bookshelf at this library is L.

- **a.** Suppose all the books have the same height h (i.e. $h = h_i = h_j$ for all i, j) and the shelves are separated by a distance greater than h, so any book fits on any shelf. Describe a greedy algorithm that will take the fewest number of shelves to fit all of the books, and show its time complexity.
- **b.** Now consider the general case where the height of the books is not constant, but we have the freedom to adjust the height of each shelf to that of the tallest book. The cost of a particular layout is the sum of the heights of the tallest books on each shelf. Draw a diagram of an example showing that the greedy algorithm will *not* always give the optimal layout.
- **c.** Use dynamic programming to design an efficient solution to the general case of this problem. Make sure to use diagrams as you provide an intuitive explanation of how your algorithm works.

4. *Taking an Exam*: The instructor of an Algorithms class gave his students an exam with n questions on it, worth a total of m points, telling them they needed to answer only 100 points worth of questions. Given that each question i has a difficulty d_i associated with it and is worth p_i points (where p_i is a whole number), give an efficient algorithm to determine the set of questions that will minimize total difficulty and be worth at least 100 points.

5. *Planning a company party*: You are consulting for a corporation that is planning a company party. The company has a hierarchical structure such that the supervisor relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality rating, which is a real number. In order to make the party fun for all attendees, the company president does not want both an employee and his or her immediate supervisor to attend.

You are given the tree that describes the structure of the corporation. Each node of the tree holds the name of an employee, his or her conviviality ranking, and pointers to other nodes representing the other employees that this one supervises. Describe an algorithm to determine a guest list that maximizes the sum of the conviviality ratings of the guests. Analyze the running time of your algorithm. How would you modify this algorithm to make sure that the president of the company is invited to his own party? You must provide an intuitive explanation for how your algorithm works, including a diagram.

6. (extra credit) *Warehouse Organization*: You are interviewing for a job as a warehouse manager. As part of your interview, you are asked to create the tallest stack of boxes you can, while following three simple rules:

1. There are n types of boxes available; you may use as many of each box type as you want.
2. You can put one box on top of another only if the upper box does not jut out past the lower box on any side and is smaller in at least one of the three dimensions.
3. You may rotate a box so that any side functions as its base.

Given that the i^{th} type box has height h_i, width w_i and depth d_i, design an algorithm to maximize total box height. Make sure to provide an intuitive explanation for how your algorithm works.