

CSE 830: Design and Theory of Algorithms

Homework #3

Due: Thursday, Oct. 15th 2009, 10:20am

Limit the solution of each portion of a problem to one sheet of paper. Please do not wait until the last minute to look at the problems.

1. The natural greedy algorithm for making change of n units using the smallest number of coins is as follows. Give the customer one unit of the highest denomination coin of at most n units, say d units. Now repeat to make change of the remaining $n - d$ units.

For each of the following nation's coinage, establish whether or not this greedy algorithm always minimizes the number of coins returned in change. If so, prove it, if not give a counter example.

- 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).
- Portuguese coinage, which includes coins for 1, 2.5, 5, 10, 20, 25 and 50 escudos. You need only consider change for an integer number of escudos.
- United States coinage, which consists of half dollars (50 cents), quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies.
- Martian coinage, where the available denominations are $1, p, p^2, \dots, p^k$, where both p and n are positive integers.

2. **Implement** the dynamic programming algorithm for approximate string matching (in whatever language you wish) and use it to find the best alignment between the following pairs of strings:

- "I have written over his word files" and "My favorite novel is Lord of the Flies"
- "this is what happens when I type slow" and "tishisd whaty havpens when uI type fasht"
- 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.

3. 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. Give an example to show that the greedy algorithm will not always give the optimal layout.
- c. Use dynamic programming to give an efficient solution to the general case of this problem.

4. Suppose you are given an array A of n sorted numbers that has been circularly shifted k positions to the right. For example, $\{35, 42, 5, 15, 27, 29\}$ is a sorted array that has been circularly shifted by $k=2$ positions, which $\{27, 29, 35, 42, 5, 15\}$ has been shifted $k=4$ positions.

a. Suppose you know what k is. Give an $O(1)$ algorithm to find the largest number in A .

b. Suppose you do not know what k is. Give an $O(\log n)$ algorithm to find the largest number in A .

5. 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 only needed to answer 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 always a whole number), give an efficient algorithm to determine the least difficult set of questions for the students to answer that will be worth at least 100 points. Assume that the total difficulty of an exam is the sum of the individual difficulties.

6. (*Extra Credit*) *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 rating, 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 the tree that describes the structure of the corporation, using the left-child, right-sibling representation described in Section 10.4, as figure below. Each node of the tree holds, in addition to the pointers, the name of an employee and that employee's conviviality ranking. Describe an algorithm to make up a guest list that maximizes the sum of the conviviality ratings of the guests. Analyze the running time of your algorithm.