CSc 545: Homework Assignment 2

Assigned: Monday, September 5 2016
Due: 9:30 AM, Monday, Sept 19 2016

Clear, neat and concise solutions are required in order to receive full credit so revise your work carefully before submission, and consider how your work is presented. If you cannot solve a particular problem, state this clearly in your write-up, and write down only what you know to be correct. For complicated proofs, first outline the argument and the delve into the details.

1. (10 pts) Consider the following algorithm for sorting an array $A$ of $n$ numbers:

Algorithm $\text{Three\_Part\_Sort}(A)$
1. $n \leftarrow |A|$
2. if $n \leq 4$
3. then Sort $A$ using insertion sort
4. else
5. $\text{Three\_Part\_Sort}(A[1 \ldots \lceil \frac{2n}{3} \rceil])$ // Sort first two-third of $A$
6. $\text{Three\_Part\_Sort}(A[\lceil \frac{n}{3} \rceil + 1 \ldots n])$ // Sort last two-third of $A$
7. $\text{Three\_Part\_Sort}(A[1 \ldots \lceil \frac{2n}{3} \rceil])$ // Sort first two-third of $A$

(a) Does the algorithm correctly sort any input array $A$? If yes, prove its correctness, if no, give an example of $A$, where the algorithm fails.

(b) Analyze the running time of the algorithm.

2. (10 pts) Prove or disprove:

(a) If we sort the letters in a given alphabet by frequency in non-increasing order, then there exists an optimal prefix code in which the codewords lengths are non-decreasing.

(b) If we are given an alphabet of 256 letters such that the most common letter is less than twice more common than the least common letter, then the fixed-length encoding performs as well as Huffman’s encoding.

3. (10 pts) Consider $n$ stationary javelinas on the wall. We want to tag all of them with tiny GPS trackers (for research into their movement patterns). To do that we can use drones that drop cages than can capture multiple javelinas (each cage is 3-yards long). Design and analyze an efficient algorithm that minimizes the number of cages needed.

4. (15 pts) Graphs:

(a) Prove that an outerplanar graph has at most $2n - 3$ edges.

(b) Prove that the vertices of an outerplanar graph can be placed along a line so that when the edges are drawn as circular arcs (the diameter for the circle connecting $u$ and $v$ is equal to the distance between $u$ and $v$) no two edges cross.

(c) Design and analyze an efficient algorithm that computes the drawing above.

5. (10 pts) You work for a large company in Mountain View, CA. You need to process a collection of text queries so that you can identify whether one of the queries occurs at least half of the time. This, of course, is easy to do if you can store the collection of queries along with a counter for each entry. But the size of the collection is so large that even that company cannot afford the additional storage. You can keep your job if you can solve a slightly easier version of the problem, using only constant amount of memory, namely
two counters (registers). Specifically, design and analyze an efficient algorithm that processes a stream of \( n \) numbers (you see one of them at a time) using only two counters so that if there is a number that occurs at least half of the time you must return it; otherwise return any element.

6. (15 pts) This problem deals with optimal change-making. Given the value of some purchase and a dollar amount that is handed in, the optimal change is the one that returns the smallest number of coins.

   (a) Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies. Prove that your algorithms works correctly.
   (b) Is it true that the greedy algorithm works for any set of coin denominations that contains a penny (so one can make change for every value of \( n \))? If not, provide an example.
   (c) Suppose that the denominations are \( c^0, c^1, \ldots, c^k \), where \( c \) and \( k \) are integers, \( c > 1 \). \( k \geq 1 \). Prove that the greedy algorithm yields an optimal solution.
   (d) Design and analyze an \( O(nk) \) dynamic programming algorithm that computes the optimal change for arbitrary denominations, assuming a pennies are always one of the options (where \( k \) is the number of different coins and \( n \) is the total change amount).

7. (10 pts) Recall the Citibank happiness problem. Design and analyze an \( O(n \log n) \) divide and conquer algorithm and an \( O(n) \) dynamic programming algorithms for this problem.

8. (10 pts) You work for AZ Billboards Inc. and are put in charge of placing billboards on Interstate Highway 10. Instead of solving the specific problem, you decide to solve the generic problem, so that later on, you can use it for other problem instances. Therefore, you are considering a highway that runs for \( M \) miles, with possible billboard locations \( x_1, x_2, \ldots, x_n \), each in the interval \([0, M]\) (specifying their position along the highway). The revenue received for placing a billboard at location \( x_i \) is given by \( r_i > 0 \). State regulations prohibit placing two billboards within 5 miles of less of each other. You would like to find the subset of billboards that maximize the revenue, subject to these constraints. Describe and analyze a dynamic programming algorithm that solves the problem.

9. (10 pts) These questions refer to the LCS problem.
   (a) Describe how to construct the actual LCS from the \( c \) table and the input sequences \( X \) and \( Y \) in \( O(m + n) \) time, without using the \( b \) table.
   (b) Describe a memoized \( O(mn) \) version of LCS-Length.

Extra Credit: Luxembourg is a country surrounded by Germany, France and Belgium. Each of the four pairs of countries shares a border of significant length. Inspired by the idea of building walls between neighboring countries, but also concerned about the costs of such walls, the four countries consider an abstract version of the problem. Straight-line walls are cheaper to build, compared to walls with many corners. Suppose each country had the shape of a triangle (which has the fewest number of corners in any polygon). Prove or disprove that the four countries, each of which is represented by some triangle, can be arranged so that all pairs share non-zero length borders.