CS 5633 Analysis of Algorithms – Spring 12

3/1/12

6. Homework

Due **3/8/12** before class

1. **Range trees**

   (a) Describe a recursive algorithm that constructs a 1D range tree for a **sorted** set of n numbers in $O(n)$ time.

   (b) Same as above, but for a set of unsorted numbers. Your algorithm should run in $O(n \log n)$ time.

   (c) Describe a recursive algorithm that constructs a 2D range tree for a set of n two-dimensional points in $O(n \log n)$ time. *(Hint: Use a bottom-up approach, and the merge-routine from mergesort.)*

2. **Edit Distance**

   Let A and B be two strings of length m and n, respectively. The *edit distance* of A and B is the minimum number of *transformations* needed to transform A into B. Allowed transformations are: *insert* a character into A, *delete* a character from A, *replace* a character in A with another character.

   (a) What is the edit distance for "ABCBDAB" and "BDCABA"? For visualization purposes, align both strings on top of each other such that spaces are inserted into the strings for insertions/deletions, and such that characters on top of each other either mismatch (= replacement operation), match (= no operation), or line up with a space (= insertion or deletion).

   (b) Develop a recurrence for the edit distance. This is similar to the LCS problem.

   (c) Give pseudocode for a dynamic programming algorithm that computes the edit distance for two strings of length m and n. What is its runtime?

3. **Checkerboard**

   Suppose that you are given an $n \times n$ checkerboard. A checker is allowed to move from its current square to the square immediately above, to the square that is one up and one to the left, and to the square that is one up and one to the right (as long as it does not leave the checkerboard). You are also given a cost function c which specifies for every valid move from square x to square y a (possibly negative) amount of $c(x, y)$ dollars.

   Your task is to design an algorithm that finds a path from some position on the bottom edge to some position on the top edge of the checkerboard, such that the total dollar amount that is generated by the moves on the path is maximized.

   Use dynamic programming for this task. First define a recurrence relation for the total cost of a path to a given square, then fill a dynamic programming matrix, and then trace back to find an optimal path. Also give the runtime of your algorithm.

Related questions from previous PhD Exams

Just for your information. You **do not** need to solve them for homework credit.

Consider the problem of finding the longest substring contained in two strings. The substring must consist of consecutive characters in both strings. For example, 101 is the longest substring that appears in both 101010 and 1101.

1. In pseudocode, write a function

   SUBSTRING-LENGTH($string1, position1, string2, position2$)

   that returns the length of the longest common substring (if any) starting at $position1$ in $string1$ and $position2$ in $string2$. What is the worst-case running time of SUBSTRING-LENGTH?

2. In pseudocode, write a function that uses SUBSTRING-LENGTH to return the length of the longest substring in both $string1$ and $string2$. What is the worst-case running time of your function? Give an example that results in the worst-case.

3. Suppose the length of $string1$ is n_1, and the length of $string2$ is n_2. Consider an $n_1 \times n_2$ table T, where the task is to set $T[i, j]$ equal to the value that would be returned by SUBSTRING-LENGTH($string1, i, string2, j$), but to do it more efficiently. How can $T[i, j]$ be determined from the values of neighboring values in the table? Justify your answer.

4. In pseudocode, write an algorithm based on the your answer in the previous exercise. Determine and justify the worst-case running time of your algorithm.