CS 577: Introduction to Algorithms — Homework 2

Out: 02/03/17 — **Due: 02/10/17**

Ground Rules

- Review problems will be discussed at the following week's review.
- Self-graded problems will be discussed at the following week's review, and are to be self-graded by you during the review. Only students present at the review session and turning in their graded solutions in person will earn points.
- Graded problems are to be turned in on the due date at the beginning of the lecture.
- Self-graded problems should be done individually.
- Graded problems should be done in pairs.
- Please follow page limits for all self-graded and graded problems.
- Write your name(s) clearly on your submissions, and turn in each problem on **separate sheets of paper**.

Review Problems

1. A **supersequence** of a sequence is obtained by adding elements to the given sequence (in any locations). For example, *dynamicprogramming* is a supersequence of the sequence *damn*. A **palindrome** is a sequence which reads the same backwards and forwards. For example, *amanaplanacanalpanama* is a palindrome.

   In this problem you are given a sequence A of length n and your goal is to find the shortest supersequence of A that is a palindrome. Develop a recursive algorithm for this problem. Memoize it to obtain a dynamic program. Analyze its running time. (Your DP should run in $O(n^2)$ time.)

2. You are given an array A of n numbers, each of which may be positive, negative, or zero. Your goal is to find a contiguous subarray $A[i \cdots j]$ such that the sum of its elements $A[i], \cdots, A[j]$ is maximized. For example, for the array $[4, -11, 9, 3, -2, 10, -5, 1, 2]$, the subarray $[9, 3, -2, 10]$ achieves the maximum sum, 20.

   Develop a recursive algorithm that finds the maximum sum in a contiguous subarray. Memoize it to obtain a dynamic program. Analyze its running time. (Your DP should run in $O(n)$ time.)

Self-graded Problems

3. **(Page limit: 2 sheets; 4 sides)**

   (a) Suppose you are given two sets of n points, one set $\{p_1, p_2, \cdots, p_n\}$ on the line $y = 0$ and the other set $\{q_1, q_2, \cdots, q_n\}$ on the line $y = 1$. Create a set of n line segments by connect each point p_i to the corresponding point q_i. Describe and analyze a divide and conquer algorithm to determine how many pairs of these line segments intersect, in $O(n \log n)$ time. Your algorithm should take the $2n$ points as input, and return the number of intersections.

   (Hint: What does this problem have in common with the problem of counting inversions in a list?)

   (b) Now suppose you are given two sets $P = \{p_1, p_2, \cdots, p_n\}$ and $Q = \{q_1, q_2, \cdots, q_n\}$ of n points on the unit circle. Connect each point p_i to the corresponding point q_i. Once again our goal is to determine the number of intersections among the resulting line segments.

   Note that the number of intersections do not depend on the actual positions of the points along the circle, but only on their position relative to other points, for example, for each i, which other pairs (p_j, q_j) have exactly one end-point in the arc clockwise from p_i to q_i. Therefore, for simplicity we will assume that the

points are numbered and specified according to their appearance in clockwise order starting from some arbitrary location, call it "origin", on the circle. So, for example, if $p_1 = 5$, then p_1 is the fifth point clockwise from the origin in the set $P \cup Q$.

Given the sets P and Q in this format, describe and analyze a divide and conquer algorithm to determine the number of intersections. Prove the correctness of your algorithm and analyze its running time. Your algorithm should run in $O(n \log^2 n)$ time.

(Hint: Use your solution to part (a).)

Graded Problems

4. **(Page limit: 2 sheets; 3 sides)** An $n \times n$ grid is a graph whose node set is the set of all ordered pairs of natural numbers (i, j) with $1 \leq i \leq n$ and $1 \leq j \leq n$; the nodes (i, j) and (k, ℓ) are joined by an edge if and only if $|i - k| + |j - \ell| = 1$. In other words, it is the adjacency graph of an $n \times n$ chessboard.

   In this problem you are given a grid with each node v labeled by a distinct number x_v. v is called a *local minimum* if x_v is smaller than the labels of each of v's neighbors. Develop a divide and conquer algorithm for finding any local minimum in the grid that runs in $O(n)$ time. (Recall that the number of nodes in the grid is n^2.)

   (a) Describe your algorithm succinctly but precisely.

   (b) Prove its correctness.

   (c) Analyze its running time.

5. **(Page limit: 2 sheets; 3 sides)** You are designing background art for a video game in the form of a cityscape. You are given a list of n buildings. Building B_i is represented as a triplet (L_i, H_i, R_i) where L_i and R_i denote the left and right x coordinates of the building, and H_i denotes the height of the building.

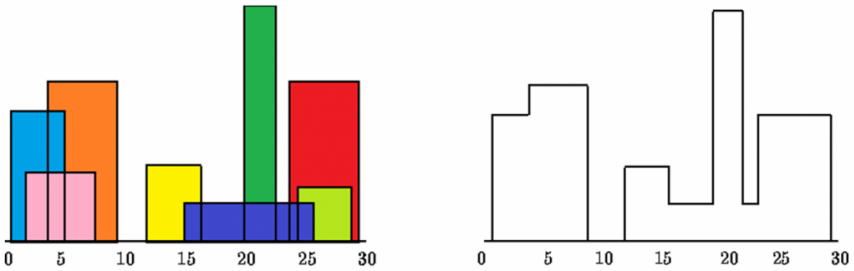
   The cityscape for this problem is an outline of all the buildings put together when seen against a backdrop. It is represented as a list of x coordinates along with the height from that coordinate to the next, arranged in order from left to right. (Note that the list is of length at most $4n$.)

   Example: (See picture below.) The cityscape for the buildings in the list

   $$\{(3, 13, 9), (24, 4, 28), (19, 18, 22), (1, 11, 5), (12, 7, 16), (14, 3, 25), (2, 6, 7), (23, 13, 29)\}$$

   is given by

   $$\{(1, 11), (3, 13), (9, 0), (12, 7), (16, 3), (19, 18), (22, 3), (23, 13), (29, 0)\}.$$



buildings cityscape

   (a) Let the size of a cityscape be the number of elements (tuples) in its list. Describe an algorithm for combining a cityscape A of size n_1 and a cityscape B of size n_2 into one cityscape S of size $O(n_1 + n_2)$. Your algorithm should run in time $O(n_1 + n_2)$. Prove the correctness of your algorithm.

   (b) Describe an $O(n \log n)$ time algorithm for finding the cityscape of n buildings. Prove the correctness of your algorithm and analyze its running time.