CSCI 361, Lab 2

Thursday, February 4

Problem 1

A sorting algorithm is said to be **stable** if it preserves the relative order of any two equal elements in its input. In other words, if an input list contains two equal elements in positions i and j where $i < j$, then in the sorted list they have to be in positions i' and j', respectively, such that $i' < j'$.

a) Is the bogosort algorithm from class stable?

b) **(Levitin 3.1.6)** Is the selection sort algorithm stable?

c) **(Levitin 3.1.10)** Is the bubble sort algorithm stable?

Problem 2

(Levitin 3.2.8) In the string-matching problem, would there be any advantage in comparing pattern and text characters right-to-left instead of left-to-right?

Problem 3

(Based on Levitin 3.2.9) Consider the problem of counting, in a given text, the number of substrings that start with an A and end with a B. (For example, there are four such substrings in $CABAAXBYA$.)

a) Design a brute-force algorithm for this problem, and determine its efficiency class.

b) Find a $\Theta(n)$ algorithm for this problem.

Problem 4

(Levitin 3.3.2) Let $x_1 < x_2 < ... < x_n$ be real numbers representing coordinates of n villages located along a straight road. A post office needs to be built in one of these villages.

a) Design an efficient algorithm to find the post office location minimizing the average distance between the villages and the post office.

b) Design an efficient algorithm to find the post office location minimizing the *maximum* distance from a village to the post office.

Problem 5

(Levitin 3.3.5) The closest-pair problem can be posed on the k-dimensional space in which the Euclidean distance between two points $P' = (x'_1, \cdots, x'_k)$, and $P'' = (x''_1, \cdots, x''_k)$ is defined as:

$$d(P', P'') = \sqrt{\sum_{s=1}^{k} (x'_s - x''_s)^2}$$

What is the efficiency class of the brute-force algorithm for the k-dimensional closest-pair problem?

Problem 6

(Levitin 3.3.3) There is more than one way to define the distance between two points ($P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$). In particular, the *Manhattan distance* is defined as

$$d_M(P_1, P_2) = |x_1 - x_2| + |y_1 - y_2|$$

Does a solution to the closest-pair problem depend on which of the two metrics– d_E (Euclidian) or d_M (Manhattan) – is used? If yes, come up with a counterexample. If no, prove it.