HW 5

Due on **Wednesday, October 9, 2024 at 9am CST**

CS/ECE 374A: Intro. Algorithms & Models of Computation, Fall 2024 Version: **1.0**

Submission instructions as in previous homeworks.

9 (100 PTS.) Report them all.

You are given a set P of n points in the plane, and a parameter r. A pair of (distinct) points $p, q \in P$ is ***r-close*** if the Euclidean distance between p and q is at most r. (You can assume all the points of P have distinct x and y values.)

9.A. Let P_L and P_R be two sets of points, each given to you in sorted order by their y-coordinate, where $|P_L| + |P_R| \leq n$ in total. Assume that all the points of P_L have x-coordinate in the interval $[\alpha - r, \alpha]$, while all the points of P_R have x-coordinate in the interval $(\alpha, \alpha + r]$. Describe how to modify the "elevator" algorithm seen in class, to report all the r-close pairs of points in $P_L \times P_R$. The running time of your algorithm should be $O(n + t)$, where t is the number of pairs of $P_L \times P_R$ that are $\mathbf{4}r$-close. *Prove* that the running time of your algorithm is as desired, and *prove* the correctness of your algorithm (i.e., it outputs all the desired pairs).

9.B. Modifying the algorithm seen in class for computing the closest pair, provide a divide & conquer algorithm that given P and r, outputs all the r-close pairs of P. For credit, the running time of your algorithm should be $O(n \log n + t)$, where t is the number of $4r$-close pairs in P. Prove the running time of your algorithm, and argue why its output is correct.

10 (100 PTS.) Sort in bulk.

You are given n distinct numbers in an unsorted array A.

10.A. (50 PTS.) Describe an algorithm, as fast as possible, that given a parameter k, reports all the numbers of rank ik in A, for $i = 1, \ldots, n/k$. Here, a number in A has ***rank*** t, if exactly $t - 1$ numbers are smaller than it. What is the running time of your algorithm?

10.B. (50 PTS.) You are now given A, and a list of t integer numbers $1 \leq k_1 < k_2 < \cdots < k_t$. Describe an algorithm, as fast as possible, that reports all the numbers in A of rank k_i, for $i = 1, \ldots, t$. What is the running time of your algorithm?