HW 8

Due on **Wednesday, October 25, 2023 at 9pm**

CS 473: Algorithms, Fall 2023 Version: **1.2**

Submission instructions: As in previous homework.

22 (100 PTS.) Computing the smallest square II.

You are given a set P of n points in the plane, and a parameter k. Like in Question 4 in the second homework, we would like to compute the smallest square containing k points of P, denoted by $\square(P, k)$. Let $f(P)$ denote the sidelength of $\square(P, k)$. You are given two subroutines:

- Verify(S, r): In $O(|S| \log |S|)$ time, returns if $f(S) > r$, $f(S) = r$, or $f(S) < r$.
- compMinSq(S): Computes in $O(|S|^4)$ time the square $\square(S, k)$.

By modifying the linear time closest pair algorithm seen in class, describe an algorithm, as fast as possible (in expectation) that computes $\square(P, k)$. Prove the bound on the expected running time of your algorithm. What is the running time of your algorithm?

Note, that you can not assume k is a small or a constant.

23 (100 PTS.) Closest pair in linear time. Again.

for a set of n points P in the plane, and a point $p \in P$, let $\ell(p) = \min_{q \in P-p} \|p-q\|$ be the distance of p to its nearest neighbor in $P - p = P \setminus \{p\}$.

23.A. (50 PTS.) Given a threshold r, describe in detail, using the appropriate grid, how to compute the following two sets of points

$$P_{\leq r} = \{q \in P \mid \ell(q) \leq r\} \qquad \text{and} \qquad P_{>r} = \{q \in P \mid \ell(q) > r\}.$$

Your algorithm should be as fast as possible.

23.B. (50 PTS.) Consider the randomized algorithm that picks a random point $p \in P$, computes $r = \ell(p)$ by scanning the points (i.e., in linear time), computes $P_{>r}$ (using the previous part), updates $P \leftarrow P \setminus P_{>r}$, and repeats the process till $|P| = 2$. Prove that this algorithm computes the closest pair in the original point set P (you can assume all pairwise distances in P are distinct). Namely, the two points remaining in the set when the algorithm terminate are the closest pair. What is the expected running time of this algorithm? Prove your answer! (For simplicity, you can assume all the pairwise distances are unique.)

24 (100 PTS.) Faster randomized algorithm?

You are given a randomized algorithm that solves a problem of size n, in time X (here X is a random variable), such that for any integer t, we have that

$$\mathbb{P}[tn \leq X \leq (t+1)n] \leq \min(\frac{4}{t^2}, 1).$$

(Think about X as an exact count of certain operations, for example, the number of comparisons performed by QuickSelect.)

24.A. (25 PTS.) What is the expected running time of the algorithm? I.e., $\mathbb{E}[X]$?

24.B. (25 PTS.) Show if you can run k copies of the algorithm in parallel, and stop as soon as one of them succeeds, then if Y is the running time of the new parallel algorithm, then

$$\mathbb{P}[Y > (t+1)n] \leq \min\Big(\frac{c}{t^k}, 1\Big),$$

where c is some constant (that can depend on k). Here, in each time step, Y increases by 1, even as each of the k algorithms each performs one step in their execution during time period.

24.C. (50 PTS.) Show how to modify the original algorithm, so the new algorithm solves the same problem, and if Y is the random variable that is its running time, then

$$\mathbb{P}[Y > ctn] \leq e^{-t},$$

where c is some appropriate constant.

(Here, importantly, you are NOT allowed to run parallel copies of the original algorithm – all the execution is done consecutively.)

(BTW, what is the expected running time of the new algorithm?)