CS 598TMC, Fall 2019
Timothy Chan

Homework 3 (due Nov 13 Wednesday (11am in class))

Instructions: You may work in groups of at most 2. Hand in one set of solutions per group. Acknowledge any discussions you have with other students and other sources you have consulted. Solutions must be written *in your own words*.

1. [*40 pts*] Given a set S of n circles in 2D (possibly overlapping and with different radii), a *chain* of length ℓ is a sequence of circles $c_1, \ldots, c_\ell \in S$ such that c_i is contained inside c_{i+1} for every $i \in \{1, \ldots, \ell - 1\}$. We are interested in the problem of finding a chain with the largest length.

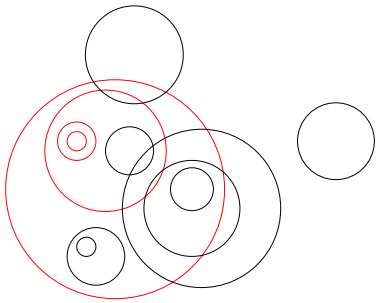
   (a) [*15 pts*] First give a static data structure for a set S of n circles in 2D so that given a query circle q, we can decide whether there exists a circle of S contained inside q. The preprocessing time should be $O(n \log n)$, space $O(n)$, and query time $O(n^\alpha)$ for some constant $\alpha < 1$. Make the exponent α as small as you can.

   [Hint: use linearization; you may use known results from class on halfspace or simplex range searching.]

   (b) [*5 pts*] Next, suppose we are given a set S of n circles in 2D, where each circle is assigned a weight (a real number). Describe how to modify the data structure from part (a) so that given a query circle c, we can find the circle with the largest weight that is contained inside q. The time and space bounds should be the same as part (a).

   (c) [*5 pts*] Describe how to make the data structure in part (b) semidynamic, to support insertions. What are the insertion time and query time?[1]

   (d) [*15 pts*] Finally, describe a subquadratic-time algorithm to solve the problem of finding a longest chain, by using the data structure from part (c).



(the longest chain has 4 circles, shown in red)

2. [*30 pts*] Prof. X claims to have found an $O(n)$-space (static) data structure for storing any set S of n points in 3D so that given any query point, its nearest neighbor in S can be found in $O(\log^{598} n)$ time.

[1] As usual, you may still get full credit for one part of a question even if you are unable to solve earlier parts.

(a) [*15 pts*] Assuming that Prof. X's data structure is correct, describe how to obtain a new $O(n \log n)$-space data structure that can find the 10-th nearest neighbor for any query point in $O(\log^{599} n)$ time.

[Hint: use a tree, where each node stores an instance of Prof. X's structure. First consider finding the 2nd nearest neighbor...]

(b) [*15 pts*] Under the same assumption, but using randomization, describe a better $O(n)$-space data structure that can find the 10-th nearest neighbor for any query point q in $O(\log^{598} n)$ time, with correctness probability at least 0.9. Thus, the query algorithm is allowed to occasionally give incorrect answers (i.e., it is a *Monte Carlo* algorithm). You may assume that the query point q is independent of the random choices made by your data structure.