CS 598TMC, Fall 2021
Timothy Chan

Homework 1 (due Sep 24 Friday 5pm)

Instructions: You may work individually or in groups of 2; submit one set of solutions per group. Always acknowledge discussions you have with other people and any sources you have used (although most homework problems should be doable without using outside sources). In any case, *solutions must be written in your own words.*

1. [*24 pts*] We are given a set P of n points in d dimensions (where d is a constant), where each point has a weight. A *monotone chain* refers to a subsequence of points $p_1, \ldots, p_\ell \in P$ such that for each $j \in \{1, \ldots, d\}$, the j-th coordinates of the points $p_1, \ldots, p_\ell$ are monotonically increasing. (For example, for $d = 2$, we want $p_1, \ldots, p_\ell$ to be increasing in both x and y simultaneously.)

   Describe an efficient algorithm for finding the monotone chain with the largest total weight. The run time should be $O(n \operatorname{polylog} n)$.

   (Hint: use dynamic programming with some version of range tree.)

2. [*32 pts*] We are given a set S of n (possibly overlapping) axis-aligned rectangles in 2D, where each rectangle has a weight. We want to build a data structure so that for any query point q, we can find the rectangle with the largest weight that contains q.

   (a) [*8 pts*] First show that this problem can be reduced to orthogonal range searching and can be solved in $O(n \operatorname{polylog} n)$ space and $O(\operatorname{polylog} n)$ time.
   (Hint: map rectangles in 2D to points in 4D.)

   (b) [*24 pts*] Describe a better solution to the problem by adapting the "recursive grid" approach from class (by Alstrup, Brodal, and Rauhe). Aim for close to $O(n)$ space and close to $O(\log n)$ query time.

3. [*20 pts*] We are given a set S of n disjoint line segments in 2D, where each segment has a weight. We want to build a data structure to answer the following type of queries: given a vertical line segment q, find the segment $s \in S$ with the largest weight that intersects q.

   Describe a solution with $O(n \operatorname{polylog} n)$ space and $O(\log n)$ query time.

   (Hint: one way is via persistence.)

4. [*24 pts*] Recall the trapezoid tree method for planar point location problem, as described in class. Consider a variant of the trapezoid tree method, where instead of dividing a trapezoid into two sub-trapezoids via the vertical line at the median x-coordinate, we divide a trapezoid into b sub-trapezoids via $b-1$ vertical lines, for a fixed parameter b. (And as before, if there are long segments, we divide the trapezoids along these segments.) Analyze the space and query time of the resulting data structure as a function of n and b.

   For full credit, analyze the worst-case number of comparisons made by the query algorithm, and show that it can be made smaller than $1.00001 \log n + O(1)$ for example, by choosing b appropriately. (Thus, this is better than the naive slab method (Method 0 from class), which needs $2 \log n + O(1)$ comparisons besides having larger space.) Determine the best choice of b.