CMPS 3130 Computational Geometry – Spring 20

4/7/20

9. Homework (undergrad)

Due **4/23/20** before class

Please justify all your answers. Often it helps to draw pictures.

1. **Range Counting (10 points)**
   (a) (7 points) Describe how to *augment* a 1D range tree of n elements such that range **counting** queries can be answered in $O(\log n)$ time. (Note that the size k of all the elements in the range is **not** part of the runtime. So, you cannot afford to print all elements in the range.)

   Augmentation means that you store an additional variable in each node of the tree. Usually this variable contains some information about the subtree rooted at this node. For this problem, what would this variable be? *(Ask Carola if you need a hint.)* This augmentation is meant to be a small change. But since it modifies the balanced search tree data structure that we use to implement a 1D range tree, please argue how this additional variable can be updated during the normal operations of a 1D range tree (such as insertion). And argue that your augmentation does not change the asympotic preprocessing time and the asymptotic space complexity.

   (b) (3 points) Describe how to augment a 2D range tree of n elements such that range **counting** queries can be answered in $O(\log^2 n)$ time.

2. **KD-trees (10 points)**
   (a) (5 points) Describe an algorithm to construct a d-dimensional kd-tree for a set P of n points in $\mathbb{R}^d$. Prove that the algorithm takes $O(n \log n)$ time and that the tree can be stored in $O(n)$ space. Assume d is constant.

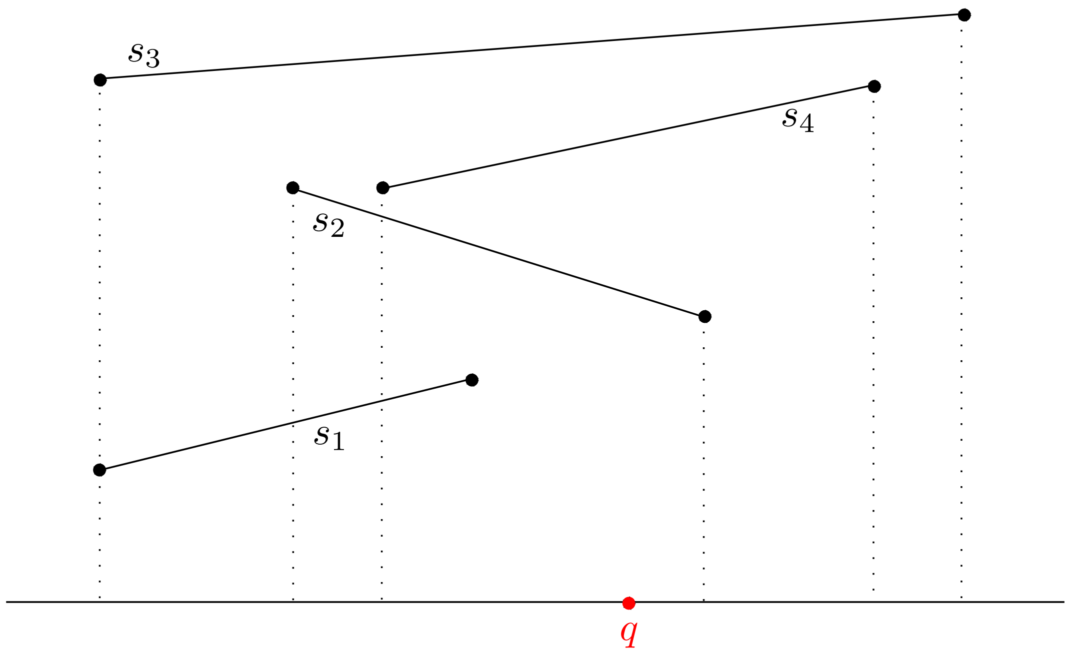
   (b) (3 points) Describe a query algorithm for performing an orthogonal range query in a d-dimensional kd-tree.

   (c) (2 points) For $d = 3$, show that your query algorithm runs in time $O(n^{\frac{2}{3}} + k)$. For this, develop a recurrence for $Q(n)$ and solve it. *(Hint:* $\log_8 4 = \frac{2}{3}$*)*

   (d) (Extra credit) Can you generalize your query time analysis for general (constant) d, to show that the query algorithm runs in time $O(n^{\frac{d-1}{d}} + k)$?

Flip over to next page ⟹

3. **Segment tree (10 points)**



(a) (7 points) Draw the segment tree for the set of line segments $\{s_1, s_2, s_3, s_4\}$ above. Your segment tree should have 15 leaves.

(b) (3 points) Now search for q in the segment tree. Draw the search path into the tree, and report all the segments that contain q.