CS 6301.002 Homework 4

Due Friday April 24th on eLearning

April 2, 2020

Please answer the following **3** questions, some of which have multiple parts. If asked for an $O(T(n))$ time algorithm, you may give a *randomized* algorithm with *expected* time $O(T(n))$.

1. **(From Mount)** You are given three sets of points R, G, and B (red, green, and blue) in $\mathbb{R}^2$. A *tricolor strip* is a pair of parallel lines such that the closed region bounded between these two lines contains exactly three points, one from each of R, G, and B. Define the strip's *height* to be the vertical distance between these lines.

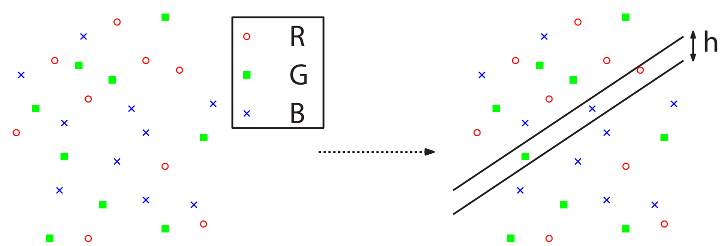

Figure 1. A tricolor strip of height h.

   (a) Explain what a tricolor strip of height h corresponds to in the dual plane.

   (b) If a tricolor strip is of minimum height, what additional conditions must be satisfied? Explain briefly.

   (c) Present an algorithm, which given inputs R, G, and B, computes the minimum height tricolor strip. Your algorithm should run in time $O(n^2 \log n)$, where $n = |R| + |G| + |B|$.

2. **(From Mount)** Let us consider a motion planning problem in the plane where the ground is the x-axis. Consider a robotic crane, whose base is anchored at the origin. The crane can stretch vertically up to any height $h \geq 0$ above the x-axis, and it can extend horizontally at its highest point to the right of the y-axis by any distance $w \geq 0$. There is a hook dangling down at a distance 1 from the tip of the crane. Defining the points $p_0 = (0,0)$, $p_1 = (0,h)$, $p_2 = (w,h)$ and $p_3 = (w,h-1)$, we require that none of the three line segments $\overline{p_0p_1}, \overline{p_1p_2}, \overline{p_2p_3}$ intersects any obstacles in the robot's workspace.

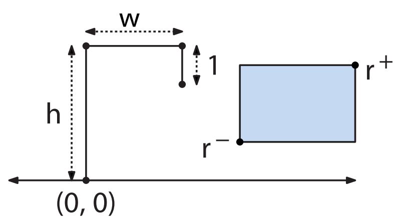

Figure 2. The robotic crane.

Suppose that you are given a workspace consisting of n disjoint axis-parallel rectangular obstacles $\mathscr{R} = \{R_1, \ldots, R_n\}$, where the ith rectangle is defined by its lower corner r_i^- and its upper right corner r_i^+. You may assume that all these rectangles lie above the x-axis, but they may lie on either side of (or overlap) the y-axis.

(a) Given a rectangle $r^- = (x^-, y^-)$ and $r^+ = (x^+, y^+)$, describe the shape of the resulting C-obstacle in the (w,h) configuration space of the crane. *[Hint: There will be a few cases depending on whether the rectangle lies to the left, right, or overlaps the y-axis.]*

(b) Given the set $\mathscr{R}$ of rectangles, and given starting and target configurations $s = (w_s, h_s)$ and $t = (w_t, h_t)$, sketch an algorithm for determining whether there is a collision-free motion of the crane between these configurations. Don't worry about running time. *[Hint: A high-level sketch of the algorithm if sufficient. To make your life simpler, you may assume that you are given a procedure that will input the C-obstacles from part (a), compute the union of these C-obstacles, and return a convinient decomposition of free-space (e.g., as a trapezoidal map).]*

3. **(From 3Marks)**

   (a) Let S be a set of n axis-parallel rectangles in the plane. We want to be able to report all rectangles in S that are completely contained in a query rectangle $Q = [x_{lo}, x_{hi}] \times [y_{lo}, y_{hi}]$. Describe a data structure for this problem that uses $O(n \log^3 n)$ space and has $O(\log^3 n + k)$ query time, where k is the number of reported rectangles. *[Hint: Transform the problem into some orthogonal range searching problem in a higher dimensional space. You may assume orthongal range trees can support both points and ranges that include* $-\infty$ *or* $+\infty$ *in some components (so 1D range* $[-\infty, 5]$ *would include all real numbers less than or equal to* 5 *and point* $(2, +\infty)$ *would lie higher than any bounded rectangular range in the plane).]*

   (b) Let P be a set of n points in the plane. We want to be able to report all points in P that are completely contained in a query *triangle*. However, the triangle is guaranteed to have one horizontal edge, one vertical edge, and one edge of slope -1 or $+1$. Describe a data structure for this problem that uses $O(n \log^3 n)$ space and has $O(\log^3 n + k)$ query time, where k is the number of reported points.