6.851 Advanced Data Structures (Spring'12)

Prof. Erik Demaine

Problem 5 *Sample solution*

Orthogonal line segment intersection. We can solve this problem using distribution sweeping in the same manner as we solved batched orthogonal range searching in lecture. However, an even simpler approach is to reduce this problem directly to batched orthogonal range searching, which takes $O(\frac{N}{B} \log_{M/B} \frac{N}{B})$ time.

Our set of points will be the endpoints of the vertical endpoints, and we will have one query rectangle for each horizontal segment. A horizontal segment with endpoints (x_1, y) and (x_2, y) will correspond to a rectangle with vertices $(x_1, y), (x_2, y), (x_2, \infty)$, and (x_1, ∞). Any vertical segment intersecting a rectangle will have its top end point inside the corresponding rectangle and its bottom end point outside the corresponding segment. Furthermore, if a vertical segment's bottom end point is contained by a rectangle, then its top end point must also be. Thus, for a given rectangle, if a is the number of top end points it contains, and b is the number of bottom end points it contains, $a - b$ vertical segments intersect the corresponding horizontal segment.

Line segment visibility from a point. There are three simple solutions to this problem:

1. The standard non-external memory algorithm for this problem is a sweep line algorithm. First sort the points by their angle to p. Then walk around the points in sorted order, maintaining a priority queue of the line segments currently intersected by the sweep line, sorted by distance from p. The priority queue tells us exactly which line segment is visible at any point. This algorithm easily extends to the cache-oblivious external memory model because we have a cache-oblivious priority queue and sorting algorithm that achieve the desired bounds.

2. We can use a distribution sweeping algorithm with no presorting in which we lazy funnel sort over the angle to p. At each stage, we output the (partial) line segments visible to p from the set of segments in the given range. The key insight is that when merging two such sets, at most two line segments overlap at any point (since there is no overlap within each set) thus we can essentially perform the sweep line algorithm without a priority queue, as the size of the priority queue would always be at most two.

3. We can use a distribution sweeping algorithm analogous to batched orthogonal range searching, in which we sort the points by angle to p, and then lazy funnel sort over the distance to p.

6.851 Advanced Data Structures
Spring 2012