Przemysław Uznański
Chih-Hung Liu

April 4, 2017

Advanced Data Structures

Spring Semester 2017

Exercise Set 7

We start with an exercise that extends the idea signalled at the end of lecture (*fractional cascading.*

Exercise 1:
Consider k *sorted* arrays $L_1, L_2, \ldots, L_k$, each of length n. We want to preprocess them, so that we can do binary search (predecessor or successor queries) in all of them simultanously, with total cost $\mathcal{O}(k + \log n)$.

1. Sorting them leads to naive cost $\mathcal{O}(k \log n)$.
2. Consider new family of lists L' defined recursively as: L'_i is a sorted order of concatenation of L_i and $L'_{i+1}[2], L'_{i+1}[4], \ldots, L'i+1[n-2], L'i+1[n]$ (all of L_i and every second element of L'_{i-1}).
3. Show that all lists $L'_1, \ldots, L'_k$ are still $\mathcal{O}(n)$ length.
4. Show that we can maintain pointers, so that given successor in L'_i, we can find successor in L'_{i-1}.
5. Show that we can maintain pointers. so that given successor in L'_i, we can find "true" successor (element of L_i) in $\mathcal{O}(1)$ time.

Now, we will step by step build a data structure for the 3D orthogonal range reporting ($[a_1, b_2] \times [a_2, b_2] \times [a_3, b_3]$) with $\mathcal{O}(n \log^3 n)$ space and the optimal $\mathcal{O}(\log n + k)$ query time, where k is the number of reported points.

Exercise 2:
In the Exercise Set 5, we already built a data structure for the dominance query ($[-\infty, b_2] \times [-\infty, b_3]$) . Here, we discuss another data structure which is applicable in the 3D orthogonal range reporting. The rough idea is as follows:

1. In addition to the vertical rays, extend a horizontal line from each point until the endpoints hit another ray as shown in the left of Fig. 1. (It could be a line segment, a ray or a line.)
2. For a query point p, find out all the regions which are left to p and intersected by the y-coordinate of p, and report their right vertical rays as shown in the right of Fig. 1.

Please explain how to process this planar subdivision such that the query can be performed in $\mathcal{O}(\log n + k)$ time. The same idea also works for the $[a_2, \infty] \times [-\infty, b_3]$, $[a_2, \infty] \times [a_3, \infty]$, and $[-\infty, b_2] \times [a_3, \infty]$ queries.

Hint:

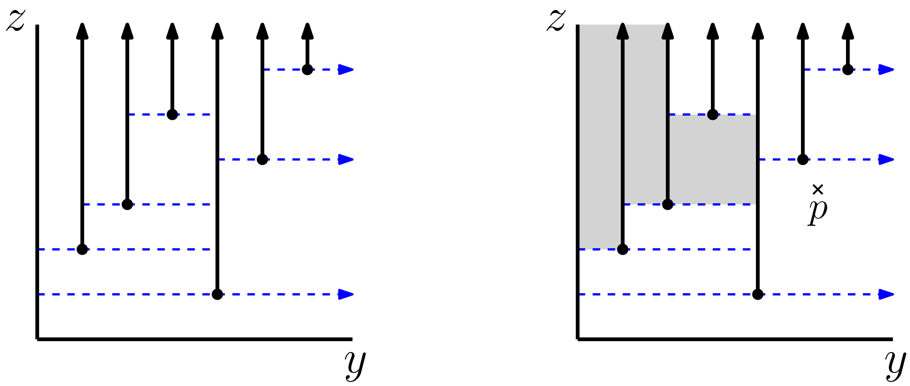

Figure 1: Left: A planar subdivision. Right: the query for p.

- A binary search from one region to another (from left to right) takes $\mathcal{O}(\log n)$ time, leading to $\mathcal{O}(\log^2 n + k)$ query time.
- Fractional cascading (Exercise 1) can improve the query time to $\mathcal{O}(\log n + k)$.

Exercise 3:
Build a data structure for the $[a_1, b_1] \times [-\infty, b_2] \times [-\infty, b_3]$ orthogonal reporting with $\mathcal{O}(n \log n)$ space and the optimal $\mathcal{O}(\log n + k)$ query time. The idea should also work for the $[a_1, b_1] \times [a_2, \infty] \times [-\infty, b_3]$, $[a_1, b_1] \times [a_2, \infty] \times [a_3, \infty]$, and $[a_1, b_1] \times [-\infty, b_2] \times [a_3, \infty]$ queries.
Hint: Build a range tree on the x-coordinates and apply Exercise 2 for the sub-tree of each node.

Exercise 4:
Build a data structure for the $[a_1, b_1] \times [a_2, b_2] \times [-\infty, b_3]$ orthogonal reporting with $\mathcal{O}(n \log^2 n)$ space and the optimal $\mathcal{O}(\log n + k)$ query time. The idea should also work for the $[a_1, b_1] \times [a_2, b_2] \times [a_3, \infty]$ query.
Hint:

- Build a range tree on the y-coordinates.
- For each node v of the range tree, apply Exercise 3:
  - to build a data structure for $[a_1, b_1] \times [-\infty, b_2] \times [-\infty, b_3]$ on the right tree of v,
  - and to build a data structure for $[a_1, b_1] \times [a_2, \infty] \times [-\infty, b_3]$ on the left tree of v.

Exercise 5:
Build a data structure for the $[a_1, b_1] \times [a_2, b_2] \times [a_2, b_3]$ orthogonal reporting with $\mathcal{O}(n \log^3 n)$ space and the optimal $\mathcal{O}(\log n + k)$ query time.
Hint:

- Build a range tree on the z-coordinates.
- For each node v of the range tree, apply Exercise 4
  - to build a data structure for $[a_1, b_1] \times [a_2, b_2] \times [-\infty, b_3]$ on the right tree of v,
  - and to build a data structure for $[a_1, b_1] \times [a_2, b_2] \times [a_3, \infty]$ on the left tree of v.