L18

Range Query

Given n points in $\mathbb{R}^d$ create a search structure to report the points within a given Range

Orthogonal range = rectangle

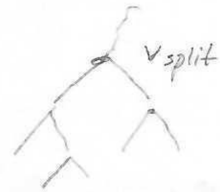
1D

Query time : $O(\log n + k)$ k = output size

Preprocess : $O(n \log n)$

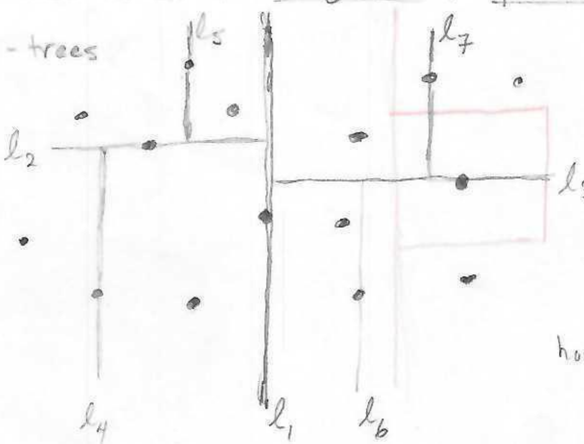
Space : $O(n)$

Update : $O(\log n)$

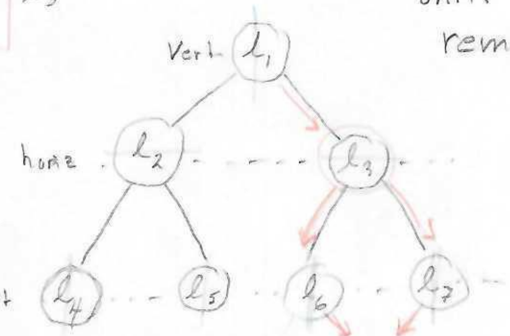


2D kd-trees, range trees, quad trees

kd-trees



Alternately split points in half vertically then horizontally until one point remains



Preprocess $O(n \log n)$ (find median + $2T(n/2)$)

alternate vertical and horiz. at each level of recursion

Space $O(n)$

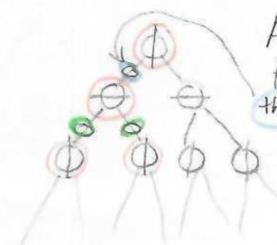
Query $O(\sqrt{n} + k)$ k = output size

k pays for all nodes visited whose region lies in query range

Only search subtree rooted at v if query rectangle intersects v's region (if rectangle contains v's region, report all points in subtree rooted at v.)

$q(n)$ = # regions in kd tree intersected by a vertical line on n points

$$q(n) = \begin{cases} 1 & \text{if } n=1 \\ 2+2q(n/4) & n>1 \end{cases}$$



A vertical line that intersects this region will intersect both regions after horiz. split

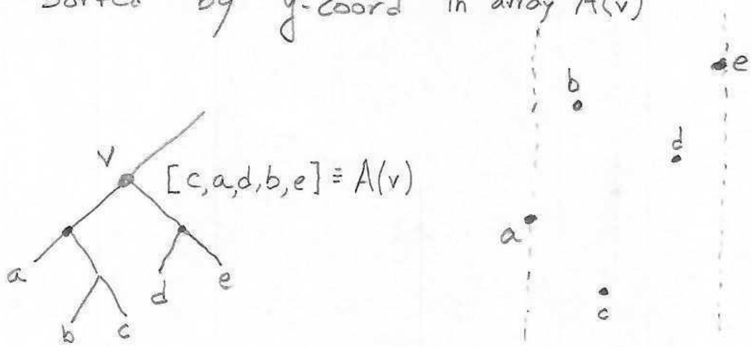
$q(n) = O(\sqrt{n})$

Total query time $q(n)+k$

Range Trees

1. Balanced binary search tree on x-coord (points at leaves sorted by x-coord)
2. At every node v, store points in slab(v) sorted by y-coord in array $A(v)$

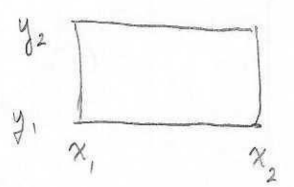
slab(v) ← x-range of points in subtree @ v



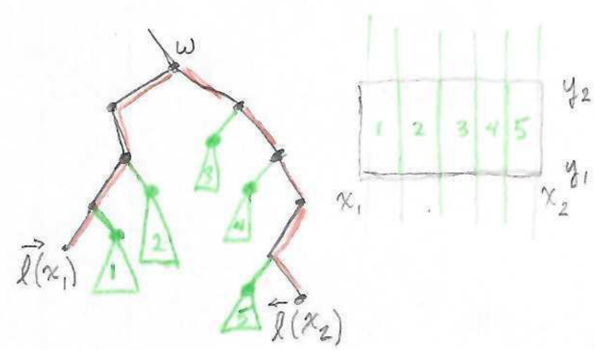
Preprocess $O(n \log n)$

Space $O(n \log n)$ — each point is in $A(v)$ for at most one vertex v per level of tree

Query



1. search tree for x_1 and x_2



2. We want points in leaves between $\ell(x_1)$ and $\ell(x_2)$ that have y-coord in $[y_1, y_2]$
3. Perform binary search in $A(v)$
   - for all right children on path to $\ell(x_1)$ from w
   - " " left " " " " $\ell(x_2)$ from w

   } • all green vertices.

$O(\log n + \text{output})$ per green vertex to report points in rectangle.

$O(\log n)$ green vertices

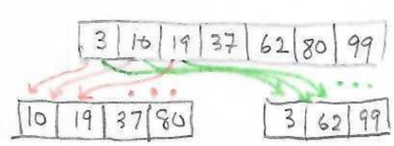
$\Rightarrow O(\log^2 n + k)$ total (since outputs for each vertical slab are disjoint)

k = total output size

Fractional Cascading

$A(w)$ contains $A(w.\text{left})$ and $A(w.\text{right})$

Search in parent gives location of same search in children @ $O(1)$ cost



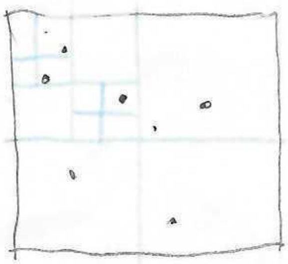
$\Rightarrow O(\log n + k)$ query

Quadtrees

Lemma Quadtree depth $\le \log(s/c) + 3/2$ where c = smallest distance btwn points

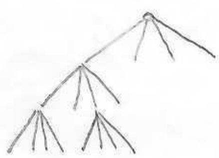
s = size of initial square

proof square side length at depth i is $s/2^i$



max distance between points in such a square is $\sqrt{2}\, s/2^i$

$\Rightarrow \sqrt{2}\, s/2^i \ge c$ for all internal nodes at depth i



$\Rightarrow i \le \log \frac{s\sqrt{2}}{c} = \log(s/c) + 1/2$

Balanced Quadtrees

See slides for RTINs