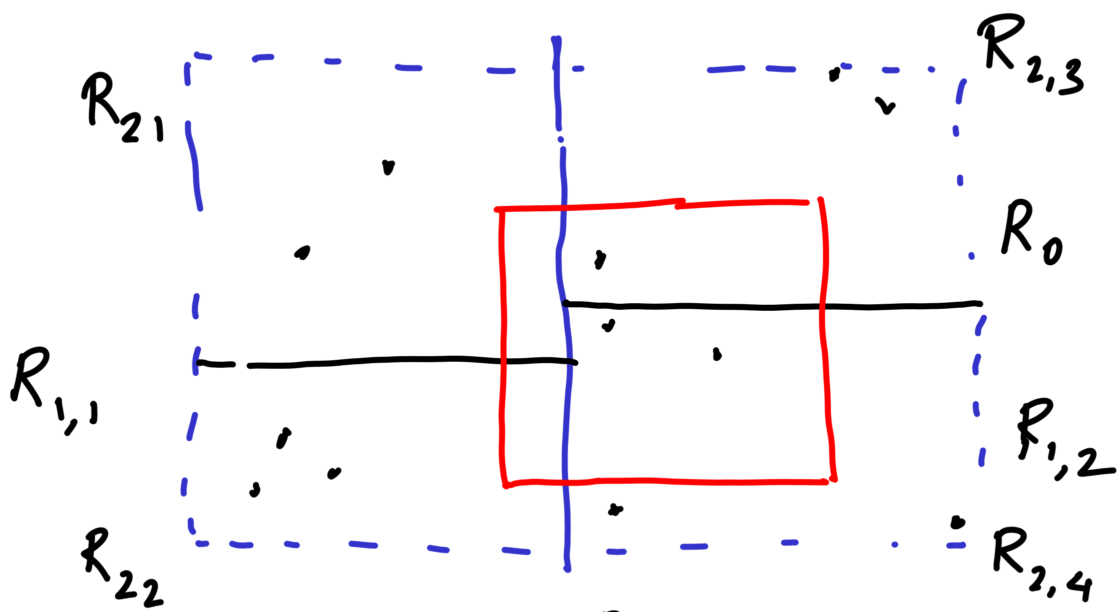
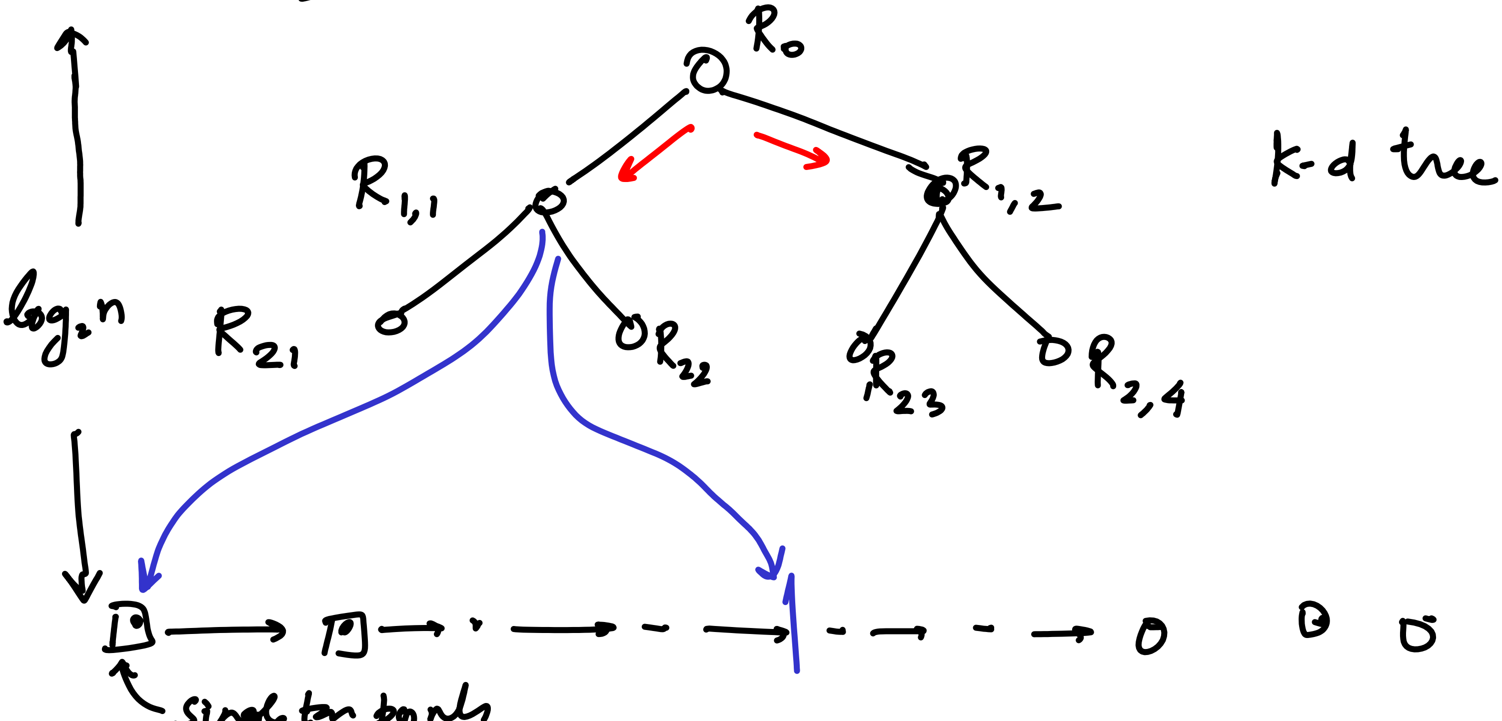
CSL 630 Lecture 25 Nov 5





Each node stores
1. information of the bounding box
2. How the box is split
3. # points (count)
4. pointers to the subarray storing all the points

$O(1)$ information space complexity

$O(n)$ for n points

$O(n \log n)$ preprocessing

Query (r_q , v) node of the tree

query rectangle

Case 1 : r_q does not intersect $R(v)$

Bounding box of v

return null

Case 2 : $R(v) \subset r_q$

Report the count or the points as the case may be

output all points in the sub-tree

Case 3 : r_q "partially overlaps" $R(v)$

Query (r_q, v_1), Query (r_q, v_2)

where v_1 , v_2 are the two children of v.

What is the query complexity?

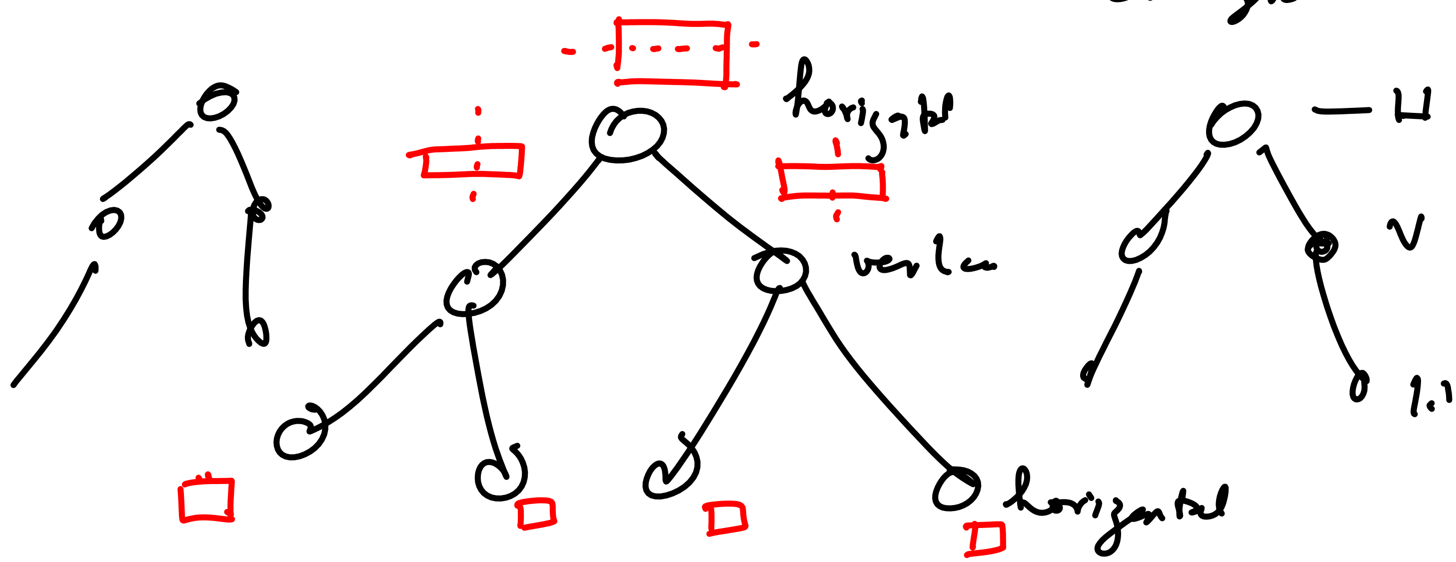
Query complexity is directly proportional to the number of nodes visited in the k-d tree

What is the maxm # nodes visited by any query rectangle?

Write a recurrence for query

$$Q(n) \leq Q\left(\frac{n}{2}\right) + Q\left(\frac{n}{2}\right) + O(1)$$

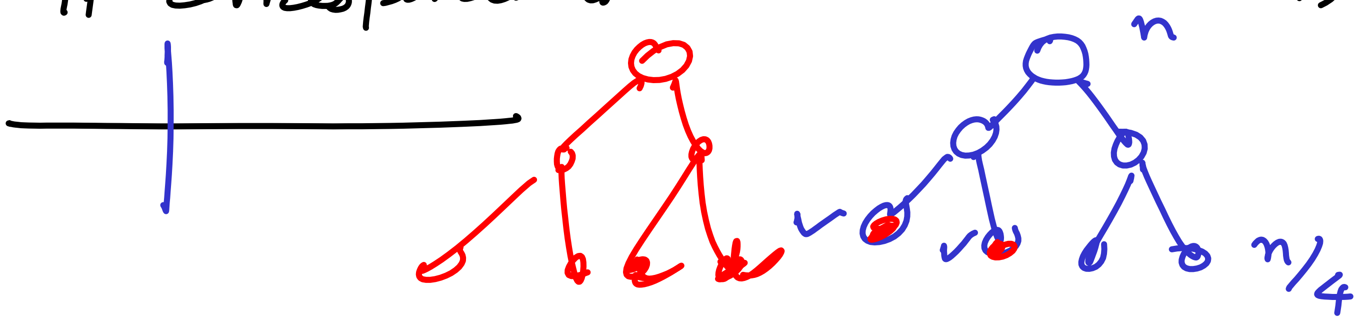
$$\Rightarrow Q(n) \leq O(n)$$ not good enough



→ How many horizontal splits can occur in the worst case

$H(n)$: is the no. of horizontal splits in a problem of size n for any query rectangle

Consider any horizontal edge - how many times can it get split (can be split only at a node where it corresponds to a vertical cut)



$$H(n) = 2 \times H\left(\frac{n}{4}\right) + 1$$

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