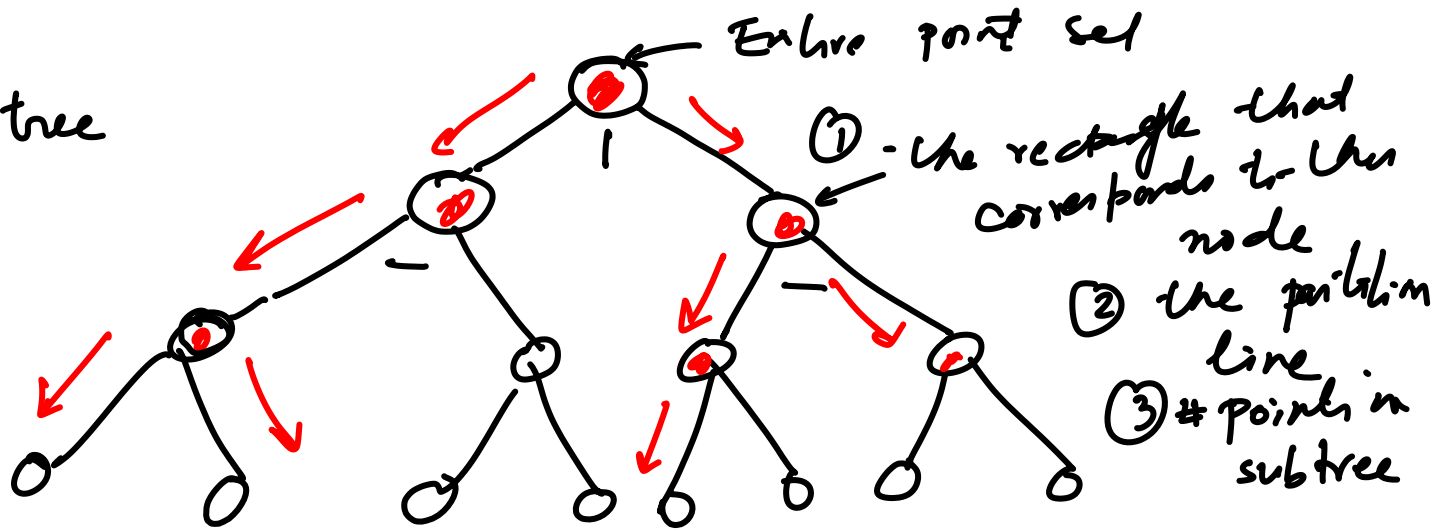


k-d tree



single point

- Cases:
1. Query rectangle R has no overlap (nothing to report)
 2. R contains the entire region of the node (report everything)

3. Which subcases we should continue to search, one or both

The cost of search is proportional to the number of nodes visited

count search

Add up the counts of the nodes completely inside

reporting search

Report all the points in the nodes contained inside

$R(v) \subset R$ $R(v)$ is the rectangle correspond to node v

Space taken by k -d tree: $O(n)$

Time to construct k -d tree: $O(n \log n)$

$Q(n)$: query time for a data structure of size n .

For any vertical/horizontal segment, what is the max # of interactions with all the partitioning lines

Claim A vertical segment can intersect
only with horizontal partitioning
lines

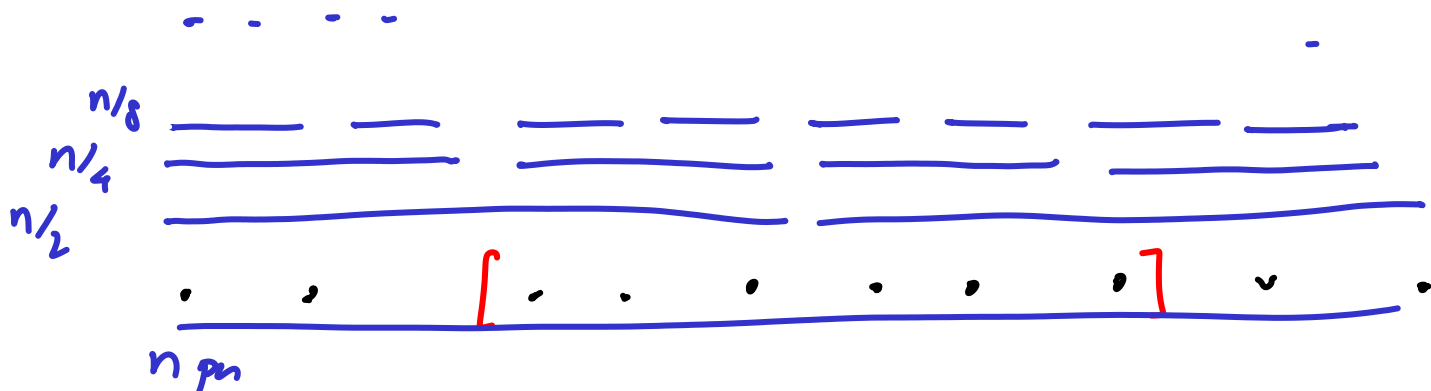
This can double in every alternate
level $\Rightarrow 2^{\lg n/2} = \sqrt{n}$ intersections

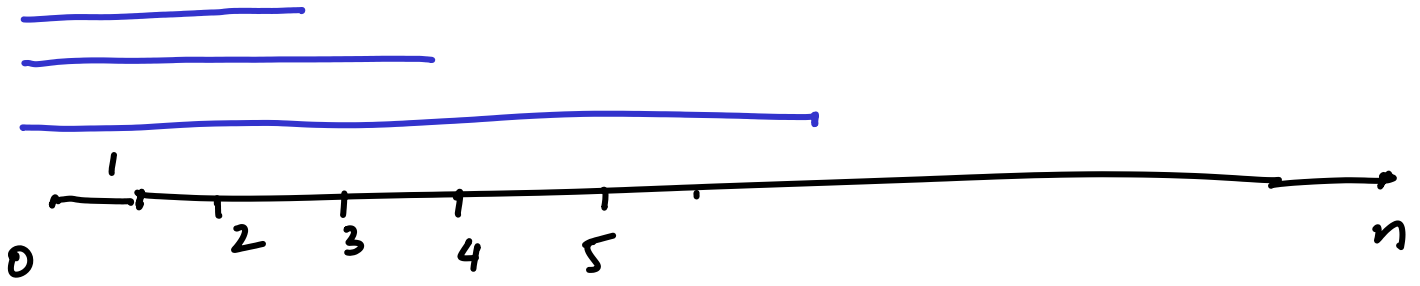
Using kd-trees we can do point search
in $O(\sqrt{n})$ steps and
reporting in $O(\sqrt{n} + m)$ $m = \# \text{ points in } R$

Can we do better?

For eg polylog query-time

Revisit 1-dim search





Given an number i , we want to
write i as a sum of $\frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \dots$

