Each node stores:

1. Information of the bounding box
2. How the box is split
3. # points (count)
4. Pointers to the subboxes

Space complexity: $O(n)$ for updates, $O(m \log n)$ preprocessing

$O(1)$ information
Query \((R_g, v)\) node of the query rectangle

**Case 1.** \(r_g\) does not intersect \(R(v)\)

Return null

**Case 2.** \(R(v) \subseteq r_g\)

Report the count \(n\) of the points in the box, as the case may be.

Output all points in the subtree

**Case 3.** \(r_g\) “partially overlaps” \(R(v)\)

Query \((R_g, v_1)\), Query \((r_g, v_2)\)

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 maximum number of nodes visited by any query rectangle?

Write a recurrence for query:

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

\[ = O(n) \leq O(n^2) \quad \text{not good enough} \]

→ 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).

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

\[
H(n) = \Theta(\sqrt{n})
\]