Range Trees for orthogonal range searching

Given a set of n points in $\mathbb{R}^d$, design a data structure that supports queries of the kind:

For any rectangle in d-dimensions

$R = [x_1^1, x_2^1] \times [x_2^2, x_2^2] \cdots \times [x_d^d, x_d^d]$

- Report points in $R \cap S$
- Count $|R \cap S|$

In general, compute some function $f$ on $R \cap S$
\[ \max_{h_i} \quad \text{RNS} \]

Canonical intervals

\[ n \leq \frac{n}{2} + \frac{n}{4} \quad \ldots \quad < 2n \]

Logn levels

\[ \{x_1, \ldots, x_n\} \]

\[ \{x_{n+1}, \ldots, x_n\} \]

Embed data into the y direction
To answer a 2D m rectangle query, we first find out the (at most 2 log m) subintervals in the x-direction.

For each subinterval, we have built a 1-dimensional range query data structure in the y-direction.

Ans = ∪ queries in disjoint subintervals

Subintervals > 2 log m × Query of each interval

Count query: O(log² m)

Storage: Sum of storage in each level

= log m × n = O(m log n)

Preprocessing:
Initially sort along \( x \): \( O(\log n) \)

We allocate the points to the appropriate canonical subintervals (\( x \)-direction) and then build data structure for each canonical subinterval.

Preprocessing time: \[
\sum_{l=1}^{\log n} 2^l \sum_{i=1}^{\log (\frac{n}{2^{l-1}})}
\]

\( \leq \sum_{l=1}^{\log n} \frac{c \cdot n}{2^{l-1}} \log \left( \frac{n}{2^{l-1}} \right) \)

Sorting in \( y \)-direction

\[
\leq 2^l \cdot \frac{c \cdot n}{2^{l+1}} \cdot \log n \leq c \cdot n \log^2 n
\]
Reducing log \( d \) for \( d \) dimensions.