

Preproc time:

(line 3) by point location + walk
call query alg'm

$$\Rightarrow \boxed{O(n \log n)}$$

Rmk - RIC is very general
(convex hull, Voronoi diagram, ...)

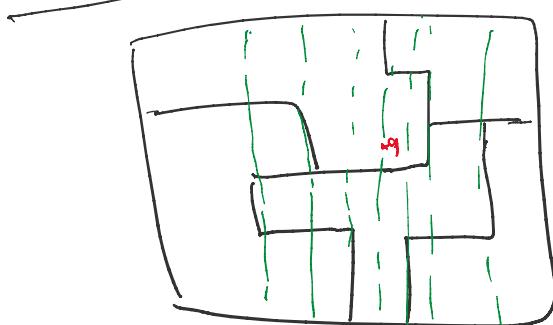
Point Location in Sublog Time?

assume word RAM model $w > \log n$ & $w \geq \log U$
& coords are integers in $[U] = \{0, 1, \dots, U\}$

1D: vEB tree $O(\log \log U)$

2D?

Orthogonal Case



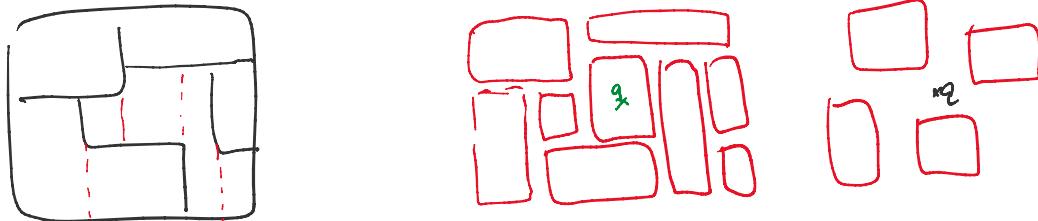
slab method O: $O(\log \log U)$ time by vEB tree
 $O(n^2)$ space

de Berg, Snoeyink, van Krevel '95: $O(n)$ space
 $\underline{O((\log \log U)(\log \log n))}$ time

e.g. use persistence
 or use separator / Sampling

C'11: $O(n)$ space
 $O(\log \log U)$ time

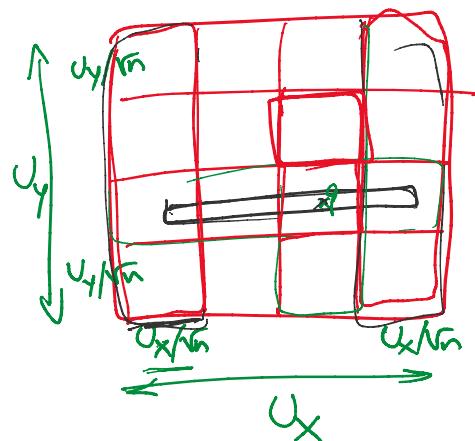
Approach 1:



Given n disjoint rects,

idea - divide & conquer with $\sqrt{n} \times \sqrt{n}$ uniform grid

(let universe be $[U_x] \times [U_y]$)



DS:

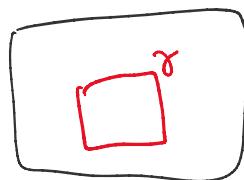
1. for each column/row σ ,
 recursively build DS for all rects that
 have a vertex in σ .

each
rect is
stored in
structs.
 \hookrightarrow recurs.

2. for each grid cell γ ,

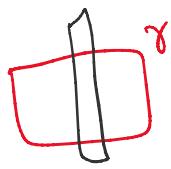
record info.

\hookrightarrow \exists rect. completely containing γ ?)

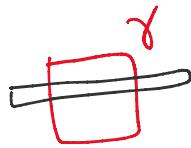


record info.

$O(n)$
space



- (is \exists rect. completely containing γ ?)
- (is \exists rect cutting across γ horizontally? -)
vertically?



Query alg'm, given pt q :

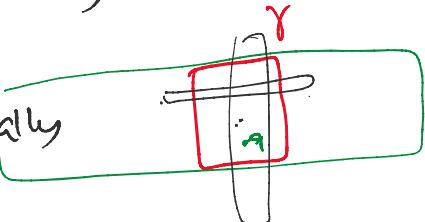
find grid cell γ containing q $\xleftarrow{O(1) \text{ time by int. division}}$

if \exists rect completely containing γ

done

if \exists rect cutting across γ horizontally
recurse in γ 's row

if - - - - - vertically
- - - - column



else recurse in γ 's row or γ 's column
(your choice)

only 1 recursive call

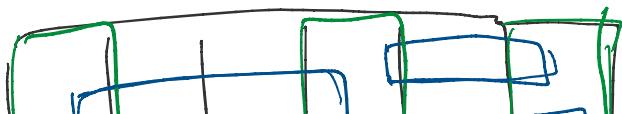
$$\Rightarrow Q(n, U_x, U_y) \leq \begin{cases} Q\left(\frac{n}{m}, \frac{U_x}{m}, U_y\right) + O(1) & \text{or} \\ Q\left(n, U_x, \frac{U_y}{m}\right) + O(1) \end{cases}$$

good initially, but not good as n gets smaller...

Approach 2 :

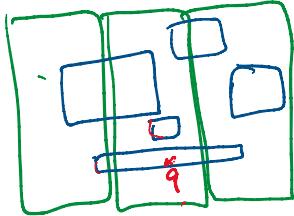
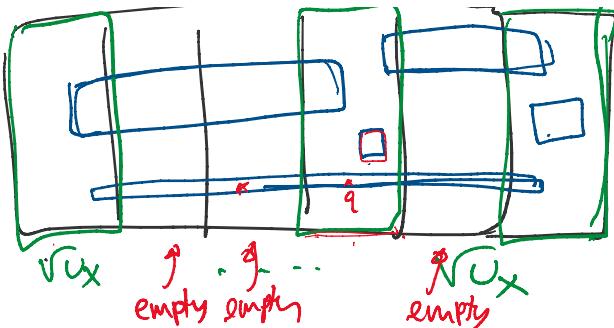
idea - divide & conquer based on vEB

divide into $\sqrt{U_x}$ columns of width $\sqrt{U_x}$



DS B

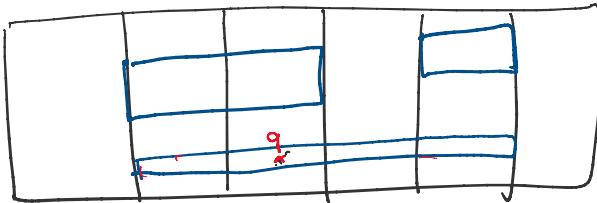




1. let $D = \text{set of } \underline{\text{nonempty}} \text{ columns}$
 has ≥ 1 vertex
 washing

2. recursively build DS A after rounding

for the case when q 's column is not in D . \rightarrow DS A



after rescaling,
universe size $\sqrt{U_x} \times U_y$.

3. recursively build DS B for
all nonempty columns combined?

for the case
when q 's column
is in D

universe size $\leq \underline{n\sqrt{U_x} \times U_y}$

each rect stored in struct.
 \Leftarrow recursive.

Query alg'm, given q :

find column σ containing q $\leftarrow O(1)$ time

if $\sigma \notin D$ recurse in DS A

else recurse in DS B
 $\xrightarrow{O(1) \text{ time by hashing}}$

$$\Rightarrow Q(n, U_x, U_y) \leq Q(n, \underline{\sqrt{U_x} U_y}) + O(1)$$

Approach 3: same but in y

$$Q(n, U_x, U_y) \leq Q(n, U_x, \underline{n\sqrt{U_y}}) + O(1).$$

Find Approach combine!

Find Approach Combine!

Case 1. $n \geq \underline{U_x}^{1/3}$ & $n \geq \underline{U_y}^{1/3}$.

Approach 1 $\Rightarrow Q(n, U_x, U_y) \leq \begin{cases} Q(n, U_x^{5/6}, U_y) + O(1) \\ Q(n, U_x, U_y^{5/6}) + O(1) \end{cases}$

$(U_x \rightarrow \frac{U_x}{\sqrt{n}} \text{ or } U_y \rightarrow \frac{U_y}{\sqrt{n}})$

Case 2. $n < U_x^{1/3}$

Approach 2 $\Rightarrow Q(n, U_x, U_y) \leq Q(n, U_x^{5/6}, U_y) + O(1)$

$(U_x \rightarrow n\sqrt{U_x})$

Case 3. $n < U_y^{1/3}$.

Approach 3 $\Rightarrow Q(n, U_x, U_y) \leq Q(n, U_x, U_y^{5/6}) + O(1).$

levels of recursion = $O(\log(\log U_x) + \log(\log U_y))$

\Rightarrow query time $\boxed{O(\log(\log U))}$

Space $O(n \cdot 4^{\underline{O}(\log(\log U))})$

$= O(n (\log U)^{O(1)}).$

can be reduced to $\boxed{O(n)}$
by separator or sampling method