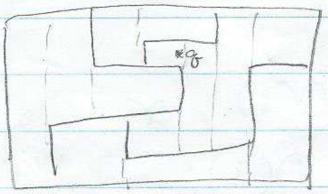
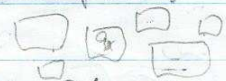
Point Location in Sublog Time: Orthogonal Case



(Assumption: word RAM model, $w \geq \log U$
integer input $w \geq \log n$)

eg.w:



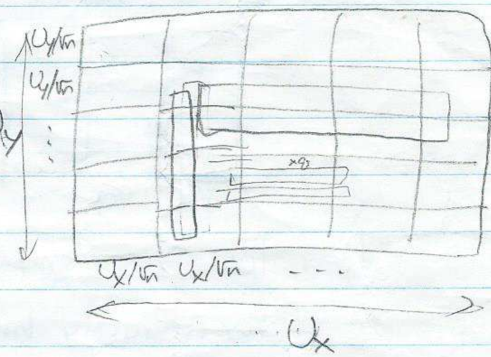
Slab Method: $O(\log\log U)$ time (by vEB tree)
$O(n^2)$ space

de Berg-Snoeyink-van Kreveld '95: $O(n)$ space
$O((\log\log U)^2)$ time
(eg. use separators or persistence)

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

Approach 1

idea - use $\sqrt{n} \times \sqrt{n}$ uniform grid
let universe be $[U_x] \times [U_y]$
form $\sqrt{n}$ columns of width $\frac{U_x}{\sqrt{n}}$
& $\sqrt{n}$ rows of height $\frac{U_y}{\sqrt{n}}$



Given n disjoint rects:
1. for each column/row σ,
   recursively build DS for all rects. that have a vertex in σ
2. for each grid cell γ,
   record info (is γ completely inside a rect.?
   cut across by a rect.?)
   horizontally/vertically

(Note: each rect. is stored ≤ 4 substructs)

Query alg'm, given pt q:
find grid cell γ containing q ← $O(1)$ time
if γ is completely inside a rectangle
  done
if γ is cut across horizontally then recurse in γ's row
" " " vertically " " " γ's col
else recurse in γ's row or column

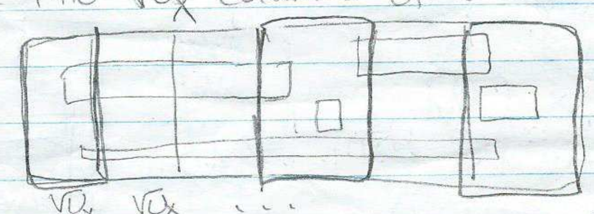
(Note: just 1 recursive call!)

$$Q(n, U_x, U_y) \leq \begin{cases} Q(n_i, \frac{U_x}{\sqrt{n}}, U_y) + O(1) & \text{or} \\ Q(n_j, U_x, \frac{U_y}{\sqrt{n}}) + O(1) \end{cases} \qquad (*)$$

[good initially but not good as n gets small]
[could reduce to rank space to make $H, W = O(n)$, but extra loglog]

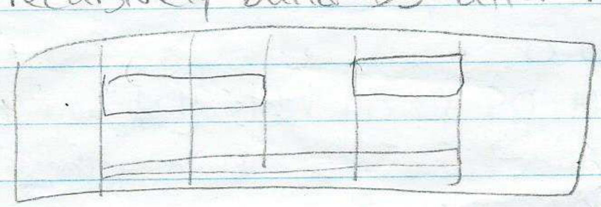
Approach 2

idea - generalize vEB
divide into $\sqrt{U_x}$ columns of width $\sqrt{U_x}$



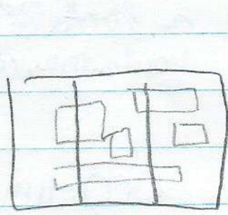
i.e. contain a vertex

1. let D = set of all nonempty columns
2. recursively build DS after rounding



universe size $\sqrt{U_x} \times U_y$
"top structure"

3. recursively build DS for all nonempty columns, glued into one!



universe size $\leq n\sqrt{U_x} \times U_y$
"bottom structure"

(Note: each rect. is stored in ≤ 2 structs)

Query alg'm, given q:

find column σ containing q ← $O(1)$ time

if $\sigma \notin D$ then recurse in top structure

else recurse in bottom structure

$$\Rightarrow Q(n, U_x, U_y) \le Q(n, n\sqrt{U_x}, U_y) + O(1). \quad (**)$$

[U_x converges to $O(n)$, not const!]

Approach 3

same but in y

$$Q(n, U_x, U_y) \le Q(n, U_x, n\sqrt{U_y}) + O(1). \quad (***)$$

Final Approach

combine!

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

$$(*) \Rightarrow Q(n, U_x, U_y) \le \begin{cases} Q(n, U_x^{5/6}, U_y) + O(1) \text{ or} \\ Q(n, U_x, U_y^{5/6}) + O(1) \end{cases}$$

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

$$(**) \Rightarrow Q(n, U_x, U_y) \le Q(n, U_x^{5/6}, U_y) + O(1)$$

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

$$(***) \Rightarrow Q(n, U_x, U_y) \le Q(n, U_x, U_y^{5/6}) + O(1)$$

levels of recursion:

$O(\log\log U_x + \log\log U_y)$

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

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

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

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

or sampling (region size b/w $(\log U)^{O(1)}$)