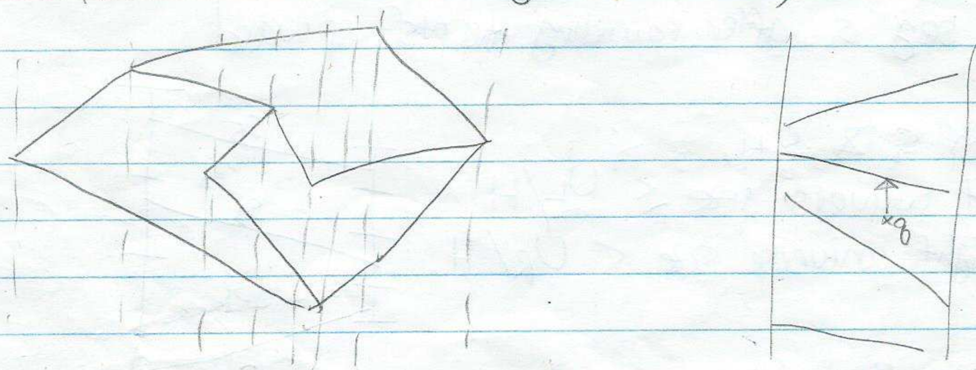
Point Location in Sublog Time: General Case

(Assumption: coords are integers in $[U]$, $w \geq \log U$, $w \geq \log n$)



C.-Patrascu '06:

$O(n)$ space, $O\left(\frac{\log n}{\log\log n}\right)$ time or $O\left(\sqrt{\frac{\log U}{\log\log U}}\right)$ time

suffice to solve slab subproblem:

apply ~~slab method~~, $\Longrightarrow$ $O(n^2)$ space for general problem

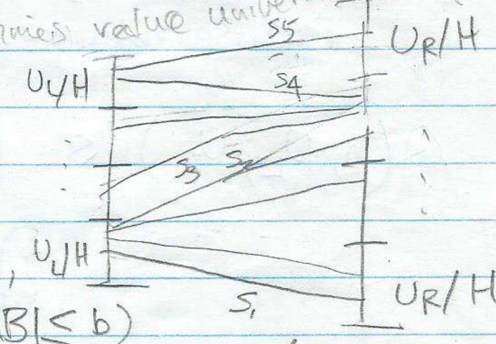
then apply separator method to reduce space...

Method 1

[vEB trees don't work, can't hash...]

idea - b-way divide & conquer (like fusion tree!)

- but can't pack b segs in a word...
- sometimes reduce n, sometimes reduce universe size



let U_L = "left universe size"

U_R = "right universe size"

sort segs S

let $B = \{$segment n/b, $2n/b$, $3n/b$, ..., $(b-1)n/b$ in $S\}$ ($|B| \leq b$)

divide left universe into subintervals of length U_L/H

right " " " " U_R/H

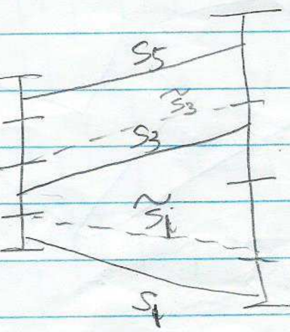
s_1 = lowest seg

for $i = 1, 3, 5, \ldots$

s_{i+1} = highest seg in B that has left endpt in same subinter as

or right endpt " " "

s_{i+2} = next seg in B after s_{i+1}
$\tilde{s}_i$ = seg s_i after rounding endpts upward



Note 1. between $\tilde{s}_i$ & s_{i+1},
left universe size $\le U_L/H$
or right universe size $\le U_R/H$

2. between s_{i+1} & s_{i+2}
\# segs $\le n/b$

3. $s_1 \prec \tilde{s}_1 \prec s_3 \prec \tilde{s}_3 \prec \ldots$

4. can pack $\tilde{s}_1, \tilde{s}_3, \ldots$ in a word
assuming $\boxed{b \log H \le w}$

Query alg'm, given pt q:
locate q in $\{\tilde{s}_1, \tilde{s}_3, \tilde{s}_5, \ldots\}$ by word op in $O(1)$ time
with 1 extra comp, can locate q in $\{s_1, s_3, s_5, \ldots\}$
with 1 more comp, can locate q in $\{s_1, s_2, s_3, \ldots\}$
say $s_i \prec q \prec s_{i+1}$
recurse between s_i & s_{i+1}

$$Q(n, U_L, U_R) \le \begin{cases} Q(n, \frac{U_L}{H}, U_R) + O(1) & \text{or} \\ Q(n, U_L, \frac{U_R}{H}) + O(1) & \text{or} \\ Q(\frac{n}{b}, U_L, U_R) + O(1) \end{cases}$$

$$\Rightarrow O(\log_H U_L + \log_H U_R + \log_b n)$$
$$= O\left(\frac{\log U}{\log H} + \frac{\log n}{\log b}\right)$$
$$= O\left(\frac{w}{\log H} + \frac{\log n}{\log b}\right)$$

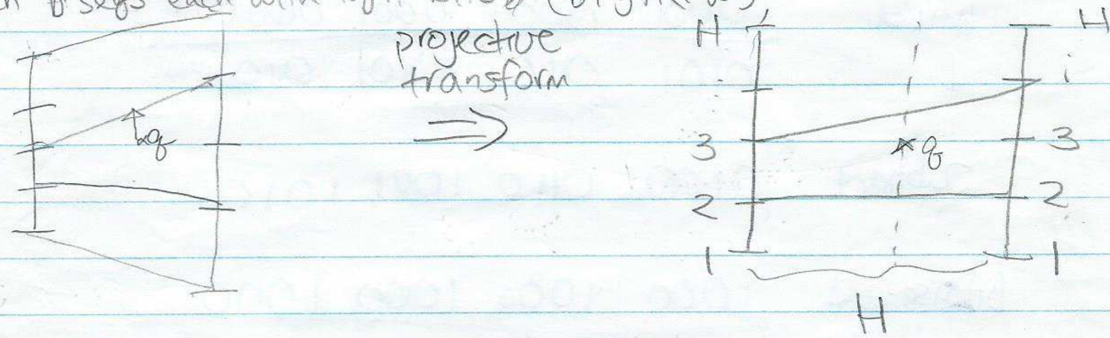
Set $b = \frac{w}{\log H}$

$$= O\left(b + \frac{\log n}{\log b} \right) \qquad \text{set } b = \log^{\varepsilon} n$$

$$= \boxed{O\left(\frac{\log n}{\log\log n} \right)}$$

space $\boxed{O(n)}$

<u>Note</u> word op can be replaced by standard ops
given b segs each with $\log H$ bits & ($b \log H \leq w$)



can round q to nearest integer
$q = (x, y)$

say segment i is from $(0, a_i)$ to (H, b_i)
want to compare $y \overset{?}{\leq} \frac{b_i - a_i}{H} x + a_i$

i.e. $Hy \overset{?}{\leq} (b_i - a_i)x + Ha_i$

| $b_1 - a_1$ | $b_2 - a_2$ | ... | |

mult | x |

| $(b_1 - a_1)x$ | $(b_2 - a_2)x$ | ... |

add | Ha_1 | Ha_2 | ... |

compare against Hy

Subproblem: compare a given value against list of b values (each $\log H$ bits)

e.g. compare 0101
with list 0001 0011 0110 0111

add 1000 1000 1000 1000

1001 1011 1110 1111

0101

mult 0001 0001 0001 0001

0101 0101 0101 0101

subtract 0100 0110 1001 1010

bitwise-and 1000 1000 1000 1000

0000 0000 1000 1000

MSB
(most signif. 1-bit)

(can avoid by multipli (convol ⇒ gives count!)

Method 2

set $b = n$ this time, don't pack

solve subproblem for $\tilde{S}_1, \tilde{S}_3, \tilde{S}_5, \ldots$ by table look-up

(table size $O(H^2)$)

$\Rightarrow$ space $O(H^2 n)$

query $O\left(\frac{\log U}{\log H} + \cancel{\frac{\log n}{\log b}}\right)$

to reduce space, select every other H^2 segs

$\Rightarrow$ space $O(n)$

query $O\left(\frac{\log U}{\log H} + \log H^2\right)$

binary search

$= O\left(\sqrt{\log U}\right)$ by setting $\log H = \sqrt{\log U}$

better still $O\left(\frac{\log U}{\log H} + \frac{\log H}{\log\log H}\right)$

by method 1

$= \boxed{O\left(\sqrt{\frac{\log U}{\log\log U}}\right)}$ by setting $\log H = \log U \log\log U$

[Better? open]

[offline queries. time $2^{O(\sqrt{\log\log n})}$, better than $\log^{\varepsilon} n$]

Pr