Next: Fusion Tree (Fredman, Willard '90)

static, $O(n)$ space,

query time $\boxed{O\left(\frac{\log n}{\log w}\right)} \le O\left(\frac{\log n}{\log\log n}\right)$

beats $\log n$, regardless of U!!

basic idea - deg-b search tree, with $b \approx \sqrt{w}$

how to search among b numbers in $O(1)$ time?

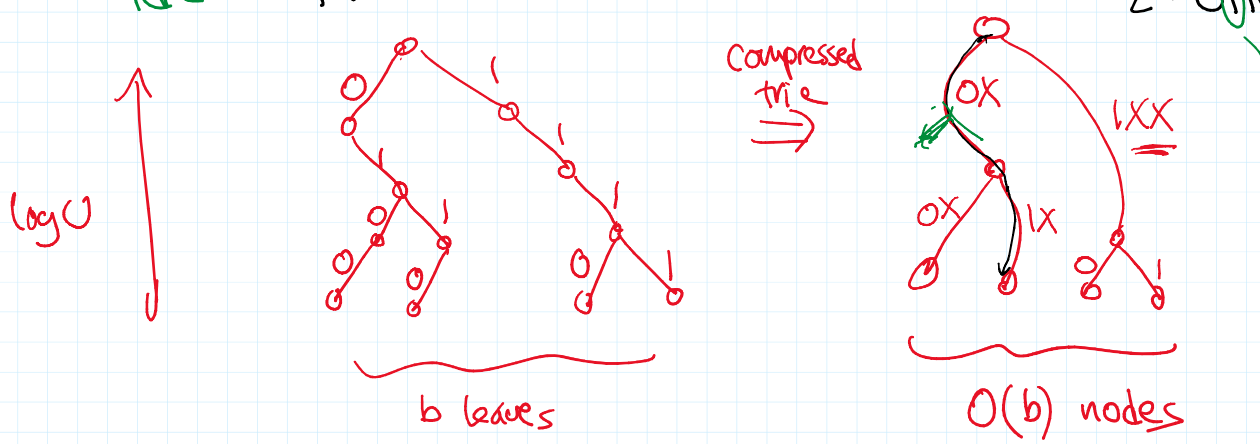
Compress b numbers in one word

e.g. $\{8, 10, 14, 15\} = \{\overbrace{0100}^{\log U}, 0110, 1110, 1111\}$

$q = 0011$

$z = 0110$

idea - trie



$\Rightarrow$ can encode compressed trie in a word

$b \log\log U \le b \log w$ bits $\le \sqrt{w} \log w = o(w)$

How to query q:

follow path, & get index of element z

may be wrong

by nonstandard word op in $O(1)$ time

find first bit where q & z differ

in $O(1)$ time

most significant 1-bit of $q \oplus z$

get index of ans

by another nonstandard word op in $O(1)$ time

can simulate everything with standard word ops (bitwise and/or, add, multiplication)

but very messy!

Combine vEB & Fusion Tree!

$w \geq \log U$

$O(n)$ space

query time $O\left(\min\left\{\log\log U, \frac{\log n}{\log w}\right\}\right)$

$\leq O\left(\min\left\{\log w, \frac{\log n}{\log w}\right\}\right)$

$\leq \boxed{O\left(\sqrt{\log n}\right)}$

Rmk: can use Beame-Fich to improve query time

$O\left(\min\left\{\frac{\log\log U}{\log\log\log U}, \frac{\log n}{\log w}\right\}\right)$

$= O\left(\min\left\{\frac{\log w}{\log\log w}, \frac{\log n}{\log w}\right\}\right)$

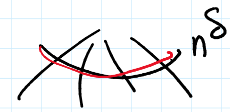
$\leq$

$$= O\left(\min\left\{ \frac{\quad}{\log\log w}, \frac{\quad}{\log w} \right\}\right)$$

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

Space is $O(n^{1+\varepsilon})$
but can be reduced

trick - use n^δ-way D & C.



$$S(n) = n^\delta S(n^{1-\delta}) + O\big((n^\delta)^{\cancel{1+\varepsilon}}\big) \qquad \varepsilon = 1$$

$$O(n^{2\delta})$$

Andersson-Thorup
"exponential search tree"

Can be made dynamic

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

$$Q(n) = Q(n^{1-\delta}) + O\left(\sqrt{\frac{\log n}{\log\log n}}\right)$$

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

Rmk - this is optimal in terms of n because of matching lower bd in "cell-probe model"

Rmk - for priority queues: can do better

Han-Thorup '02: $O(\sqrt{\log\log n})$ expected time

$\Rightarrow$ can sort n integers in $O(n\sqrt{\log\log n})$ expected time

GEOMETRY

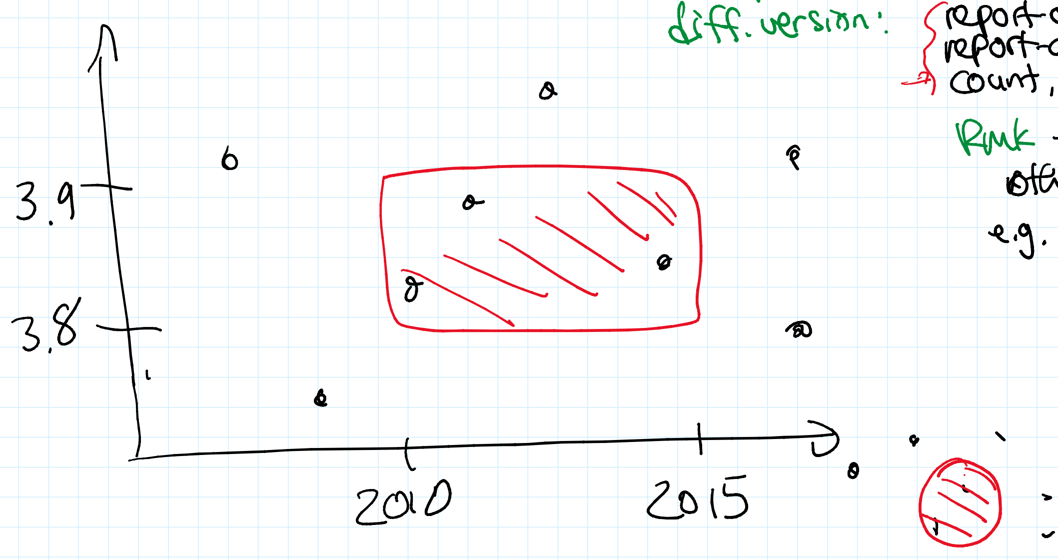
Problem (Orthogonal Range Searching)

Given n pts in 2D, build data structure to answer queries:

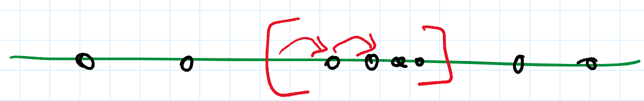
given axis-aligned rectangle q, find pts inside q,

diff. version: report-one, report-all, count, sum of weights, min weight ← range min queries ...

Rmk - can consider other ranges e.g. disks, triangles, ...



1D:



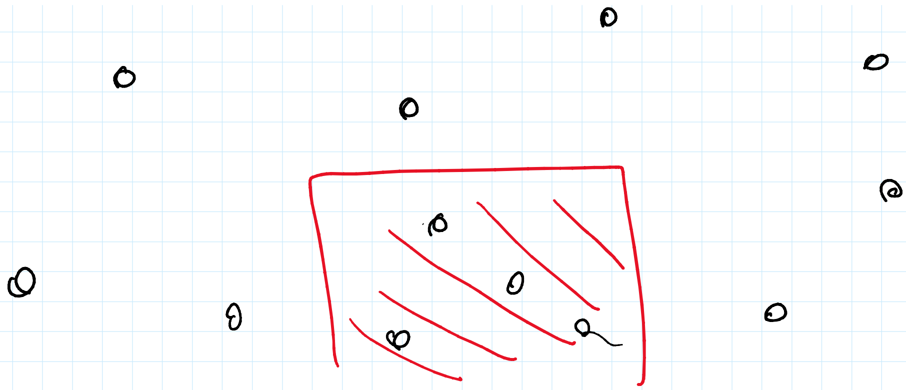
space $O(n)$

query $O(\log n)$ for report-one, count

$O(\log n + k)$ for report-all

insert/delete $O(\log n)$ by BST

2D? special case: 3-sided rectangle



reduces to 1D range min query

$\Rightarrow$ $O(n)$ space

query time $O(\log n + 1)$ $= \boxed{O(\log n)}$

for report-one

2 pred search in x

for report-all: $O(\log n + k)$

by repeatedly find min & recurse left & right