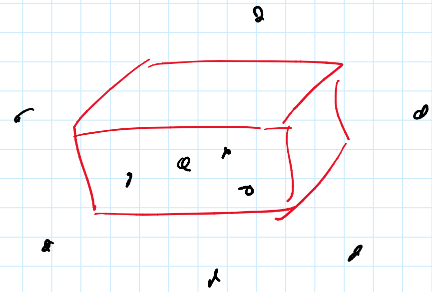
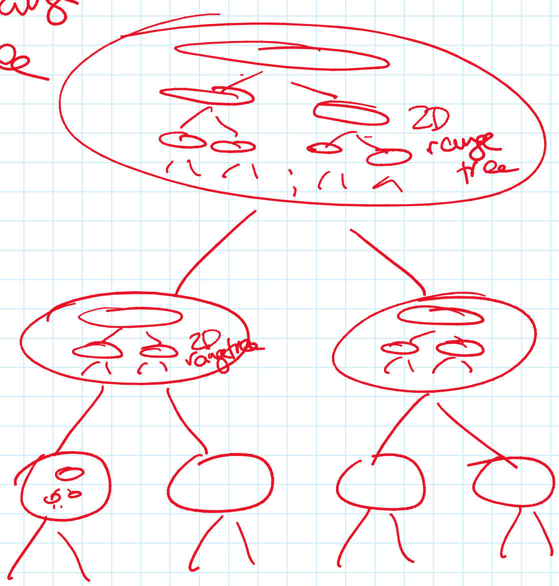
insert/delete $\boxed{O(\log^2 n)}$ amortized

query $\boxed{O(\log^2 n)}$

Extension to higher-D:



3D range tree



Space $S_d(n) = 2S_d\left(\frac{n}{2}\right) + \underline{S_{d-1}(n)}$

$\Rightarrow \quad S_d(n) = O(S_{d-1}(n) \log n)$

$\Rightarrow \quad \boxed{O(n \log^{d-1} n)}$

query time $Q_d(n) = O(Q_{d-1}(n) \log n)$

$\Rightarrow \quad \boxed{O(\log^{d-1} n)}$ $(+k$ for report)

(can be dynamic ...)

Trade-offs: deg-b range tree e.g. $b = n^{\varepsilon/d}$

Trade-offs: deg-b range tree

$$S_d(n) = O\left(n(\log_b n)^{d-1}\right) \qquad \text{e.g. } b = n^{\varepsilon/d} \Rightarrow O(n)$$

$$Q_d(n) = O\left((b \log_b n)^{d-1} \log n\right) \Rightarrow O(n^{\varepsilon})$$

or

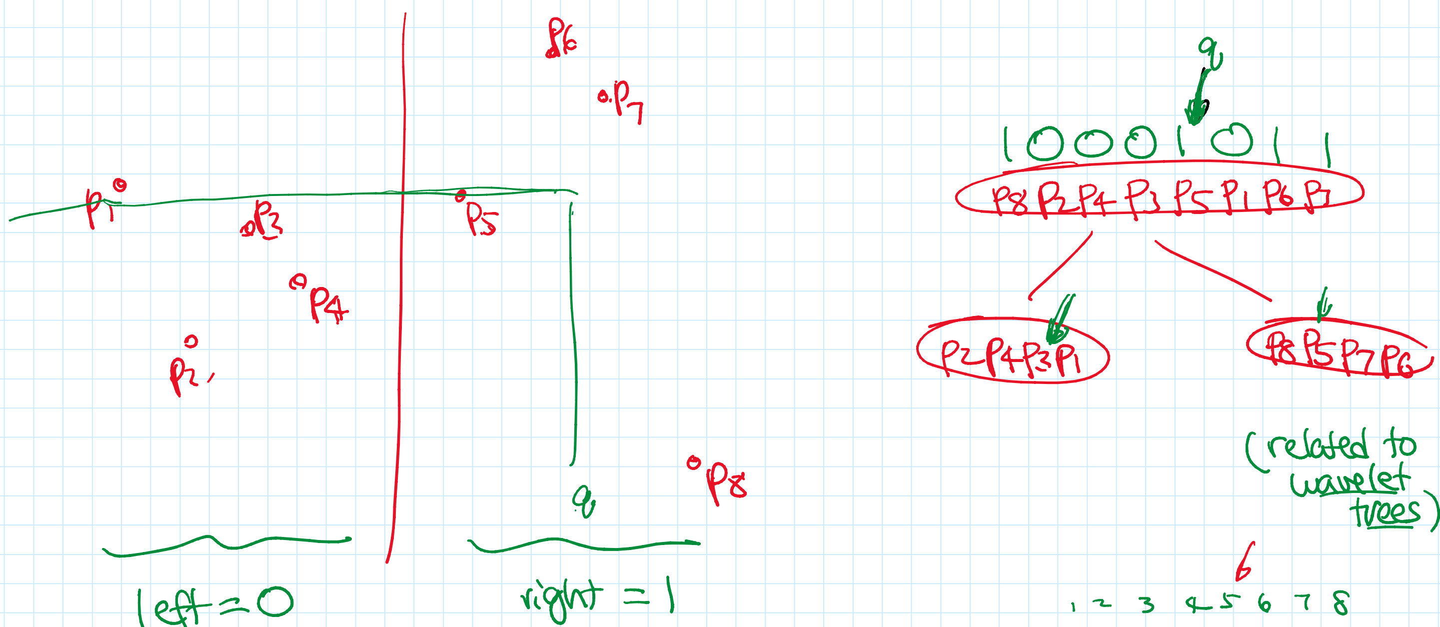
or $$S_d(n) = O\left(n(b\log_b n)^{d-1}\right) \Rightarrow O(n^{1+\varepsilon})$$

$$Q_d(n) = O\left((\log_b n)^{d-1} \log n\right) \Rightarrow O(\log n)$$

Improvement in space for 2D counting: (Chazelle '88)

idea - bit packing.

replace list of y-coords by list of bits



(related to wavelet trees)

e.g. $\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}$ ($i = 6$)

Lemma can store array A of n bits in $O\left(\frac{n}{w}\right)$ words of space ($\frac{n}{w} + o\left(\frac{n}{w}\right)$)

"compact" data structure

s.t. can compute $\text{rank}_1(i) = \#$ 1's in $A[1..i]$

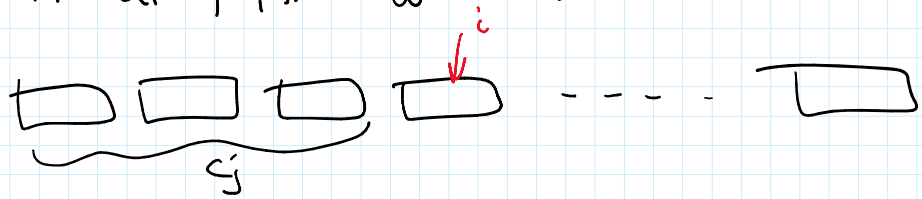
$\text{rank}_0(i) = \#$ 0's in $A[1..i]$

"Compact" data structure
(related to "succinct data structure")

$\mathrm{rank}_0(i) = \#\,0\text{'s in } A[1..i]$

in $O(1)$ time.

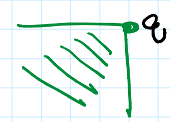
Pf: divide array into $\frac{n}{w}$ words



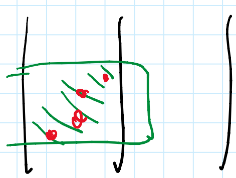
precompute $c_j = \#\,1\text{'s in first } j \text{ words}$ ← $O(\frac{n}{w})$ words

Count # 1's in word by nonstandard word ops
(if $w = \frac{\log n}{2}$, can use table lookup) □

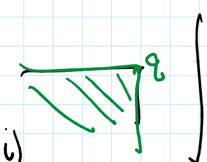
query alg'm, for 2-sided counting:



given position i of q's y-coord:

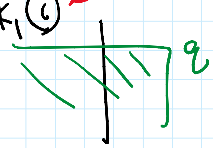


if q left of median x,
recurse on left with position $\mathrm{rank}_0(i)$



else
recurse on right with position $\mathrm{rank}_1(i)$
add $\mathrm{rank}_0(i)$ to count

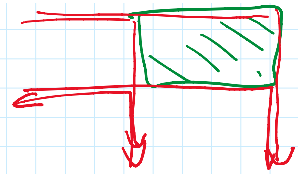
O(1) time



$$\Rightarrow \quad Q(n) = Q\left(\frac{n}{2}\right) + O(1) \Rightarrow \boxed{O(\log n)}$$

space $S(n) = 2S\left(\frac{n}{2}\right) + O\left(\frac{n}{w}\right)$

$$\Rightarrow O\left(\frac{n}{w}\log n\right) \le \boxed{O(n)} \qquad w \ge \Omega(\log n)$$

extend to general 4-sided counting 

4-sided
for reporting: $O(n)$ space

$O(\log n + \underline{k \log n})$ query time

decode each reported pt

Rmk: current best results for 2D reporting

$O(n)$ space, $O(\cancel{\log n} + k \log^{\varepsilon} n)$ query time
$\log\log U + \log^{\varepsilon} n +$

or $O(n \log^{\varepsilon} n)$ space, $O(\cancel{\log n} + k)$ query time
$\log\log U$

or $O(n \log\log n)$ space, $O(\cancel{\log n} + k \log\log n)$ query
$\log\log U$

[C., Larsen, Patrascu '10]

2D counting: $O(n)$ space, $O\left(\frac{\log n}{\log\log n}\right)$ query time

[JaJa et al. '05]

Dynamic: improvement (in 2D) (Mehlhorn-Naher '90)

instead of succ ptrs,
use succ search by vEB
$O(\log\log U) \Rightarrow O(\log n \log\log U)$

map universe $[U]$ to $[n^{O(1)}]$
while preserving order by ~~HW1 Q3~~
(monotone list labeling)

query time
update $O(\log n \log\log n)$ ✓

Rmk: current best result for 2D reporting

query $O\left(\frac{\log n}{\log\log n} + k\right)$

update $\tilde{O}\left((\log n)^{2/3}\right)$ [C., Tsakalidis '17]