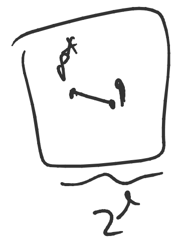
$r^* = d(p^*, q)$.

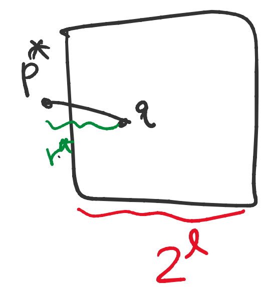
Def q is good if p^* and q lie in a quadtree cell of side length $\le 2(d+1)\, r^*$

Shifting Lemma Shift all pts by random vector in $[U]^d$.

Then q is good with prob $\ge$ const.



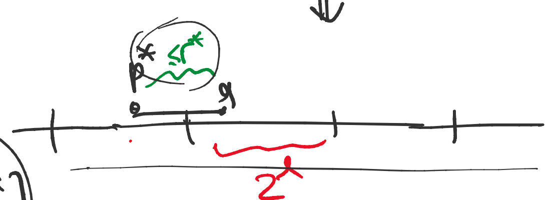
Pf: Say $(d+1) r^* < 2^\ell \le 2(d+1) r^*$



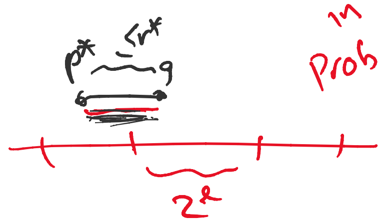
$\Pr[q \text{ bad}]$

$= \Pr[\overline{p^* q}$ crosses boundary of quadfree cell of side length $2^\ell]$

$\Downarrow$



$= \Pr[$ along some dim, shift lies in a bad interval of length $\le r^*]$



in 1D, Prob bad $\le \frac{r^*}{2^\ell}$

$\le d \cdot \frac{r^*}{2^\ell}$

$\le d \cdot \frac{1}{d+1} = 1 - \frac{1}{d+1}$. $\square$

Shifting Lemma: Deterministic Version (C.'97)

Say d even. Let $U = 2^w$.

Shift all pts by vector $v_i = \left(\frac{iU}{d+1}, \ldots, \frac{iU}{d+1}\right)$

Shift all pts by vector $v_i = \left(\frac{iU}{d+1}, \ldots, \frac{iU}{d+1}\right)$

$i = 0, \ldots, d.$

in 2D,
$v_0 = (0, 0)$
$v_1 = \left(\frac{U}{3}, \frac{U}{3}\right)$
$v_2 = \left(\frac{2U}{3}, \frac{2U}{3}\right)$

Then q is good for some i.

Pf: q bad for i along one dim

$\Rightarrow$ $\frac{iU}{d+1} \bmod 2^{\ell}$ lies in a bad interval of length $\le r^*$

$U = 2^w$

$\xRightarrow{\text{multiply by } \frac{d+1}{2^{\ell}}}$ $i2^{w-\ell} \bmod (d+1)$ lies in a bad interval of length $\le \frac{(d+1)r^*}{2^{\ell}} < 1.$

$\Rightarrow$ $i2^{w-\ell} \bmod (d+1)$ is equal to a specific integer.

Fact: $ax \equiv b \bmod m$ has unique sol'n if a and m are relatively prime

$\xRightarrow{\text{by Fact}}$ only one bad i per dim

$\Rightarrow$ only d bad choices for i

$\Rightarrow$ by pigeonhole, $\exists$ good i. $\square$

For each of the $d+1$ shifts,
build compressed quadtree
run naive-ANN-query & take best result

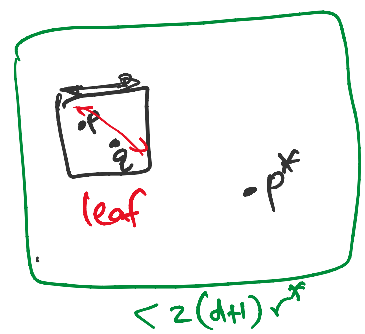
$\Rightarrow$ $\boxed{O(n)}$ space
$\boxed{O(\log U)}$ query time

$\boxed{O(\log U)}$ query ti...

Analysis: Let p be returned pt at leaf.

Suppose q is good.

p^* and q are in quadtree cell of side length $< 2(d+1)r^*$

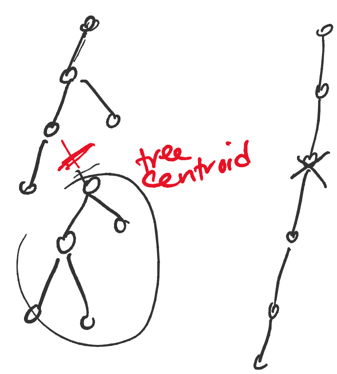


$\Rightarrow \quad d(p,q) < \underline{\sqrt{d}} \cdot 2(d+1)r^*$.

$\Rightarrow \quad$ approx factor $2\sqrt{d}(d+1) = \boxed{O(1)}$.

Q: reduce query time from $O(\underline{\log U})$ to $O(\underline{\log n})$?

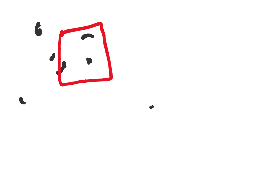
Method 3: Balanced Quadtree



Lemma Given any point set P,
$\exists$ quadtree cell B s.t.

$$|P \cap B| \le \left(\frac{2^d}{2^d+1}\right)n$$

& $$|P \setminus B| \le \left(\frac{2^d}{2^d+1}\right)n$$



Pf: Let B be smallest quadtree cell with $|P\cap B| \ge \frac{1}{2^d+1}n$.

Then $|P \setminus B| \le \frac{2^d}{2^d+1}n$.

For each child B_i of B,

$|P \cap B_i| < \frac{1}{\ldots} n$.

For each child B_i of B,

$$|P \cap B_i| < \frac{1}{2^d+1} n.$$

$$\Rightarrow \quad |P \cap B| \le 2^d \cdot \frac{1}{2^d+1} n \quad \square$$

Recurse in $P \cap B$ & $P \setminus B$

$\Rightarrow$ binary tree of height $\log_{\left(\frac{2^d+1}{2^d}\right)} n$

$= \boxed{O(\log n)}$.

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

space $O(n)$

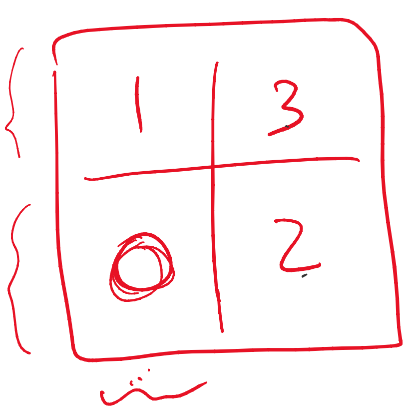
approx factor $O(1)$ by shifting.

[other variants: Arya, Mount et al.'s BBD trees,
Goodrich et al.'s BAR trees, ...]

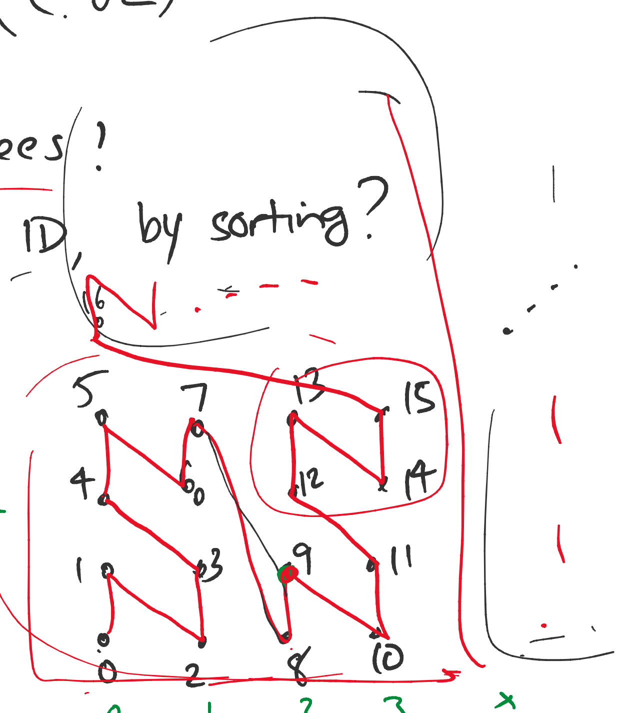
Method 4: Z-Ordering (C.'02)

idea - don't use any trees!

directly reduce to 1D, by sorting?



$\Rightarrow 2$



"space-filling curve"

"space-filling curve"
(other exs: Hilbert curve ...)

Def Given point $p = (x, y) \in [U]^2$, $U = 2^w$

write $x = a_{w-1}\, a_{w-2} \cdots a_0$ in binary
$y = b_{w-1}\, b_{w-2} \cdots b_0$

define $\text{shuffle}(p) = a_{w-1} b_{w-1}\, a_{w-2} b_{w-2} \cdots a_0 b_0$

e.g. $p = (2, 1)$

$x = 2 \rightarrow 010$
$y = 1 \rightarrow 001$

shuffle(p)
$\rightarrow 001001$
$\rightarrow 9.$

Obs Shuffle(p) corresponds to position in Z-order

Shuffle maps a quadtree cell
to an interval contiguous ints in 1D.