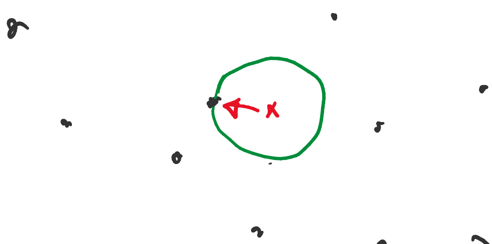
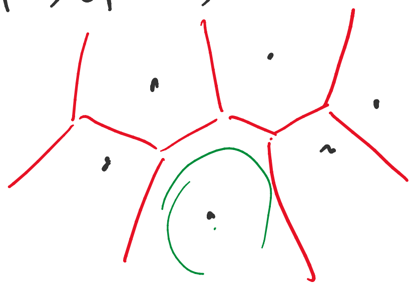
Approximate Nearest Neighbor Search (ANN)

Store n pts P in $\mathbb{R}^d$ (d const)
s.t. given query pt $q \in \mathbb{R}^d$,
find $p \in P$ with $d(p,q) = \min_{p \in P} d(p,q)$.



$d=2$: $O(n)$ space, $O(\log n)$ time

Voronoi diagram

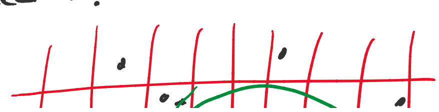


$d \geq 3$: $O^*(n^{\lfloor (d+1)/2 \rfloor})$ space, $O(\log n)$ time
or $O(n)$ space, $O^*(n^{1-1/\lfloor (d+1)/2 \rfloor})$ time.

In approx version,
find $p \in P$ with $d(p,q) \leq \underbrace{(1+\varepsilon)}_{\text{approx factor}} \cdot \min_{p' \in P} d(p',q)$

Approx decision version: given r,
return some pt of dist. $\leq (1+\varepsilon) r$
or declare all pts have dist $> r$.

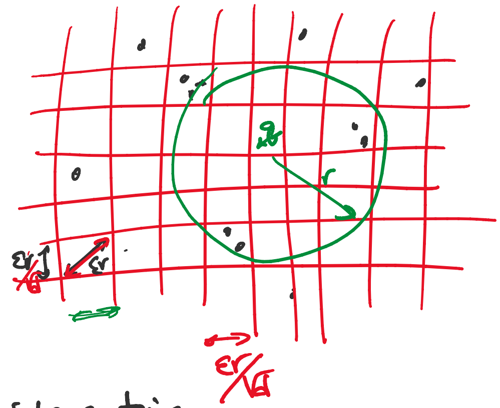
Method 0: for decision with fixed r.



Method 0: for decision

form uniform grid of side length $\varepsilon r/\sqrt{d}$

store S = all nonempty grid cells



query (q):

check if any grid cell intersecting ball(q, r) is in S.
by hashing

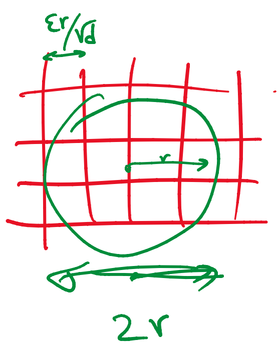
Space: $\boxed{O(n)}$

Time: $O($ # grid cells intersecting ball(q,r) $)$

$$= O\left(\left\lceil \frac{2r}{\varepsilon r/\sqrt{d}} \right\rceil^d\right)$$

$$= \boxed{O\left(\left(\frac{1}{\varepsilon}\right)^d\right)}$$

Const if ε is const.



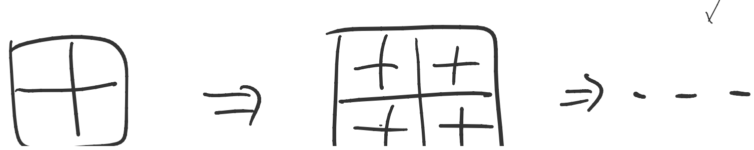
$$\left(\frac{r}{\varepsilon r/\sqrt{2}}\right)^2$$

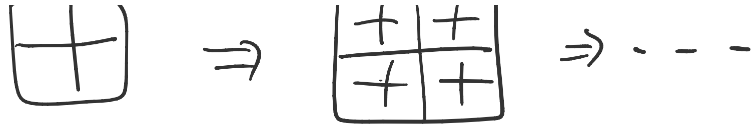
Rmk. alternatively, $\boxed{O\left(\left(\frac{1}{\varepsilon}\right)^d n\right)}$ space

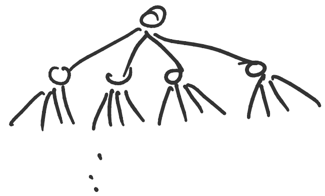
$\boxed{O(1)}$ query time

Method 1: r not fixed $\Rightarrow$ **Quadtree**

idea - hierarchy of grids

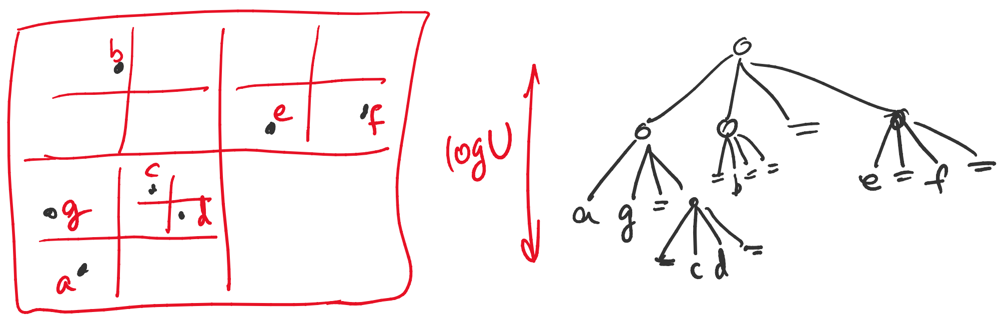


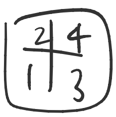


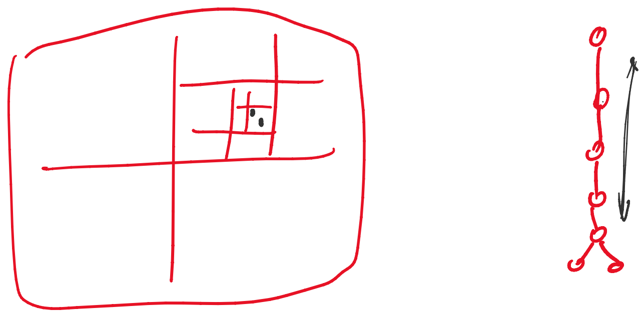


Def A quadtree cell B is a grid cell of side length 2^ℓ for some $\ell \in \{0, 1, \ldots, \log U\}$

e.g.

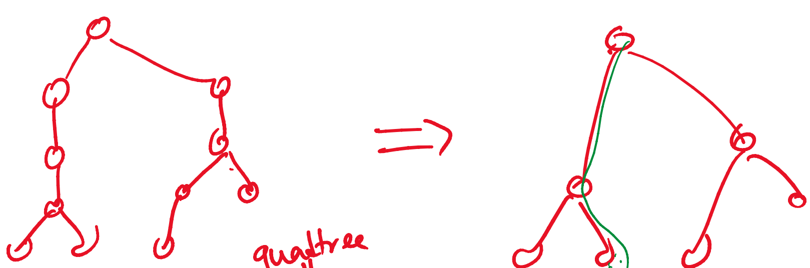






space $\boxed{O(n \log U)}$

reducible to $\boxed{O(n)}$
by compressed quadtree
(shortcutting deg-1 nodes)



quadtree cell

decision-query(B, q, r): // top down

if ball(q,r) does not intersect B, return

... any pt in B.

decision-query(B, q, r):

if ball(q,r) does not intersect B, return

if B has side length $< \frac{\varepsilon r}{\sqrt{d}}$ return any pt in B.

for each child B_i of B

    decision-query(B_i, q, r)

$$\text{time} = O\left(\begin{array}{l}\#\text{ quadtree cells of side length} \geq \frac{\varepsilon r}{\sqrt{d}} \\ \text{intersecting ball}(q,r)\end{array}\right).$$

$$= O\left(\sum_{\ell:\, 2^\ell > \frac{\varepsilon r}{\sqrt{d}}} \left\lceil \frac{\cancel{2}r}{2^\ell} \right\rceil^d \right).$$

$$= O\left(\left\lceil \frac{r}{\varepsilon r/\sqrt{d}} \right\rceil^d + \left\lceil \frac{r}{2\varepsilon r/\sqrt{d}} \right\rceil^d + \cdots \right)$$

$$= O\left(\left(\frac{1}{\varepsilon}\right)^d + 1 + \left(\frac{1}{2\varepsilon}\right)^d + 1 + \cdots \right)$$

$$= \boxed{O\left(\log U + \left(\frac{1}{\varepsilon}\right)^d \right)}$$

How to adapt to ANN?

option 1. by binary search

query time increases by log U

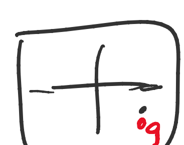
or log(log U) ?

option 2. modify query alg...

Method 2: Quadtree with Shifting (Bern '93)

first aim for const factor approx.

naive-ANN-query(B, q):



find child B_i containing q

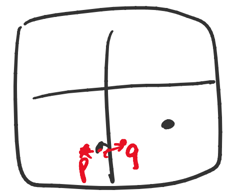
... (B_i, q)

naive-A...

find child B_i containing q

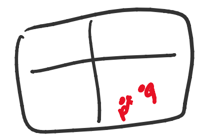
return naive-ANN-query(B_i, q)

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



Let $p^* =$ nearest neighbor of q

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



Def q is <u>good</u> if p^* and q lie in a quadtree cell of side length $< \underbrace{2(d+1)\, r^*}$

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

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