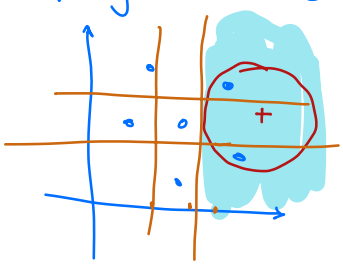
Query Answering in low D

Range Queries

Given (q, ρ),

$$O = \{ t \in D \mid \delta(v(t), q) \leq \rho \}$$

for simplicity, let $d = 2$:



k-D Tree: Creates a Tree partitioning of the data, by following a round-robin approach on the dimensions.

- at every iteration, it divides the points by half along dimension d_i

e.g.



bcz every d rounds, one branch is pruned ⇒ touched leaves get divided by half

$$|X| = n\left(\tfrac{1}{2}\right)^{\frac{\log n}{d}} = n^{1-1/d}$$ cells intersect with an axis-parallel line

$$\Rightarrow |\text{intersecting cells}| \leq 2d\left(n^{1-1/d}\right)$$ ← the # of faces of a rectangle range

if $d = 2$: $4\left(n^{1/2}\right) = \Theta(\sqrt{n})$

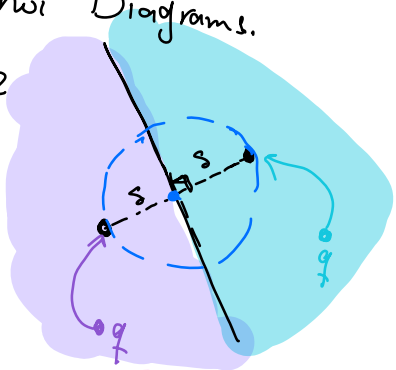
Nearest-neighbor queries

Given (q):

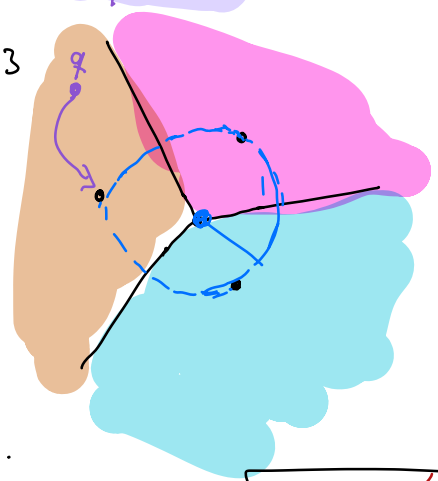
$$\underset{t \in D}{\operatorname{argmin}}\ \delta(v(t), q)$$

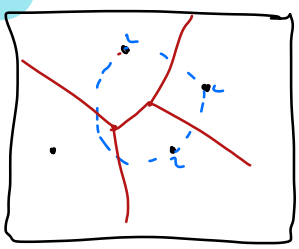
Voronoi Diagrams.

$|D| = 2$



$|D| = 3$





observations:

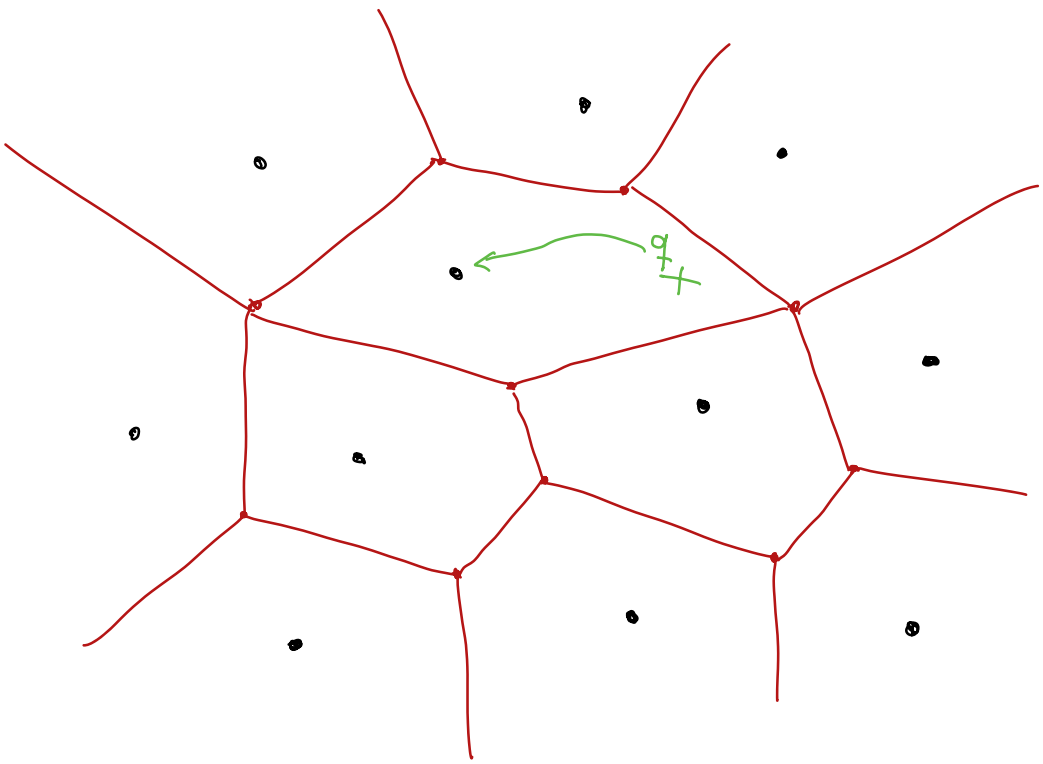
① # cells = n

② Voronoi-Vertex: the intersection of bisecting lines

↳ $\deg(v) = 3$

Lemma: The number of voronoi vertices is around $2n$, the # edges is around $3n$ $\Rightarrow$ The complexity of a voronoi diagram is $\Theta(n)$.

↳ Extension: In general a voroni diagram has the complexity of $\Theta(n^{\lceil d/2 \rceil})$



To answer the nn-query using voronoi Diagram, we ONLY need to find which cell it belongs to.

$\Rightarrow \Theta(\log n)$ using a balanced binary search.

$\Rightarrow$ In general: $\Theta(d \log n)$ query Time

K-Voronoi Diagrams ⟶ an extension of Voronoi Diagrams where the K-NN of each cell is unique.