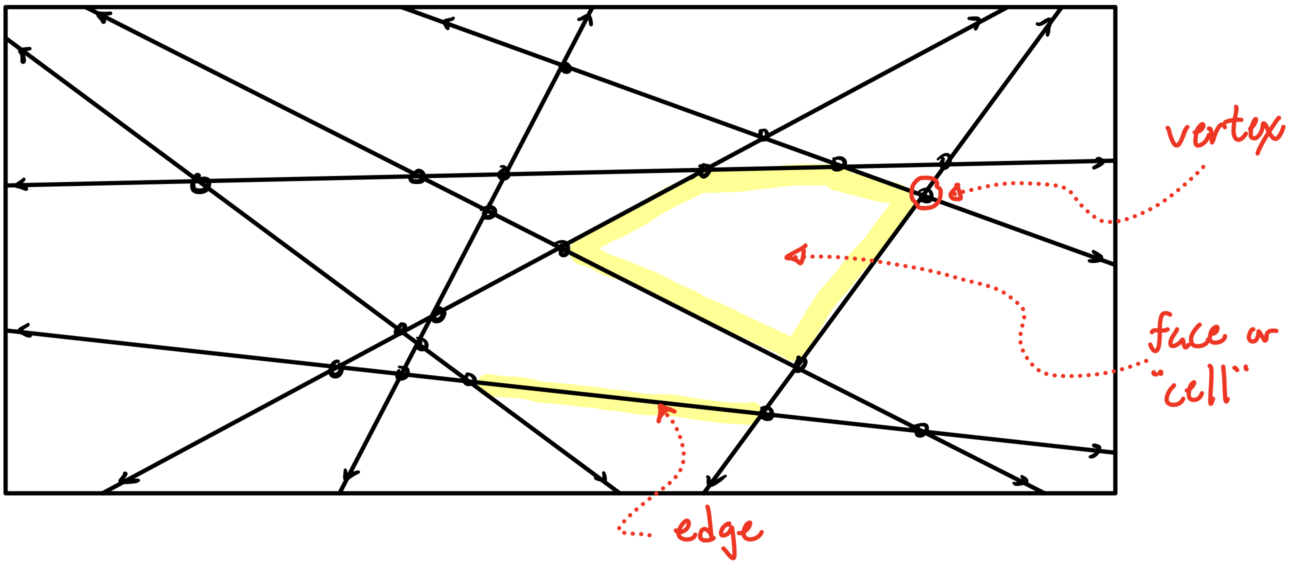
CMSC 754 - Computational Geometry
Lecture 13 - Line Arrangements

Arrangement:

Given a set $L = \{\ell_1, \dots, \ell_n\}$ of lines in $\mathbb{R}^2$ (generally $(d-1)$-dim hyperplanes in $\mathbb{R}^d$), they subdivide the plane into a cell complex called the arrangement of L, or $A(L)$.



Combinatorial Properties:

Lemma: Given n lines L in gen'l position in $\mathbb{R}^2$:

(i) $A(L)$ has $\binom{n}{2} = \frac{1}{2} \cdot n(n-1)$ vertices

(ii) $A(L)$ has n^2 edges

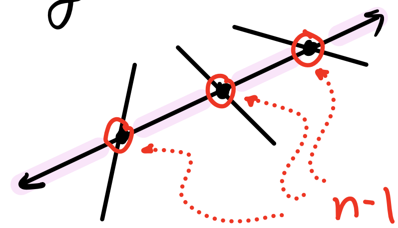
(iii) $A(L)$ has $\binom{n}{2} + n + 1 = \frac{1}{2}(n^2+n+2)$ cells

Proof:

(i) Each pair intersects once = $\binom{n}{2}$ ✓

(ii) Each line is split by $n-1$ others into n edges

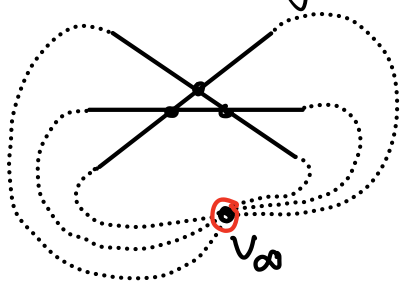
$\Rightarrow$ n^2 total ✓



(iii) Add a vertex at ∞ of degree n to tie off all unbounded edges

$v = \binom{n}{2} + 1$

$e = n^2$



By Euler's formula:

$$v - e + f = 2$$

$$\Rightarrow \left(\binom{n}{2}+1\right) - n^2 + f = 2$$

$$\Rightarrow f = 2 + n^2 - \left(\binom{n}{2}+1\right)$$

$$\Rightarrow f = 2 + n^2 - \frac{n(n-1)}{2} - 1$$

$$= \frac{1}{2}(n^2 + n + 2) \checkmark \qquad \square$$

[In $\mathbb{R}^d$, complexity is $\Theta(n^d)$]

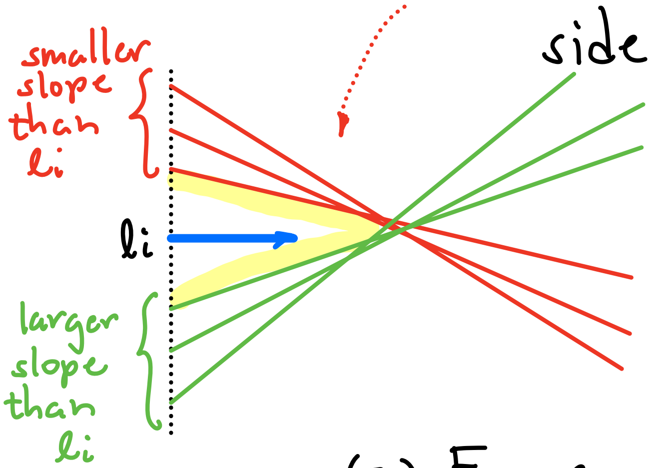
Incremental Construction: (not randomized)

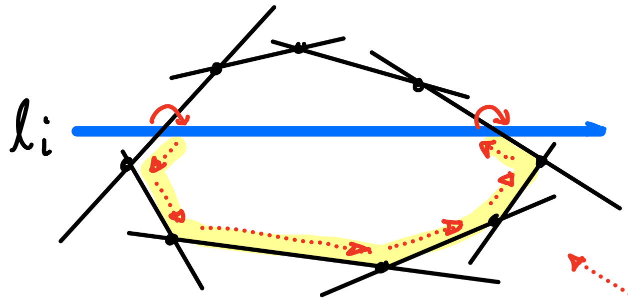
Idea: Add lines one by one (in any order)
Update the structure after each

Notation: $L_i = \{\ell_1, \ldots, \ell_i\}$

How to add the i^{th} line? ℓ_i

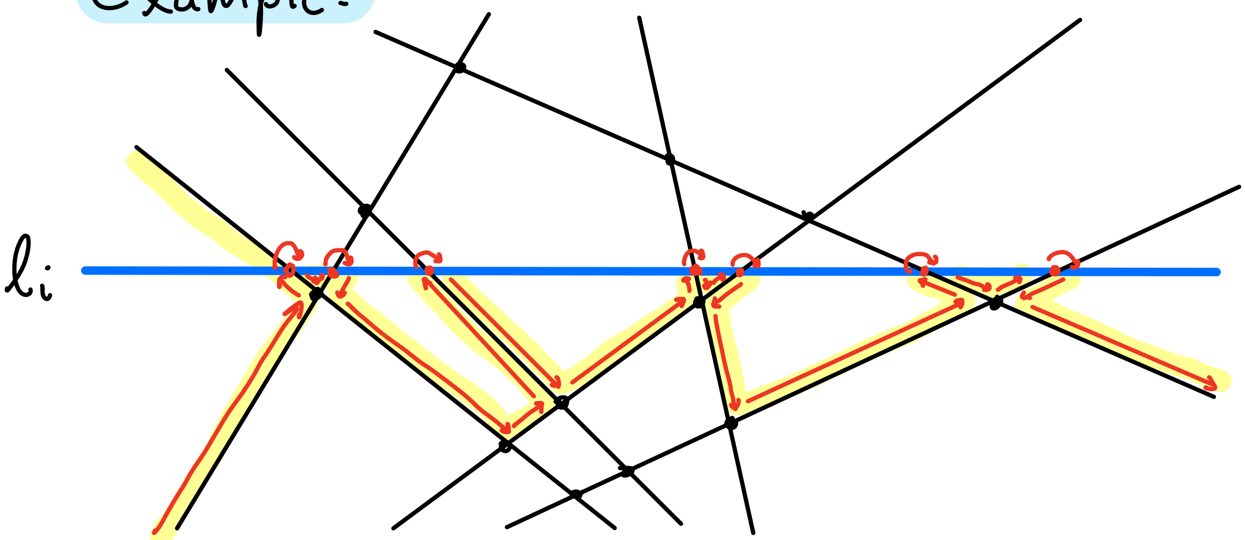
(1) Find the unbounded cell on left side where ℓ_i starts (slope based)





(2) For each face of $\mathcal{A}(L_{i-1})$ that intersects ℓ_i, walk along its lower boundary to determine where it exits this cell

Example:



- Once we know entry-exit points on each face - we update arrangement in $O(i)$ time (DCEL)
- How long to crawl around edges?

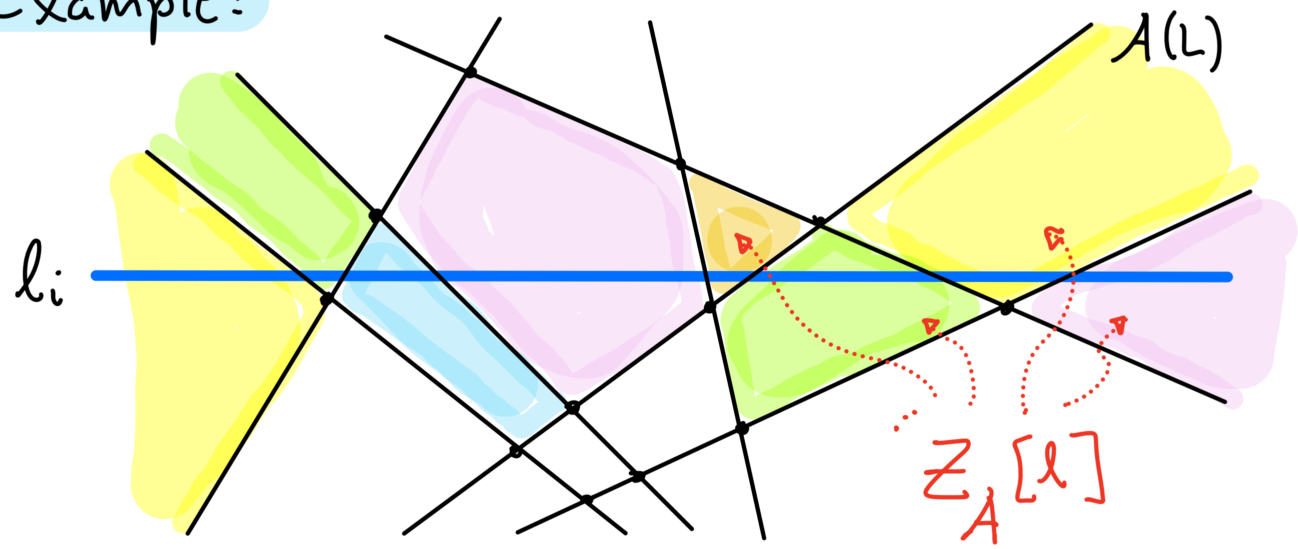
Naive analysis: On adding ℓ_i

- ℓ_i crosses i cells
- each cell may have as many as $i-1$ edges
- crawl takes $O(i(i-1)) = O(i^2)$ time
- total time $\cong \sum_{i=1}^{n} i^2 = O(n^3)$

Can it really be this bad?

Zone: Given an arrangement $A = A(L)$ and a line $\ell \notin L$, zone of ℓ in A, $Z_A(\ell)$ is the set of cells of A that ℓ intersects.

Example:



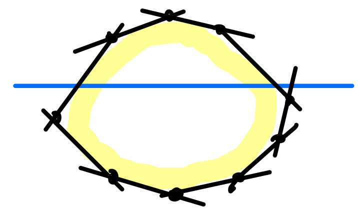
Obs: Crawl time $\leq$ no. of edges on the zone of ℓ_i in $A(L_{i-1})$ $[Z_{A(L_{i-1})}(\ell_i)]$

→ We'll show this is $O(i)$ not $O(i^2)$

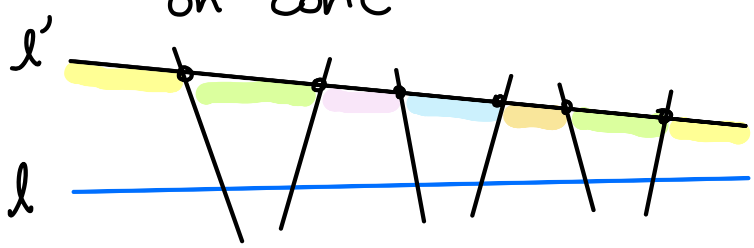
Theorem: (Zone Theorem) Given an arrangement $\mathcal{A}(L)$ where $|L| = n$ and any line $\ell \notin L$, the number of edges in $Z_{\mathcal{A}}(\ell) \leq 6n$

How to prove this?

cell by cell? Some cells have high complexity



line by line? Some lines appear many times on zone



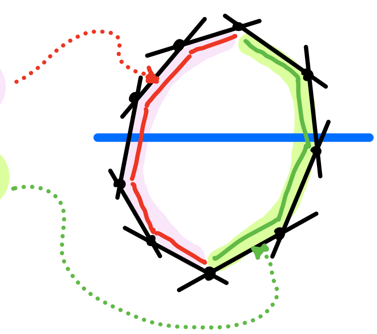
Our approach:

- Partition edges of zone into two classes (left side + right side)
- Show (by induction) at most 3n of each

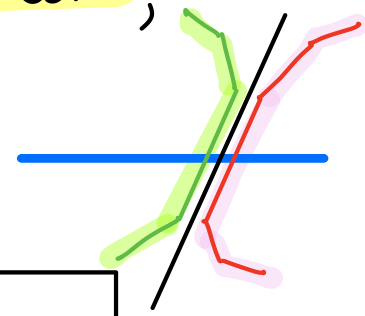
A zone edge is:

left bounding: on left side of cell

right bounding: on right side of cell



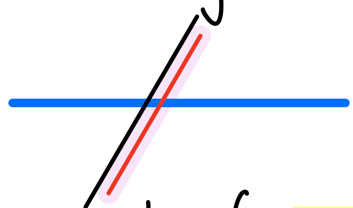
Note: Some edges appear twice in the zone, both as left/right bounding



$\boxed{\text{Claim: At most } 3n \text{ left-bounding edges.}}$

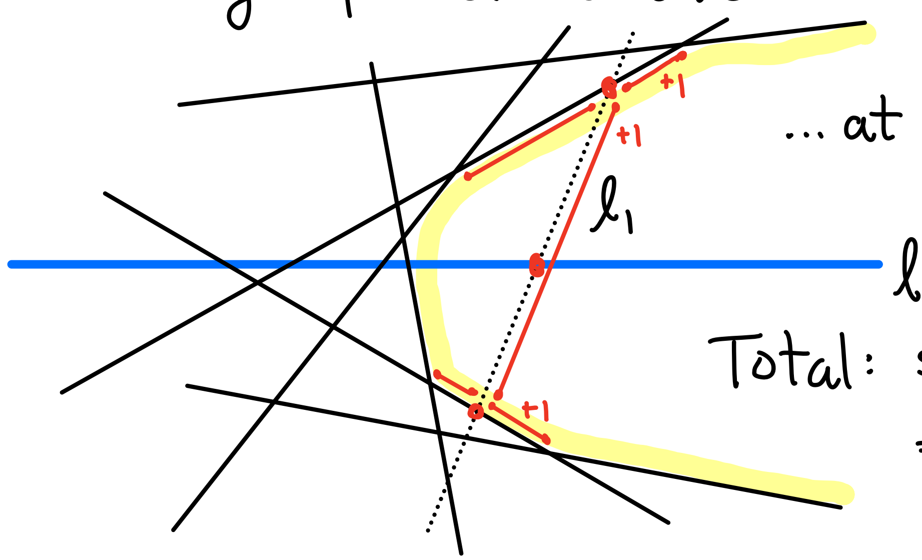
Proof: By induction on n

$n = 1$: Just one LB edge $1 \leq 3 \cdot 1$ ✓



$n \geq 2$: I.H. arrangement of $n-1$ lines has $\leq 3(n-1)$ LB edges in zone

- Let $\ell_1 \in L$ be rightmost line to cross ℓ
- Removing $\ell_1 \Rightarrow$ at most $3(n-1)$ LB edges
- Adding ℓ_1 back creates ...



... at most 3 new LB edges

Total: $\leq 3(n-1)+3$
$= 3n$ $\square$

Thm: Given a set L of n lines in $\mathbb{R}^2$, $\mathcal{A}(L)$ can be built in time $O(n^2)$ [and has size $O(n^2)$ ... so this is optimal]

Proof:
- Apply incremental construction
- Inserting ℓ_i takes time ~ no. of edges in $Z_{\mathcal{A}(L_{i-1})}(\ell_i) \leq 6(i-1)$
- Total time $\leq \sum_{i=1}^{n} 6(i-1) = 6\sum_{i=0}^{n-1} i = O(n^2)$

Applications:

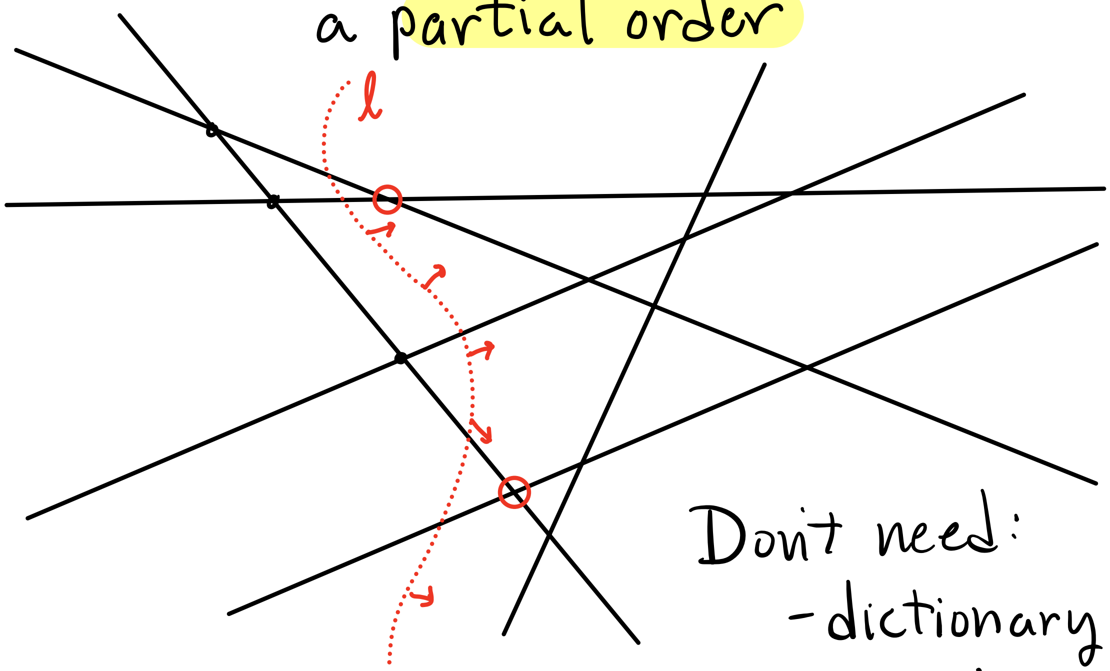
Line arrangements can be used to solve many problems
- mostly $O(n^2)$ time
- often using duality

How to process an arrangement?
- Build it + traverse it like a graph
  $O(n^2)$ time, $O(n^2)$ space
- Plane sweep
  $O(n^2 \log n)$ time, $O(n)$ space
- Topological plane sweep
  $O(n^2)$ time, $O(n)$ space

  Not covered, but applicable pretty much whenever plane sweep is. ← you may assume this

Topological plane sweep:

- A relaxed version of plane sweep
- Vertices are not swept in strict left to right order, but based on a partial order



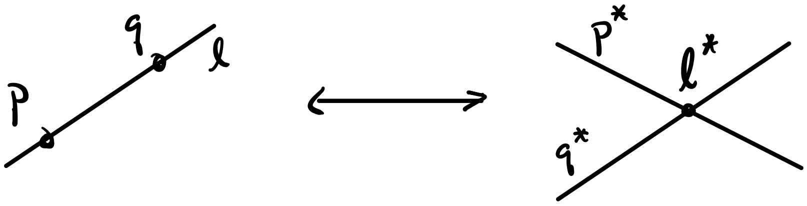
Don't need:
- dictionary
- priority queue

⇒ saves $\log n$ factor

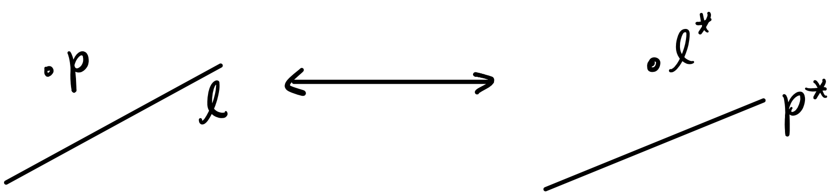
Recall: Dual transformation

$p = (a, b) \longleftrightarrow p^*: y = ax - b$

$\ell: y = ax - b \longleftrightarrow \ell^*: (a, b)$



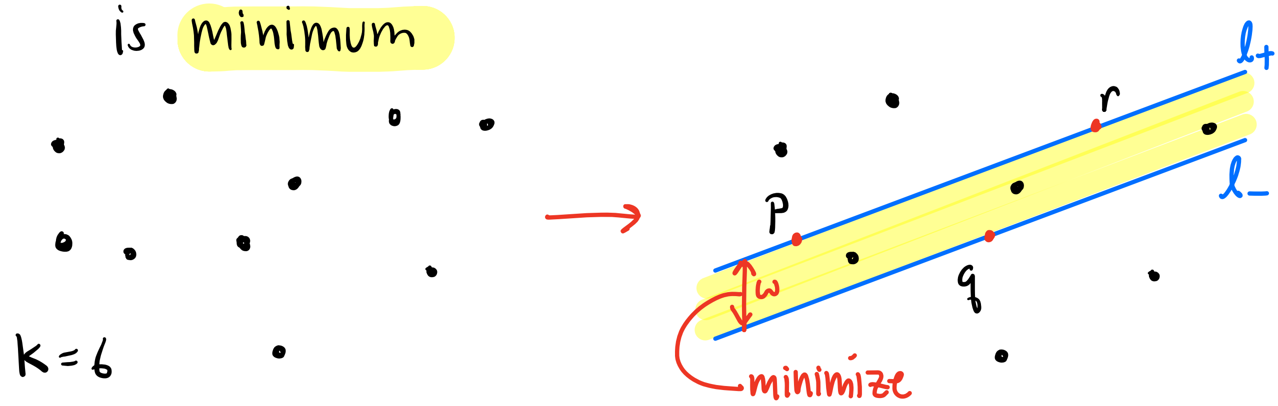
incidence preserving



order reversing

Narrowest k-corridor:

- Given a set $P = \{p_1, \dots, p_n\}$ in $\mathbb{R}^2$ and integer $3 \le k \le n$, find pair of parallel (non vertical) lines that enclose k pts so that vertical distance between lines is minimum



Primal form:

- Let ℓ_+ + ℓ_- be upper + lower lines of "slab"

parallel ⇒ same slope

$$\ell_+ : y = ax - b_+ \qquad b_+ \le b_-$$
$$\ell_- : y = ax - b_-$$

order reversed due to negation

- Vertical width: $w = b_- - b_+$
- k pts of P lie on or between ℓ_- + ℓ_+

Local optimality:

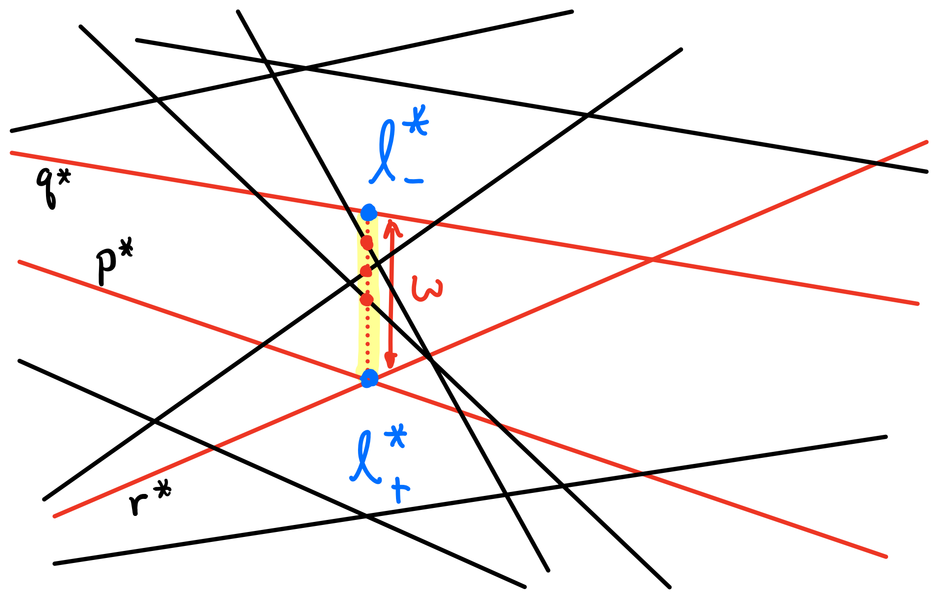
3 pts of P will lie on ℓ_+ + ℓ_-, 2 on one edge + 1 on other

- If 0, 1, or 2 can make width smaller
- If 4 or more – not gen'l position

vertical line segment

Dual form:

- ℓ_+^* + ℓ_-^* are pts (a, b_+) + (a, b_-)
- vertical distance $b_- - b_+$
- k lines of P^* pass through or between these pts

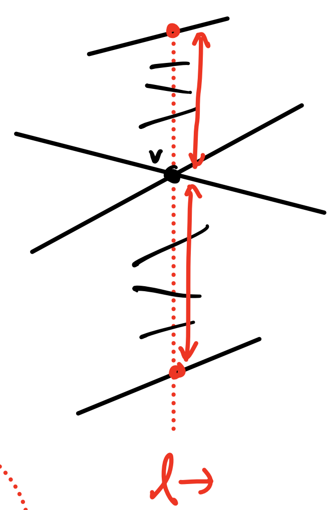


Local optimality:

3 lines will pass through ℓ_-^* + ℓ_+^* with 2 on one side + one on other

Narrowest-Corridor(P, k):

(1) $P^* \leftarrow$ dual lines of P
(2) Plane sweep through P^*.
(3) On arriving at each vertex v, compute vertical distance to lines k-2 above + k-2 below
(4) Return smallest such distance



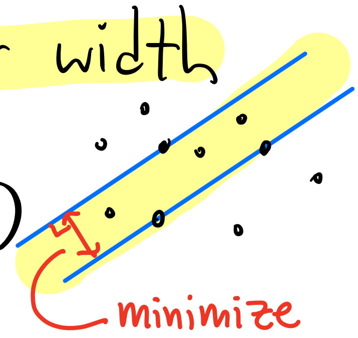
Correctness: (Argued above)

Can access in $O(1)$ time since sweep line can be stored in array

Time: $O(n^2 \log n)$ time + $O(n)$ space

↳ can reduce to $O(n^2)$ by topol. plane sweep.

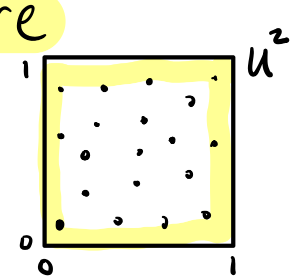
Aside: It is easy to generalize this to minimize perpendicular width (Just apply a correction factor when computing widths)



Halfplane Discrepancy:

Let $U^2 = [0,1]^2$ denote the unit square

Given n pts $P = \{p_1, \ldots, p_n\} \subset U^2$, how close is P to being uniformly distributed over U^2?



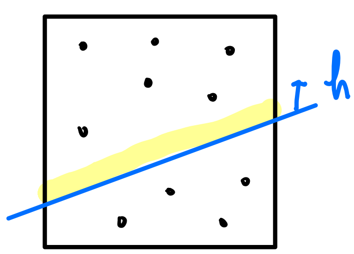
Idea:

For any halfplane h, let

$\mu(h) = \text{area}(h \cap U^2)$ $\quad [0 \le \mu(h) \le 1]$

the fraction of P in h

$\mu_P(h) = |h \cap P| / |P|$ $\quad [0 \le \mu_P(h) \le 1]$



$\mu(h) = 2/3 = 0.666\ldots$

$\mu_P(h) = 6/10 = 0.6$

If P is uniformly distrib., we expect

$$\mu(h) \approx \mu_P(h) \quad \forall h$$

To measure how uniform is P, define:

$$\Delta(P) = \max_h |\mu(h) - \mu_P(h)|$$

$[0 < \Delta(P) \leq 1]$

can't be perfect

Called the halfplane discrepancy of P

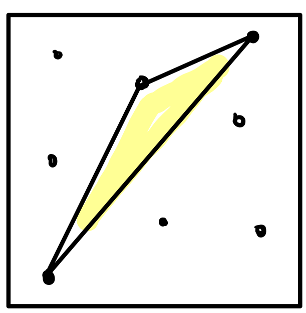
Questions:

- Given $P \subset U^2$, what is $\Delta(P)$?
- How low can $\Delta(P)$ be for any set of size n?
- How to generate optimally uniform set P_{OPT} of a given size n? ($\Delta(P_{OPT})$ is min. possible)
- Other measures of discrepancy?
  - Triangle discrepancy
  - Heilbronn's Triangle Problem:

    Given any set of n pts P in U^2, how large can the min area triangle be?

    Conj: $O(1/n^2)$

    Open for a century!



Computing $\Delta(P)$ for a set $P \subset U^2$.

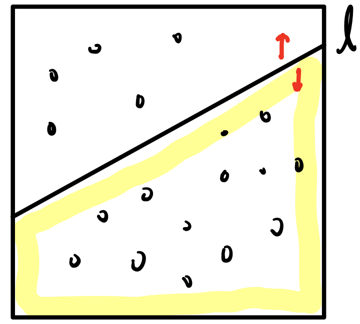
Key: Identify $O(n^2)$ candidates for halfplane that maximizes discrepancy.
- Compute discrepancy for each
- Return the max

> **Lemma:** Given pt set P, let h be halfplane of max discrepancy. Let ℓ be h's bounding line. Either:
> (i) ℓ passes through pt $p_i \in P$, and p_i is midpoint of $\ell \cap U^2$
> (ii) ℓ passes through two pts of P.

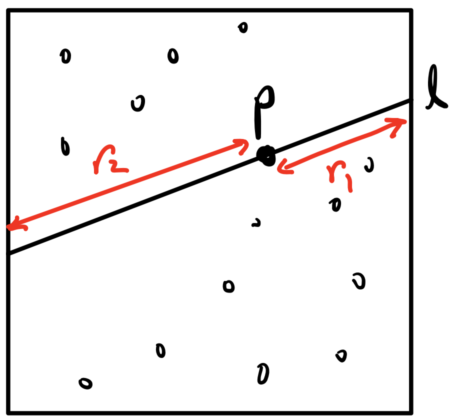
Proof:

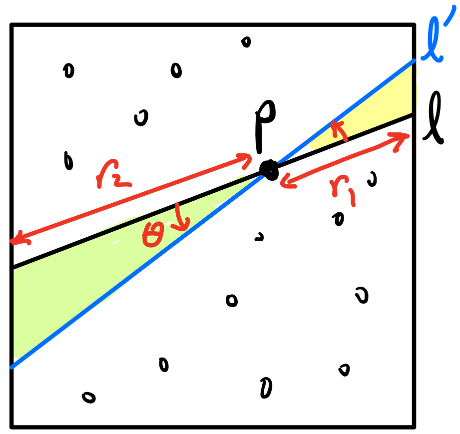
Approach: Consider any line ℓ. We'll show unless it satisfies (i) or (ii) we can perturb it to increase discrepancy.

Case 1: ℓ passes through no pt of P – perturbing ℓ up or down increases discrepancy.



Case 2: ℓ passes through a pt $p \in \mathcal{P}$, but p is not midpt of $\ell \cap U^2$





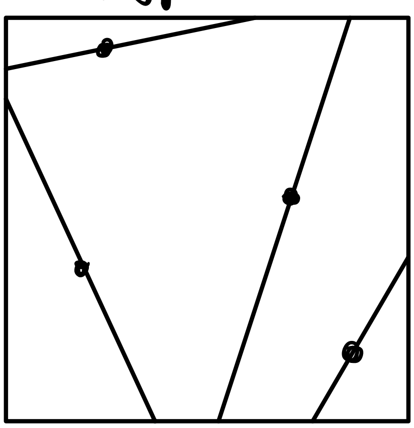
p splits $\ell \cap U^2$ into two segments of lengths $r_1 + r_2$. Since p is not midpt, may assume w.l.o.g. $r_2 > r_1$

If we rotate ℓ by small angle θ about p we increase/decrease area by $\sim$

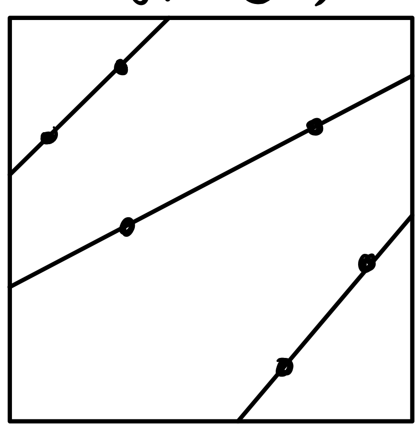
$$r_2^2 \cdot \theta - r_1^2 \theta = (r_2^2 - r_1^2)\theta > 0$$

Some small rotation will increase discrepancy. $\square$

Type (i)



Type (ii)



Computing $\Delta(P)$:

Type (i):

- for each $p_i \in P$, compute lines ℓ s.t. p_i on midpt of $\ell \cap U^2$
    - Count no. of pts on either side of ℓ

→ n pts ; $O(1)$ lines each ; $O(n)$ time to count $\Rightarrow O(n^2)$ time

Type (ii):

- Dualize P to P^*
- Perform plane sweep of arrangement $\mathcal{A}(P^*)$
- For each vertex of arrangement maintain no. of lines above + below on sweep line
- Compute discrepancy in $O(1)$ time for each vertex

→ $O(n^2)$ vertices

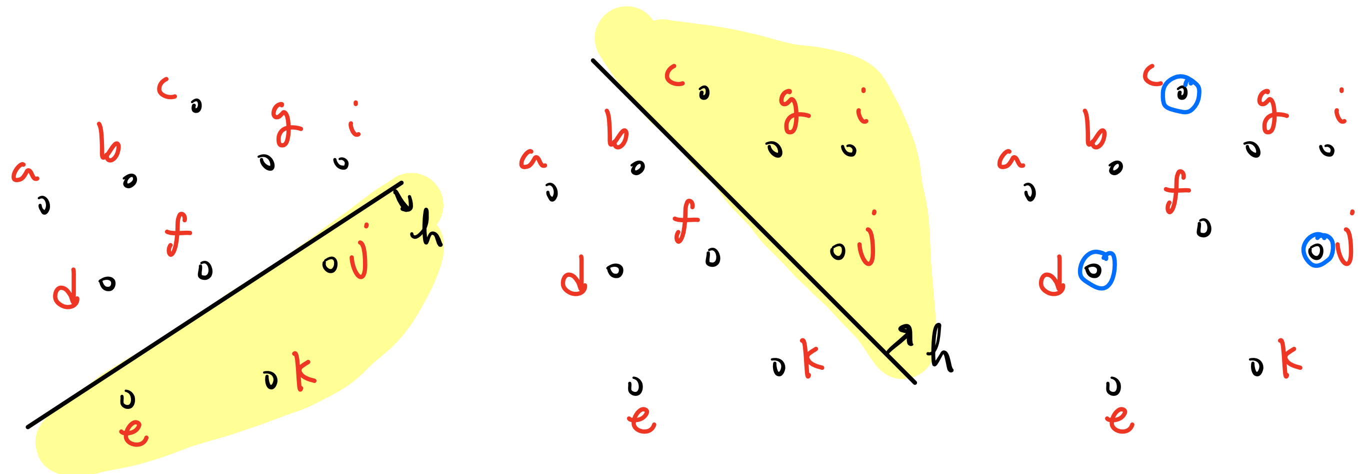
Can maintain counts in $O(1)$ time

$\Rightarrow O(n^2 \log n)$ time + $O(n)$ space

$O(n^2)$ by topol plane sweep

Computing k-sets:

Given a set $P = \{p_1, \dots, p_n\}$ in $\mathbb{R}^2$ and integer k, $1 \le k \le n-1$, a **k-set** is a **k-element subset of P of the form $P \cap h$**, for some halfplane h.



$\{e, k, j\}$ is a **3-set**

$\{c, g, i, j\}$ is a **4-set**

$\{c, d, j\}$ is **not** a 3-set

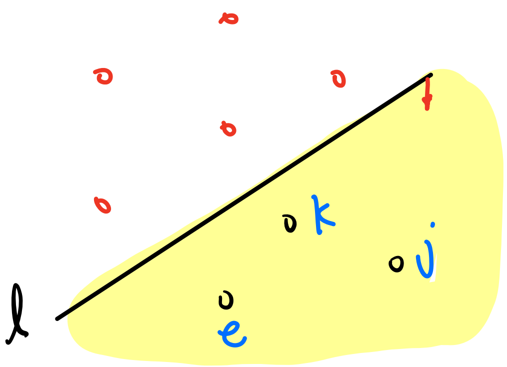
Problem: Given P and k, **enumerate all k-sets of P.**

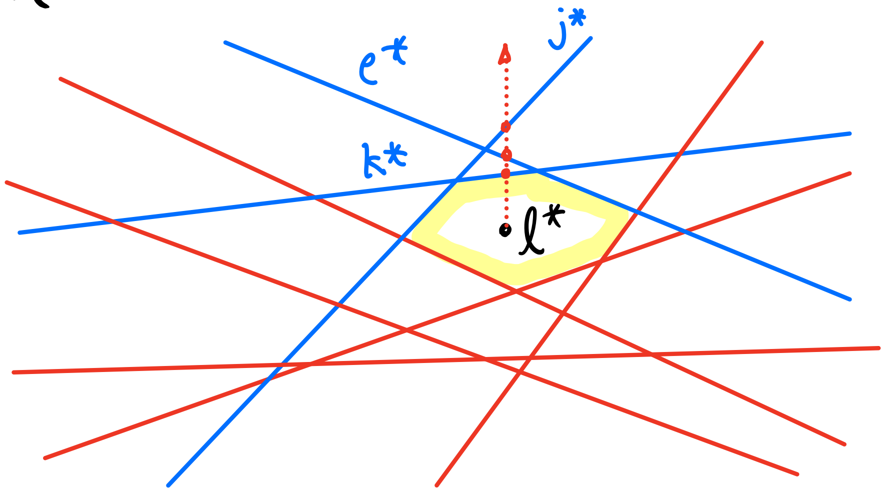
How many? **Naive** $\le \binom{n}{k} = O(n^k)$

Better $\le \binom{n}{2}$ (see below)

Best theoretic bounds: $O(n \log k) .. O(n k^{1/3})$

Dual equivalent?





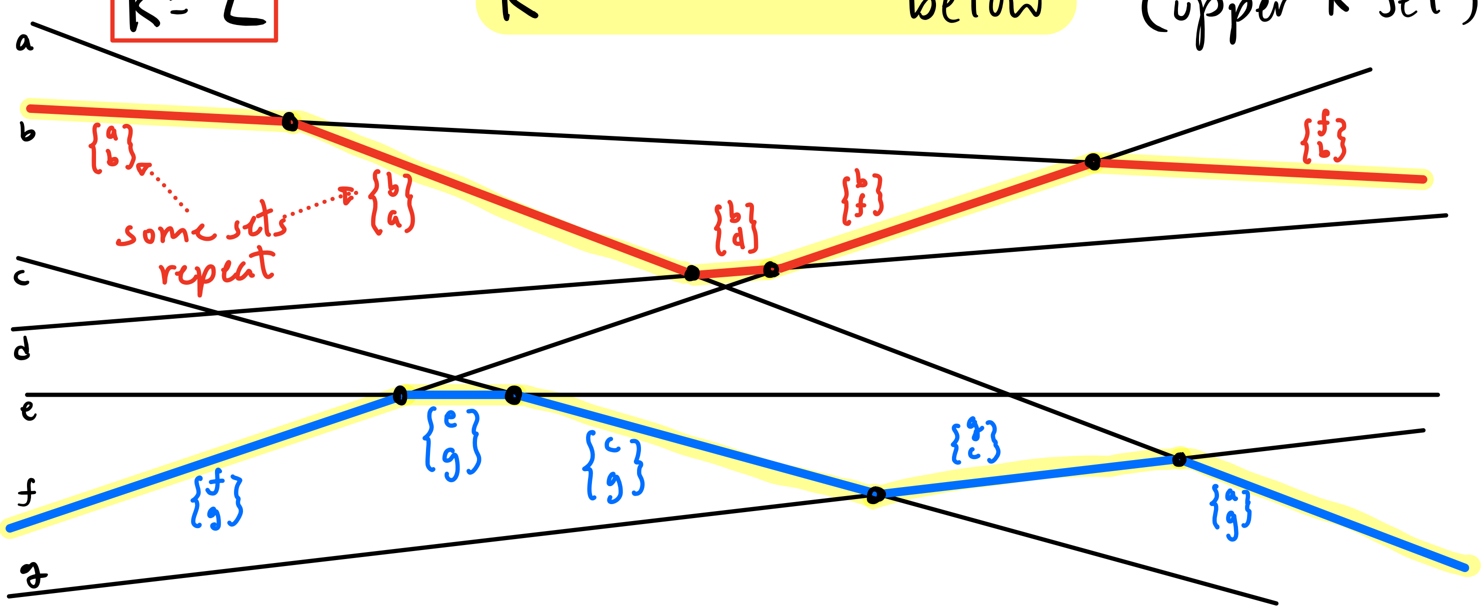
By order reversal:

k-pts of $\mathcal{P}$ lie below ℓ $\iff$ k-lines of $\mathcal{P}^*$ pass above ℓ^*

Approach:

- Traverse the arrangement $\mathcal{A}(\mathcal{P}^*)$
- Identify all edges with
  - k lines on or above (lower k-set)
  - k " " " below (upper k-set)

$\boxed{\begin{array}{l} n=7 \\ k=2 \end{array}}$



Level: Given an arrangement of n lines $A(L)$, for $1 \le k \le n$, define **level k, $\mathcal{L}_k$**, to be set of pts in $A(L)$ with

$\le k-1$ lines (strictly) above

$\le n-k$ lines (strictly) below

In above fi vre, we have shown $\mathcal{L}_2$ and $\mathcal{L}_6$
g

Obs: By applying plane sweep through $A(L)$, we can construct all levels in time $O(n^2)$

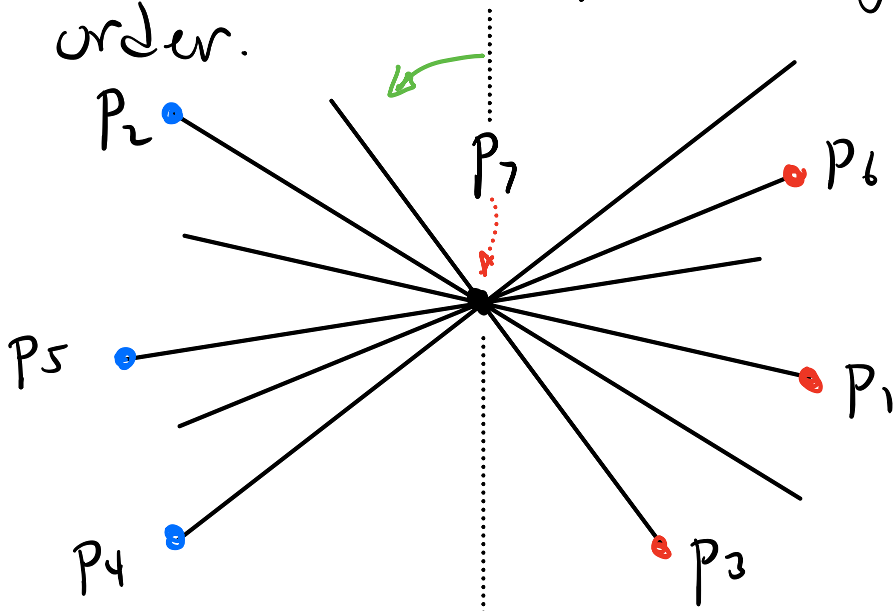
$\Rightarrow$ Can identify all k-sets of P in time $O(n^2)$ by sweeping $A(P^*)$ + extracting levels $\mathcal{L}_k$ + $\mathcal{L}_{n-k+1}$

Note: To actually list the sets adds addional k factor, total $O(k \cdot n^2)$

Avoid duplicates? Exercise

Sorting angular sequences:

Given a set $P = \{p_1, \dots, p_n\}$ in $\mathbb{R}^2$, for each p_i, sort the remaining $n-1$ pts around p_i in angular order.



$$\begin{bmatrix} 2 \\ 5 \\ 4 \\ 3 \\ 1 \\ 6 \end{bmatrix}$$

Naive: $O(n(n \log n)) = O(n^2 \log n)$
Sort angles for each point

Better: $O(n^2)$ using arrangements.

[See lect. notes for details]