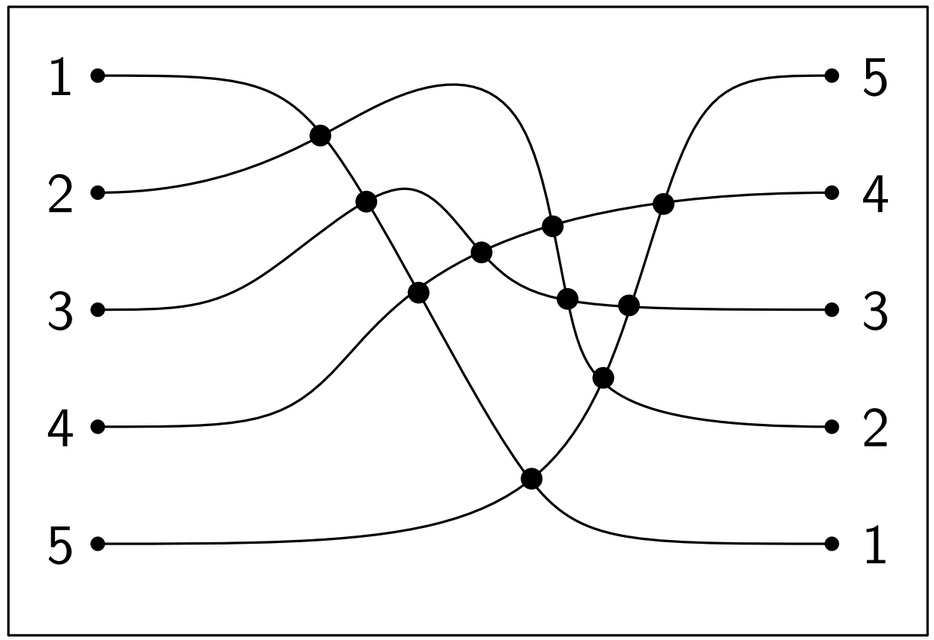
# **NumPSLA** — An experimental research tool for . . . **pseudoline arrangements** and (abstract) **order types**

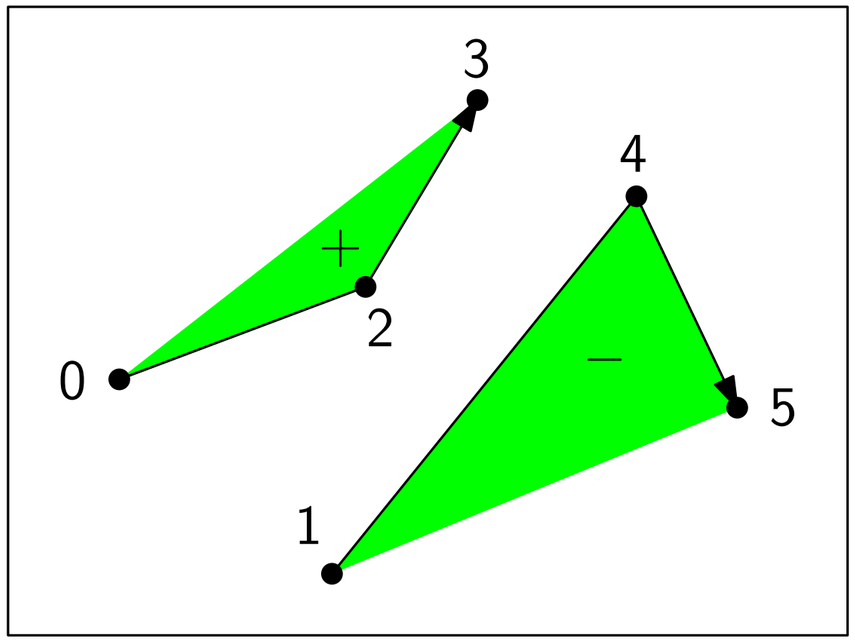
Günter Rote
Freie Universität Berlin

PSLA

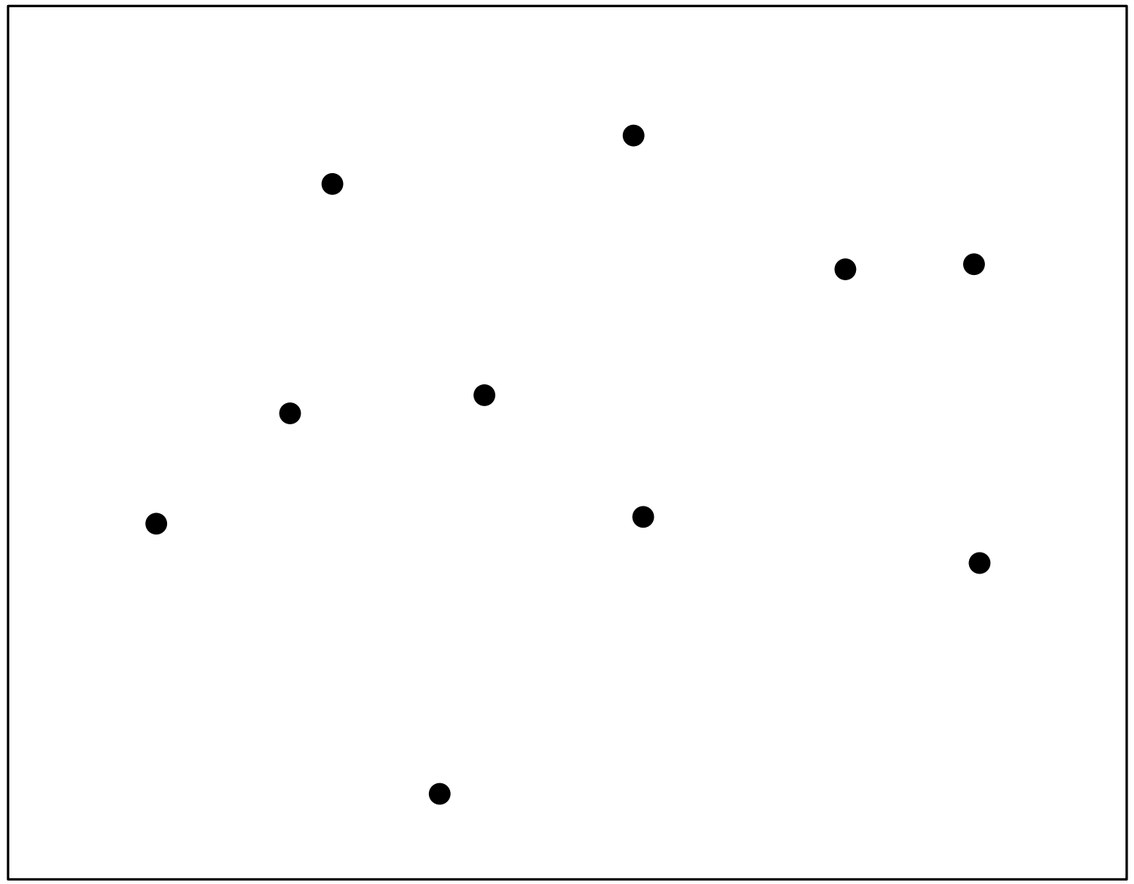



point-line duality

AOT

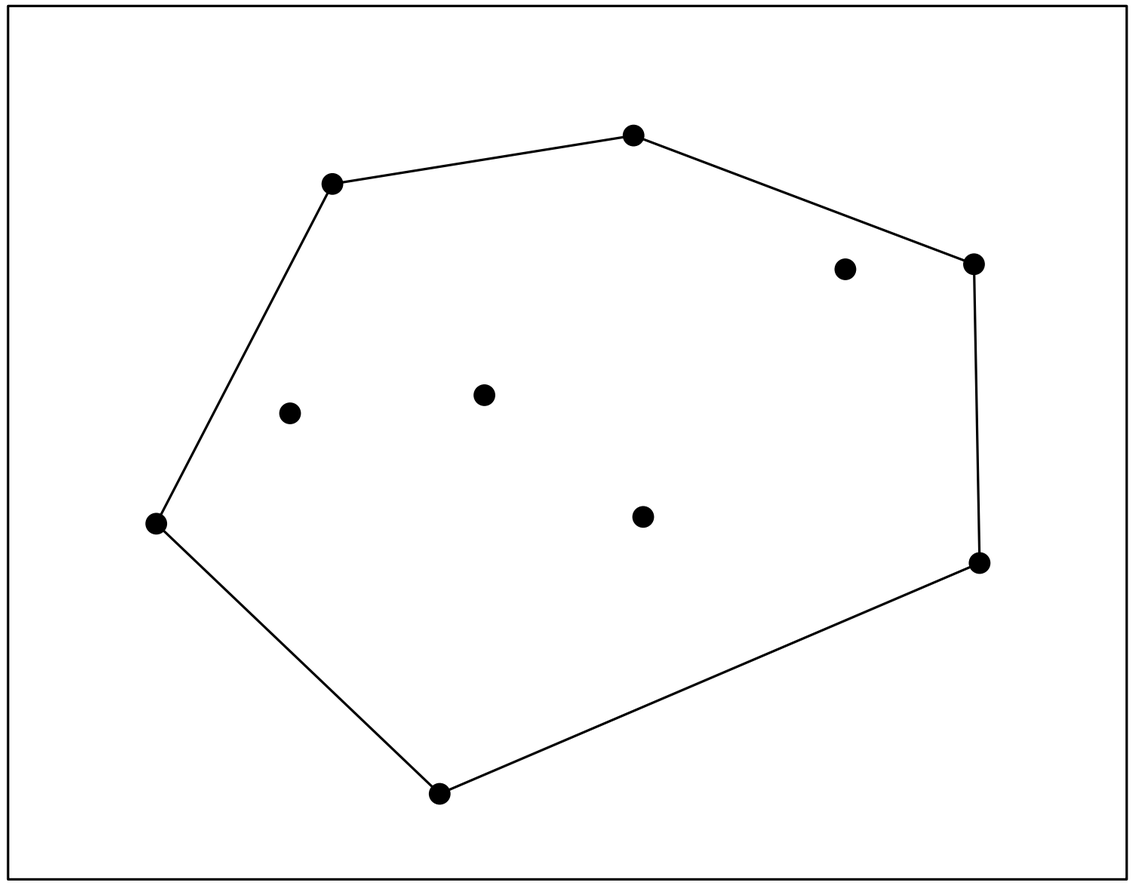


# Orientation predicate and order type



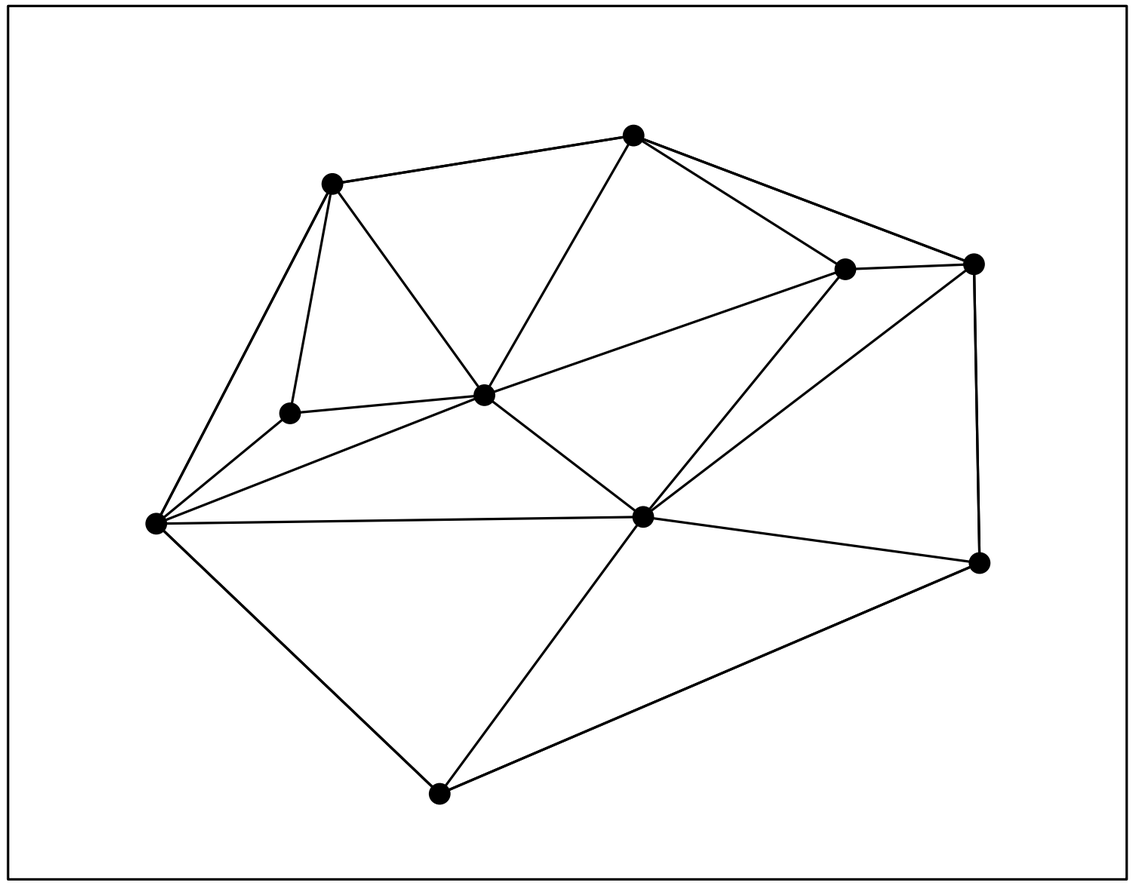
- How many convex hull vertices $(h)$

# Orientation predicate and order type



- How many convex hull vertices ($h$)

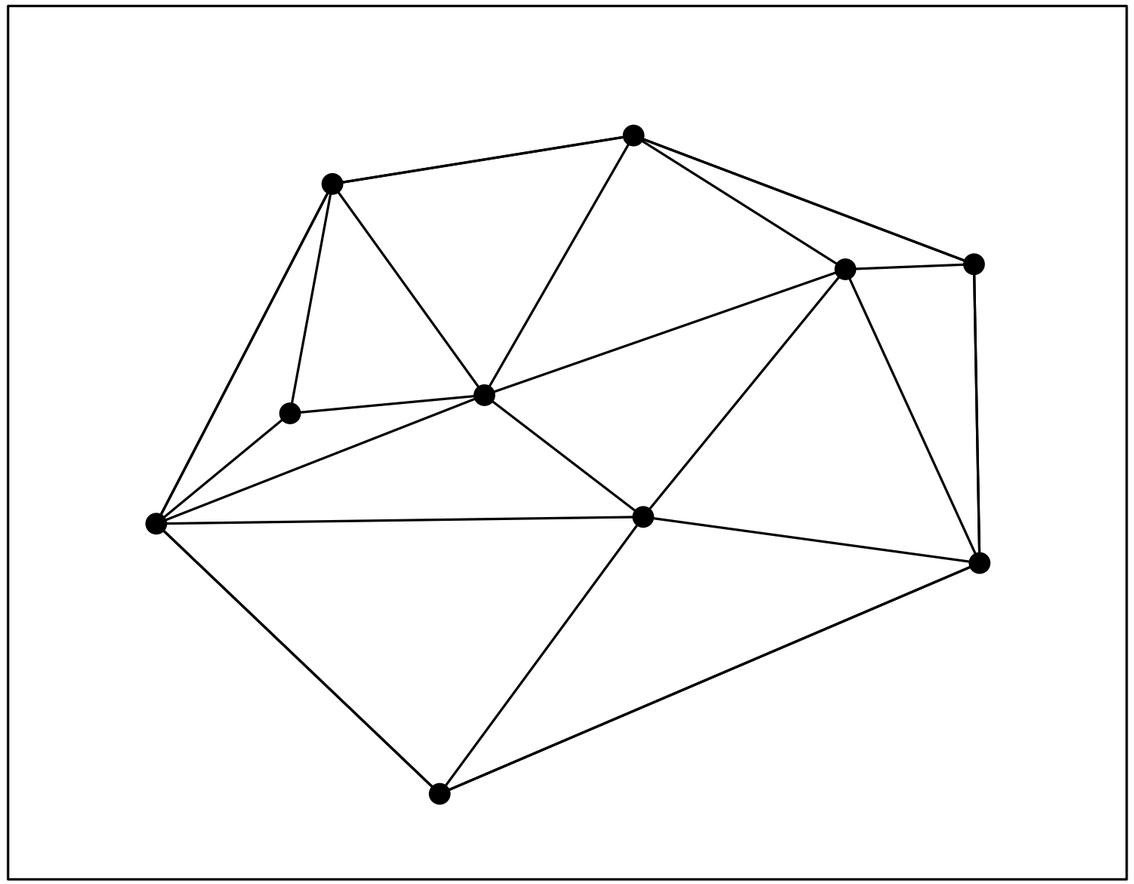
# Orientation predicate and order type



- How many convex hull vertices $(h)$
- How many triangulations?

(GENERAL POSITION!)

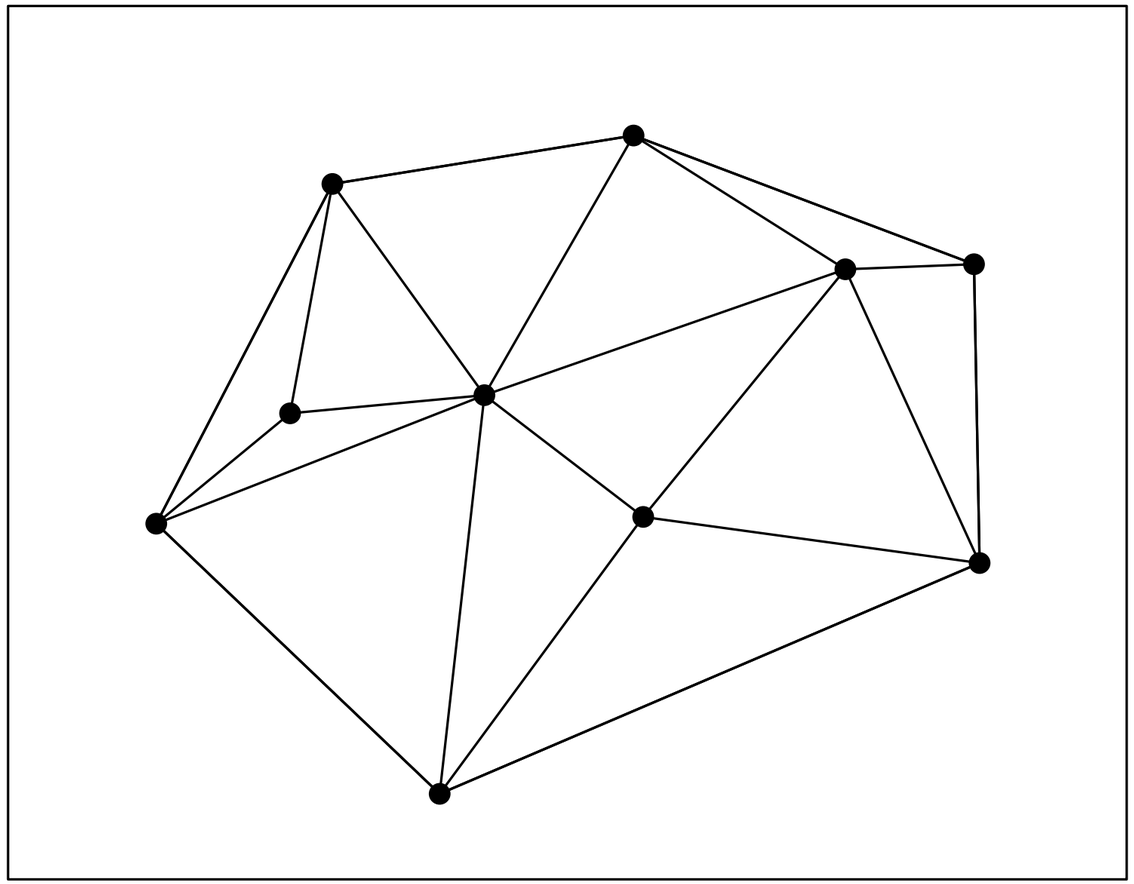
# Orientation predicate and order type



- How many convex hull vertices $(h)$
- How many triangulations?

(GENERAL POSITION!)

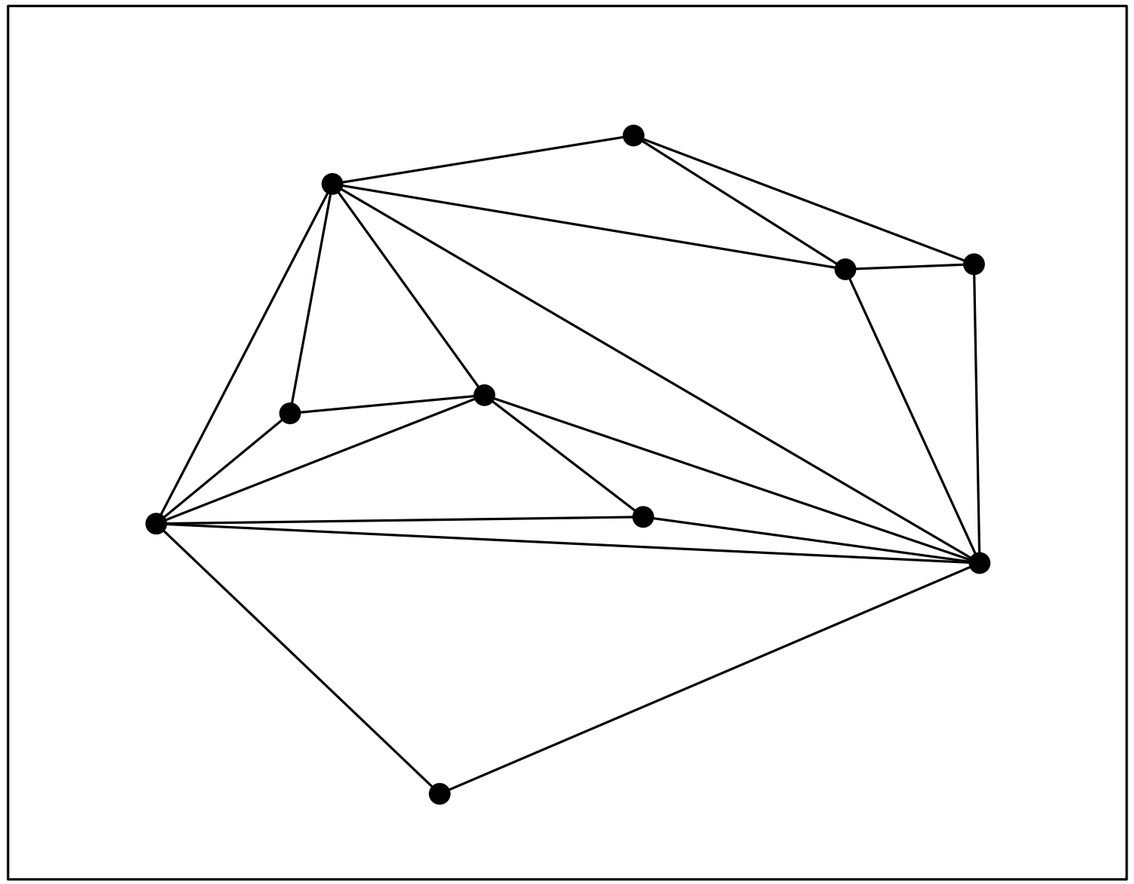
# Orientation predicate and order type



- How many convex hull vertices $(h)$
- How many triangulations?

(GENERAL POSITION!)

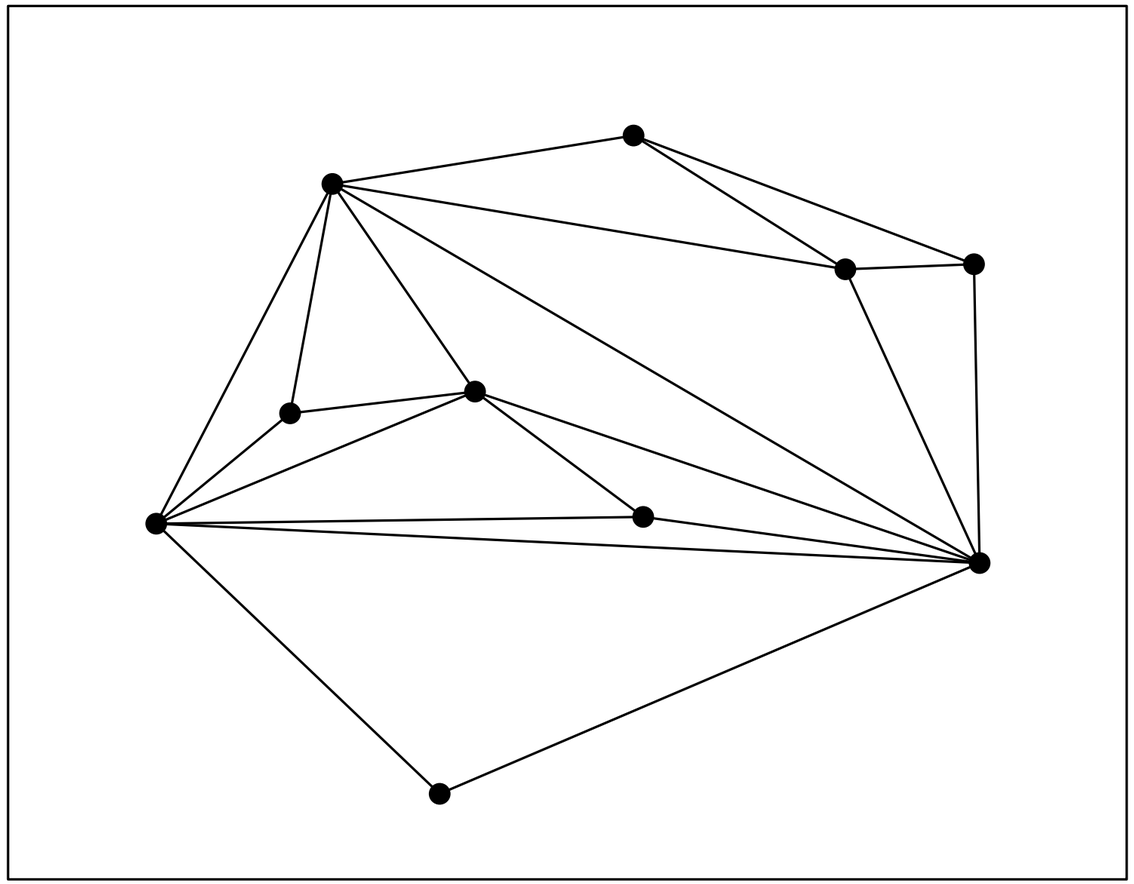
# Orientation predicate and order type



- How many convex hull vertices $(h)$
- How many triangulations?

(GENERAL POSITION!)

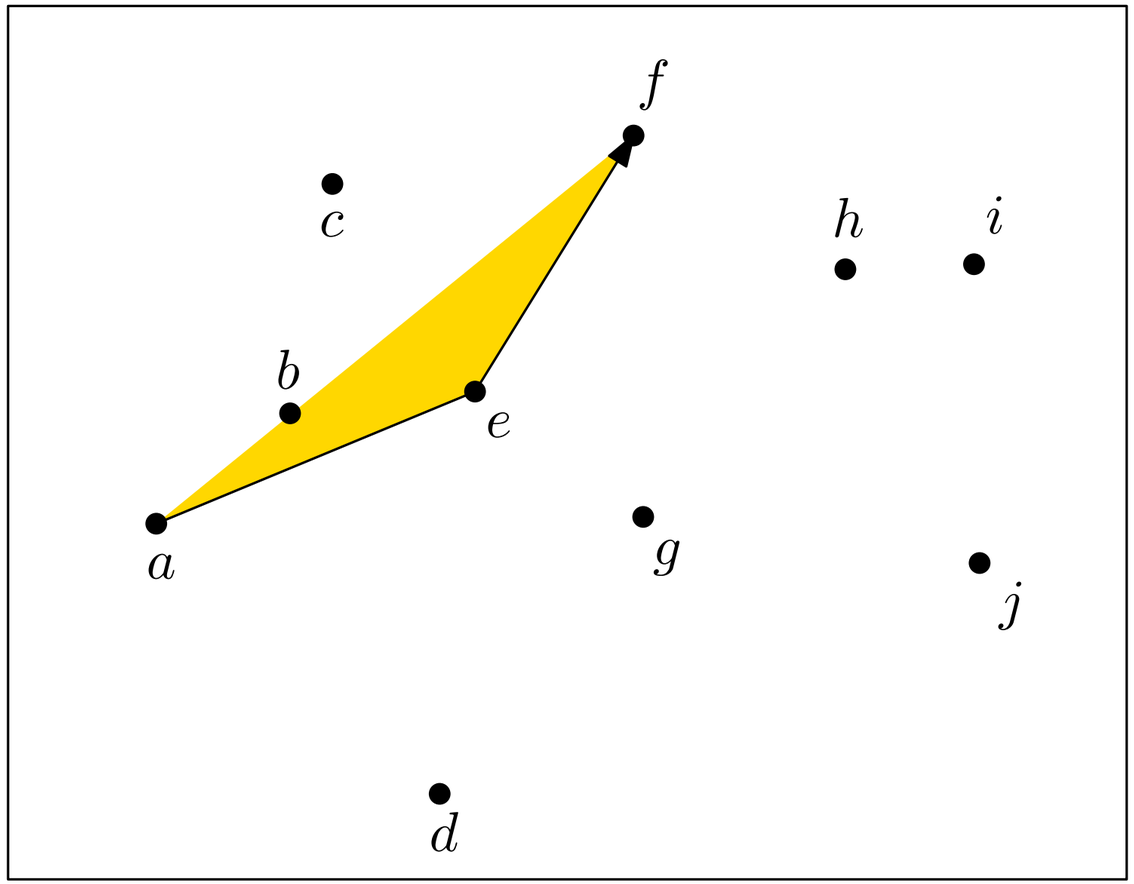
# Orientation predicate and order type



- How many convex hull vertices $(h)$
- How many triangulations?

(GENERAL POSITION!)

# Orientation predicate and order type

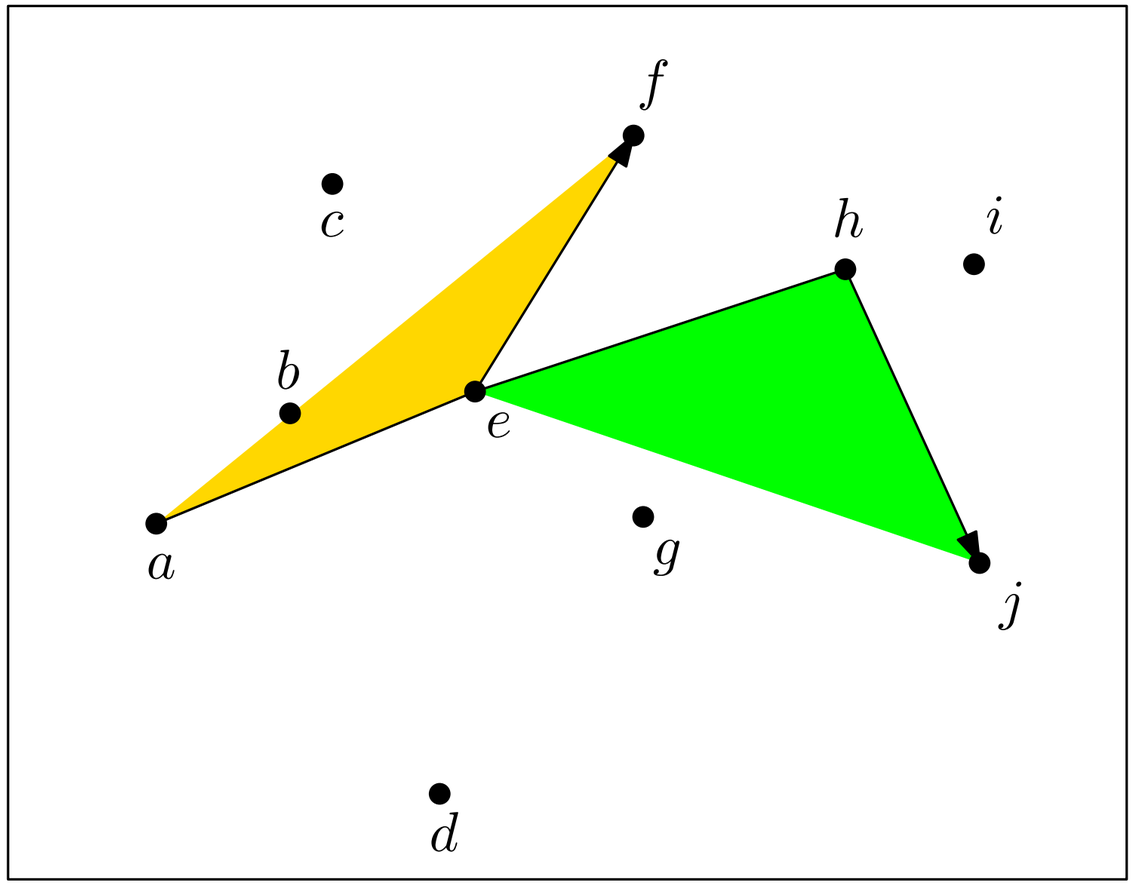



- How many convex hull vertices ($h$)
- How many triangulations?

(GENERAL POSITION!)

$aef$: $+$

# Orientation predicate and order type

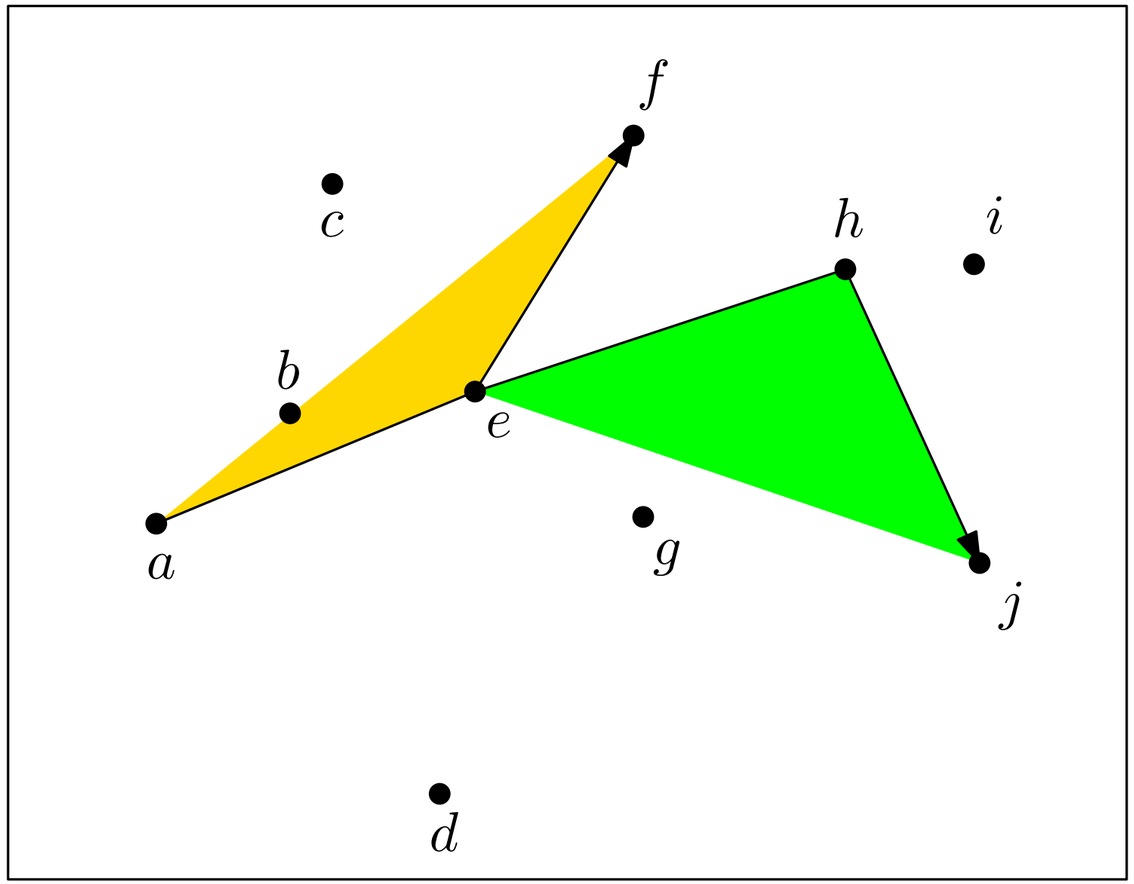



- How many convex hull vertices ($h$)
- How many triangulations?

(GENERAL POSITION!)

$aef$: $+$
$ehj$: $-$

# Orientation predicate and order type




- How many convex hull vertices ($h$)
- How many triangulations?

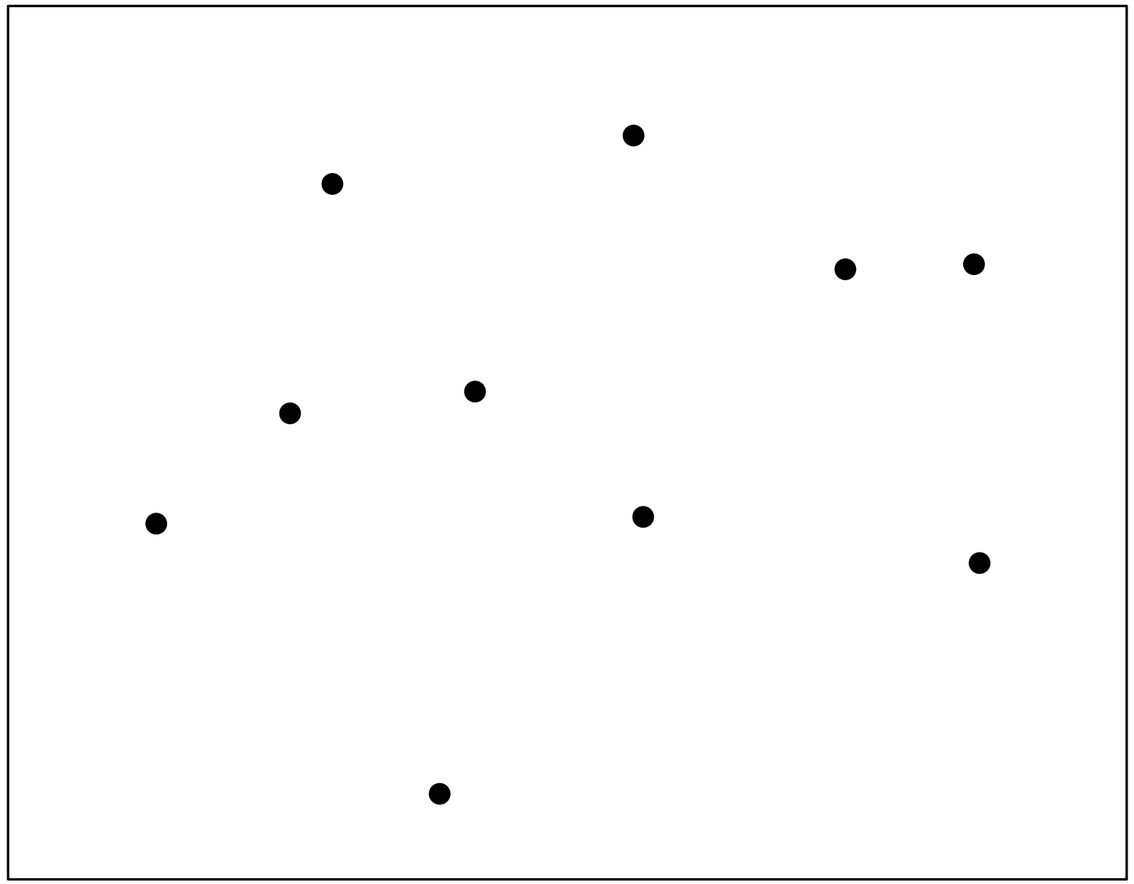
(GENERAL POSITION!)

$aef$: $+$
$ehj$: $-$
$abc$: $+$
$abd$: $-$
$\vdots$
$hij$: $-$

} the *order type* (combinatorial structure)
(chirotope / oriented matroid / rotation system)

- How many triangulations can a set of $n = 10$ points have? (at most? at least?)

# Orientation predicate and order type



- How many convex hull vertices ($h$)
- How many triangulations?

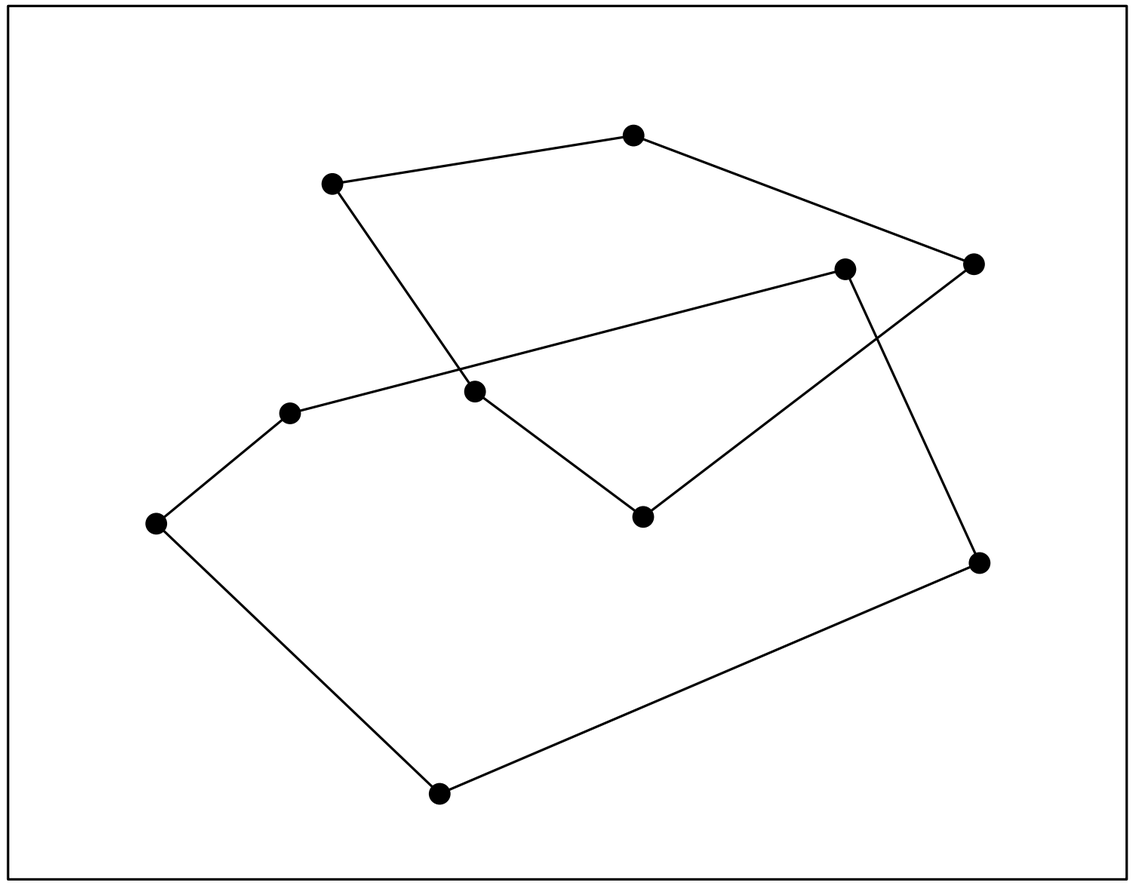
(GENERAL POSITION!)

$aef$: $+$
$ehj$: $-$
$abc$: $+$
$abd$: $-$
$\vdots$
$hij$: $-$

} the *order type* (combinatorial structure)
(chirotope / oriented matroid / rotation system)

- How many triangulations can a set of $n = 10$ points have? (at most? at least?)
- Can every set of 10 points be decomposed into two intersecting convex 5-gons?

# Orientation predicate and order type



- How many convex hull vertices ($h$)
- How many triangulations?

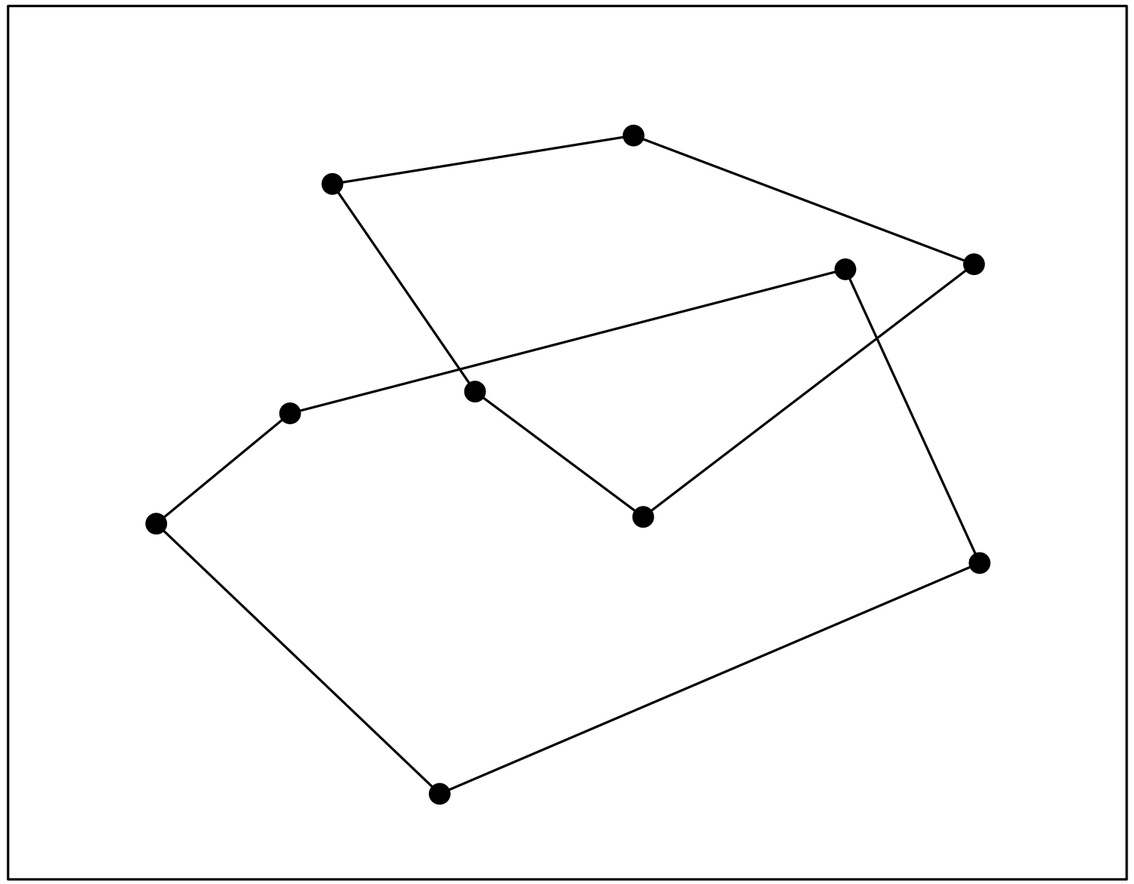
(GENERAL POSITION!)

$aef$: $+$
$ehj$: $-$
$abc$: $+$
$abd$: $-$
$\vdots$
$hij$: $-$

} the *order type* (combinatorial structure)
(chirotope / oriented matroid / rotation system)

- How many triangulations can a set of $n = 10$ points have? (at most? at least?)
- Can every set of 10 points be decomposed into two intersecting convex 5-gons?

# Orientation predicate and order type



- How many convex hull vertices ($h$)
- How many triangulations?

(GENERAL POSITION!)

$aef$: $+$
$ehj$: $-$
$abc$: $+$
$abd$: $-$
$\vdots$
$hij$: $-$

} the *order type* (combinatorial structure)
(chirotope / oriented matroid / rotation system)

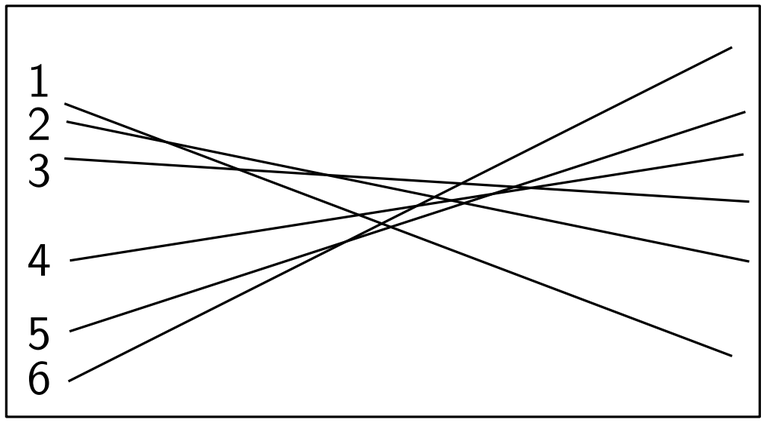
- How many triangulations can a set of $n = 10$ points have? (at most? at least?)
- Can every set of 10 points be decomposed into two intersecting convex 5-gons?

The order-type database!
[Aichholzer, Aurenhammer, and Krasser 2000–2006, 21st EuroCG]

$14,309,547$ OTs with 10 points
$2,334,512,907$ OTs with 11 points

# Duality between lines and points

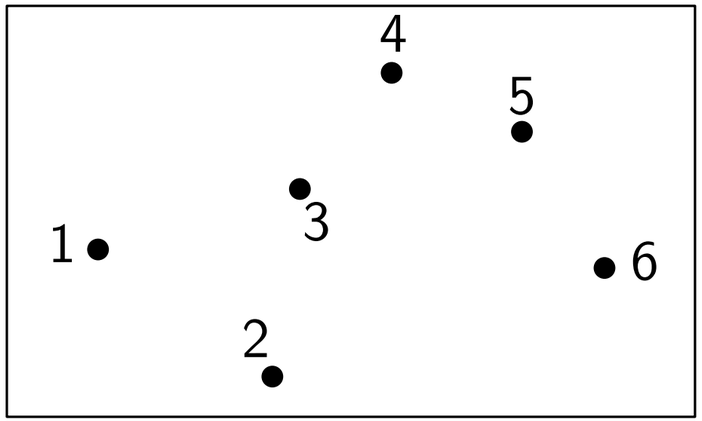
line arrangement




point-line duality

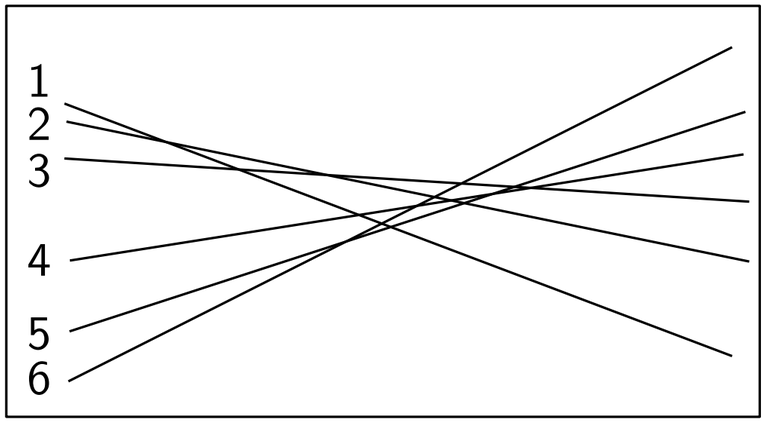
$y = ax - b$ $\qquad$ $(a, b)$

OT (point configuration)



# Duality between lines and points

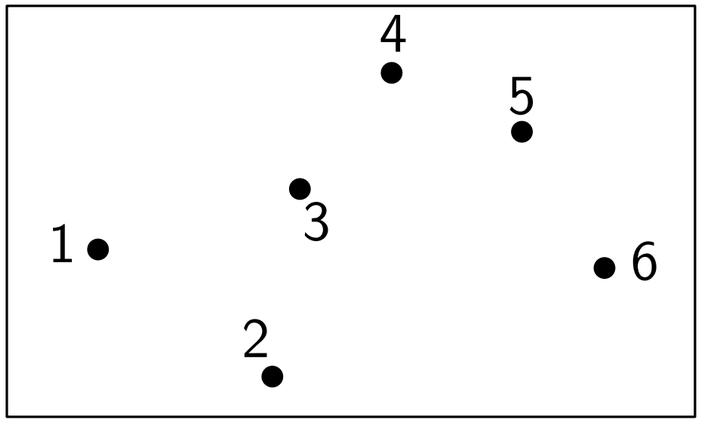
line arrangement



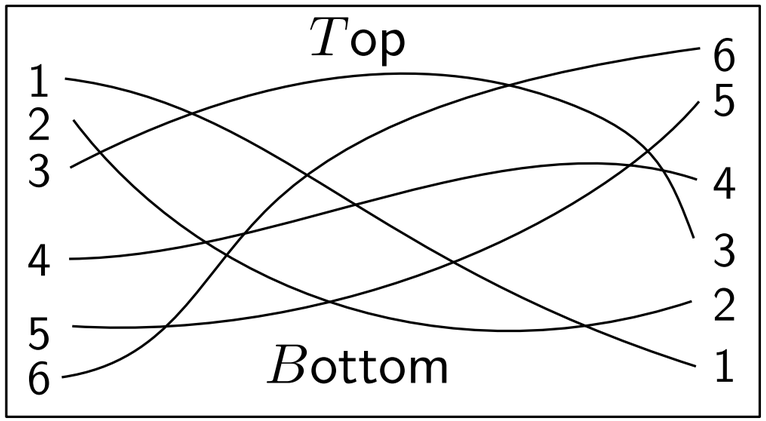

point-line duality

$y = ax - b$ $(a, b)$

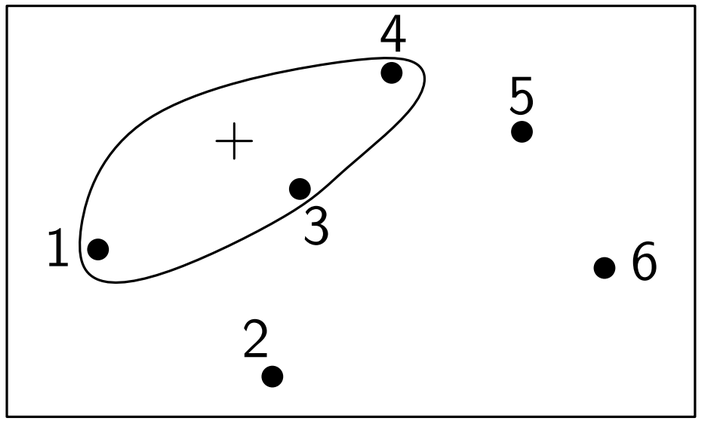
OT (point configuration)




pseudoline arrangement (PSLA)

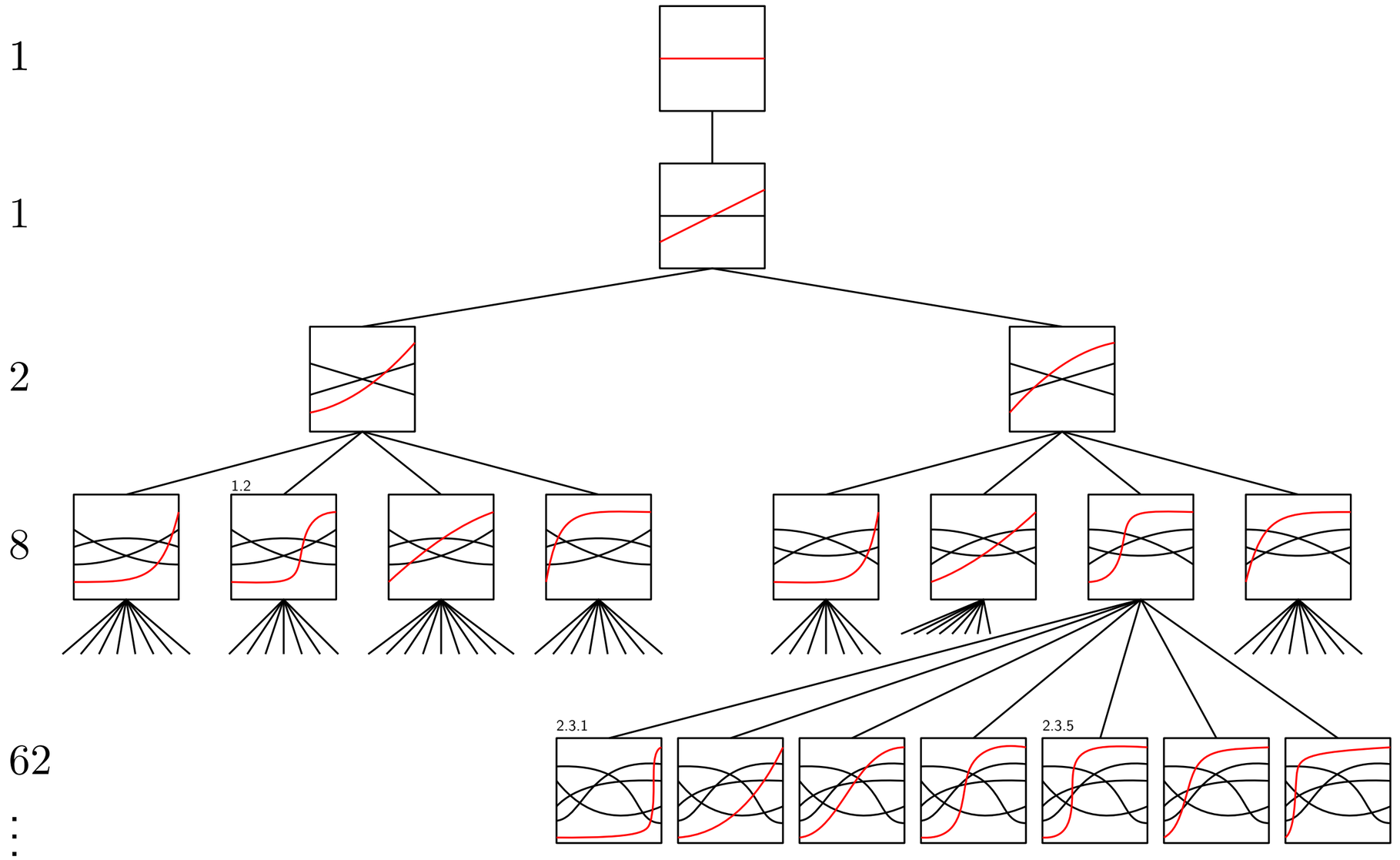



abstract order type (AOT)

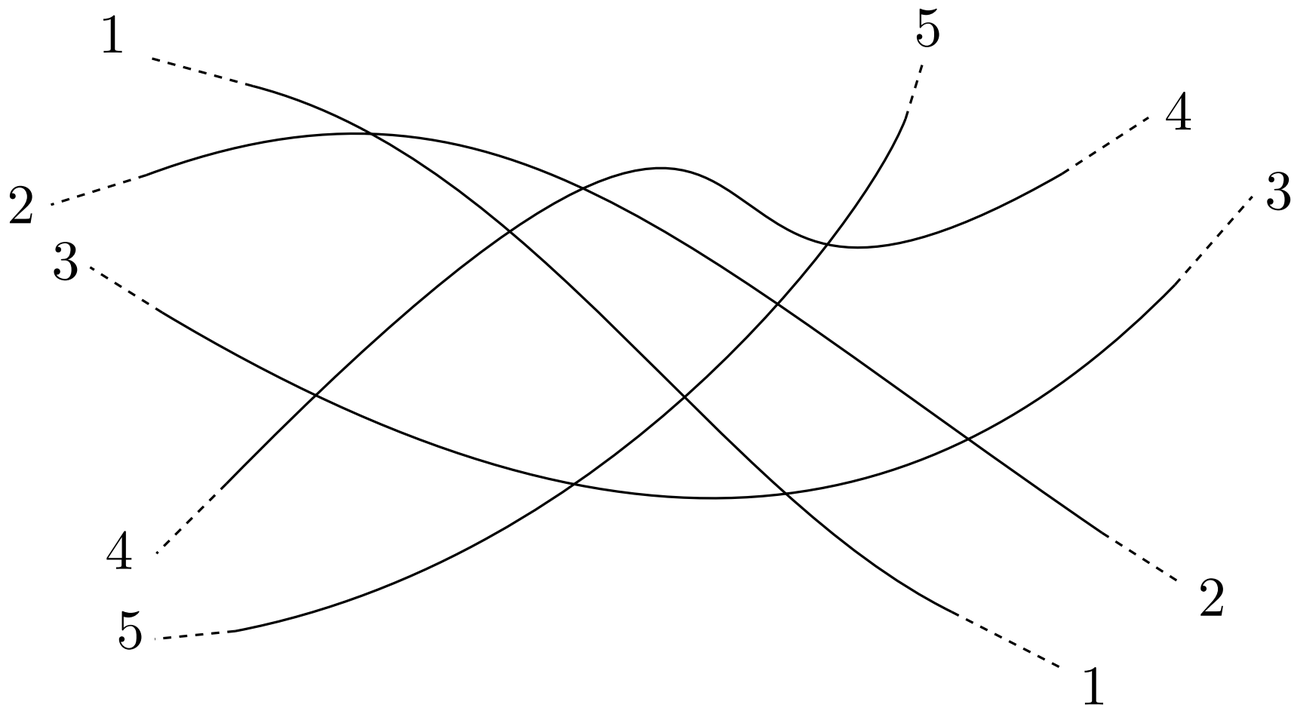



$x$-monotone PSLA

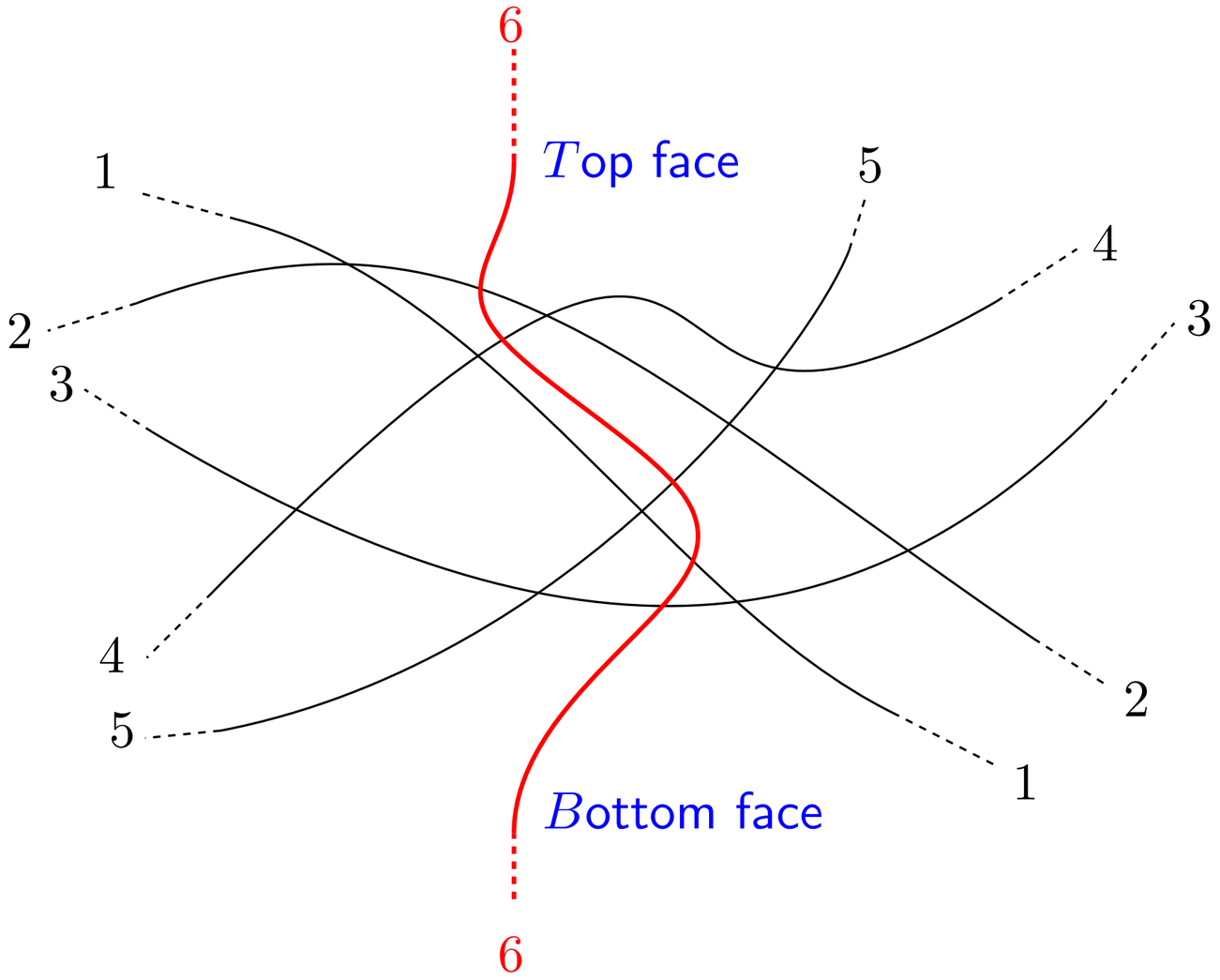
# Inductive enumeration of PSLAs



# Extending a PSLA

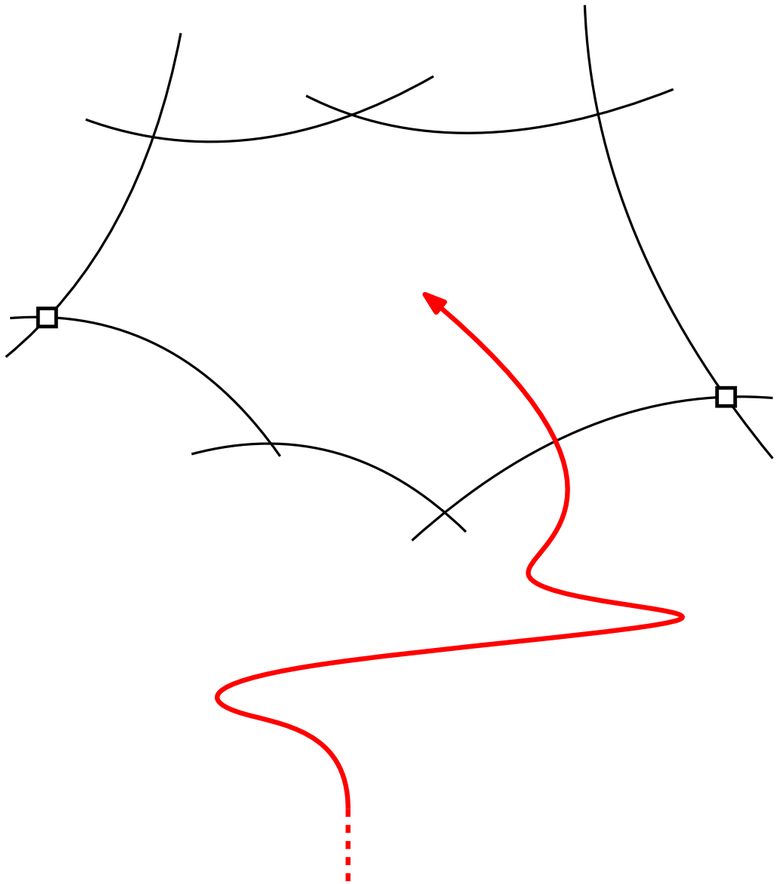


# Extending a PSLA



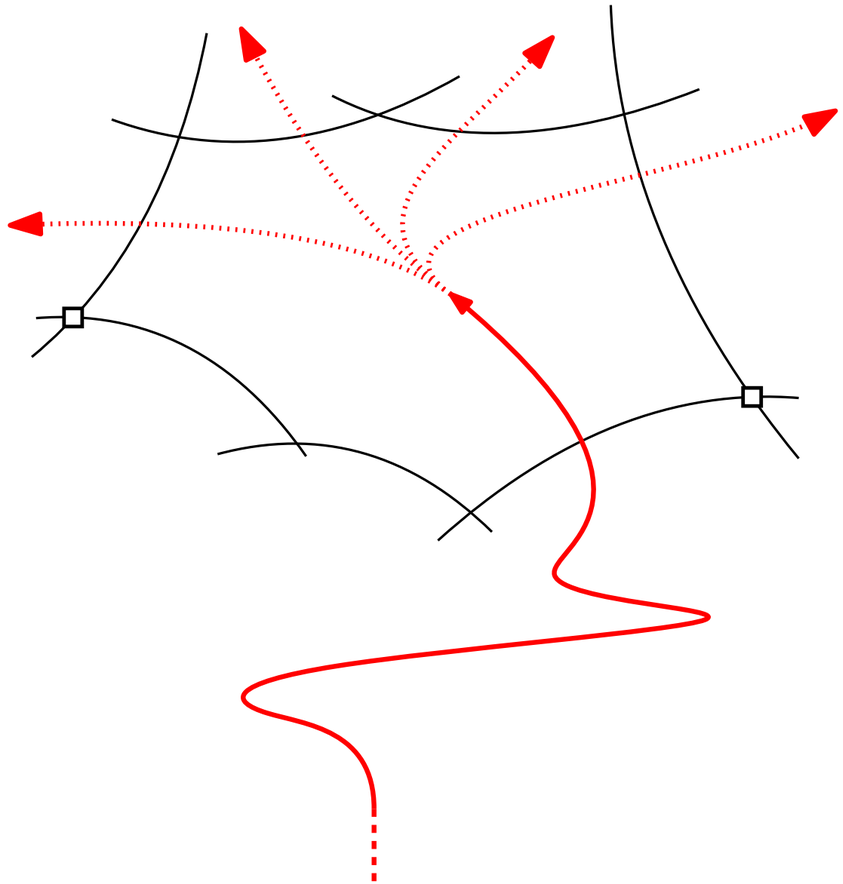
# Extending a PSLA

Generation (enumeration) of paths is straightforward: Recursive procedure with no dead ends!



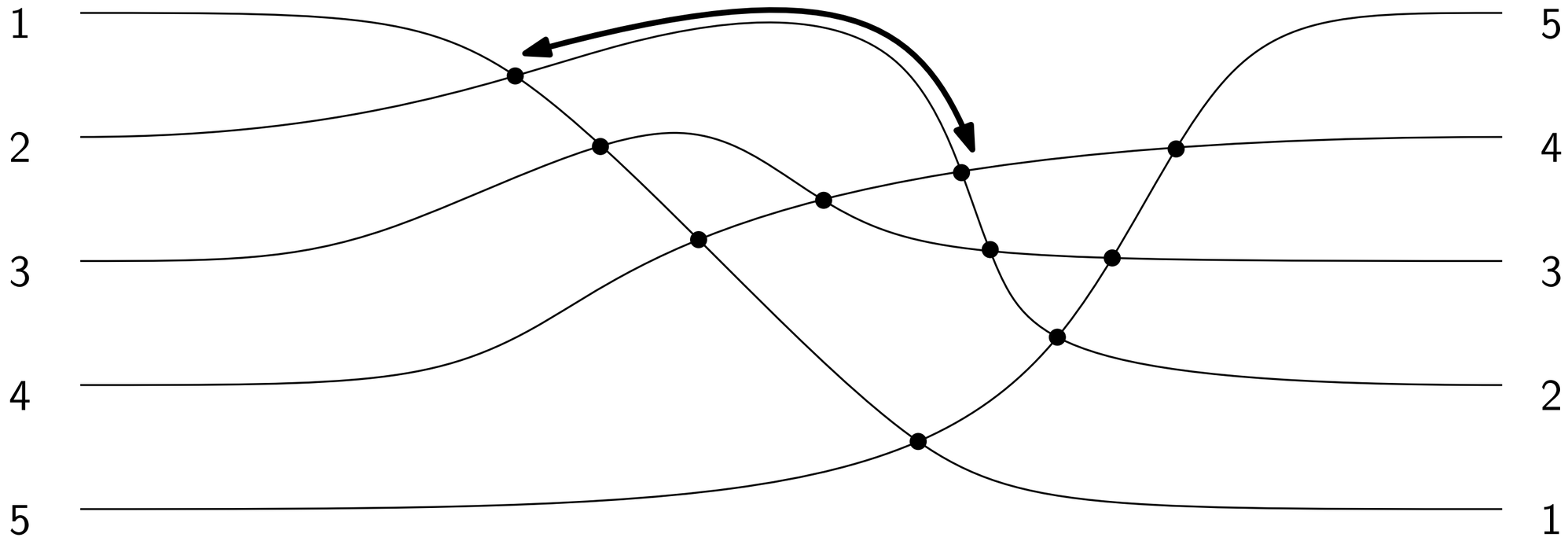
# Extending a PSLA

Generation (enumeration) of paths is straightforward: Recursive procedure with no dead ends!

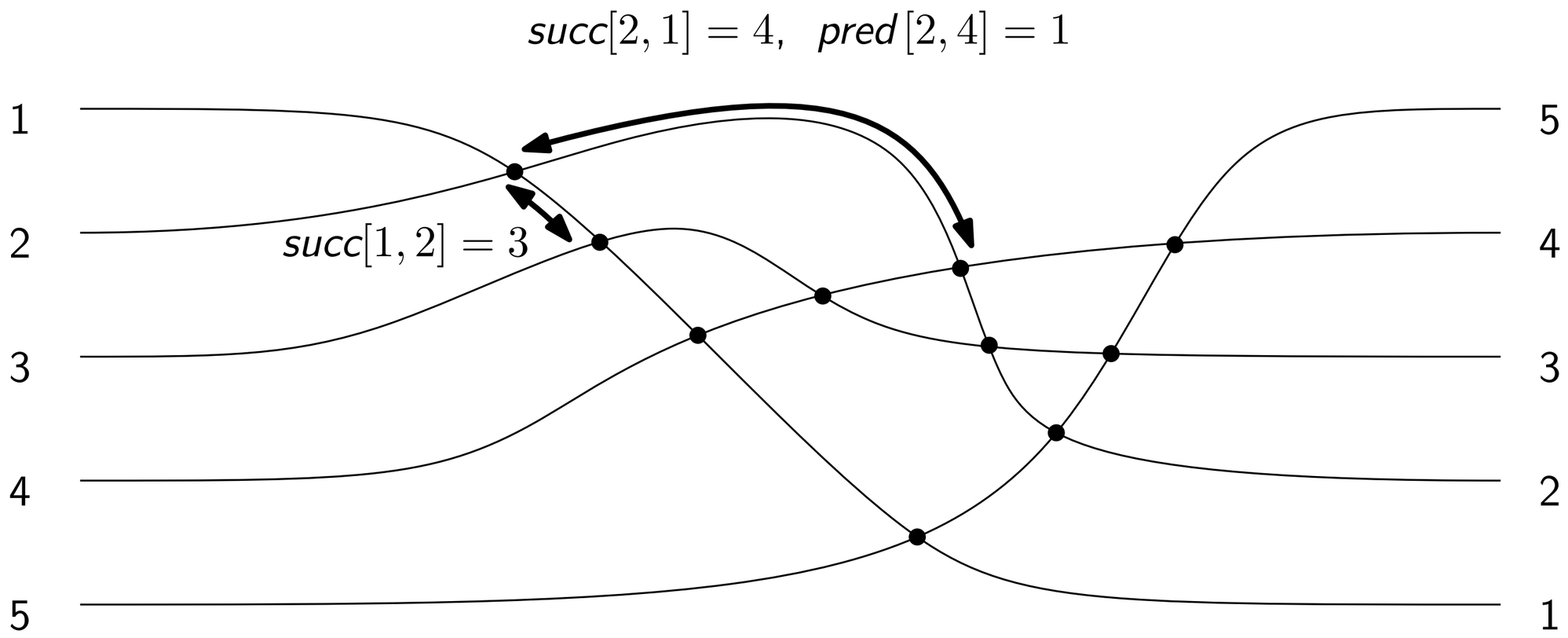


# Plane graph data structure for PSLAs

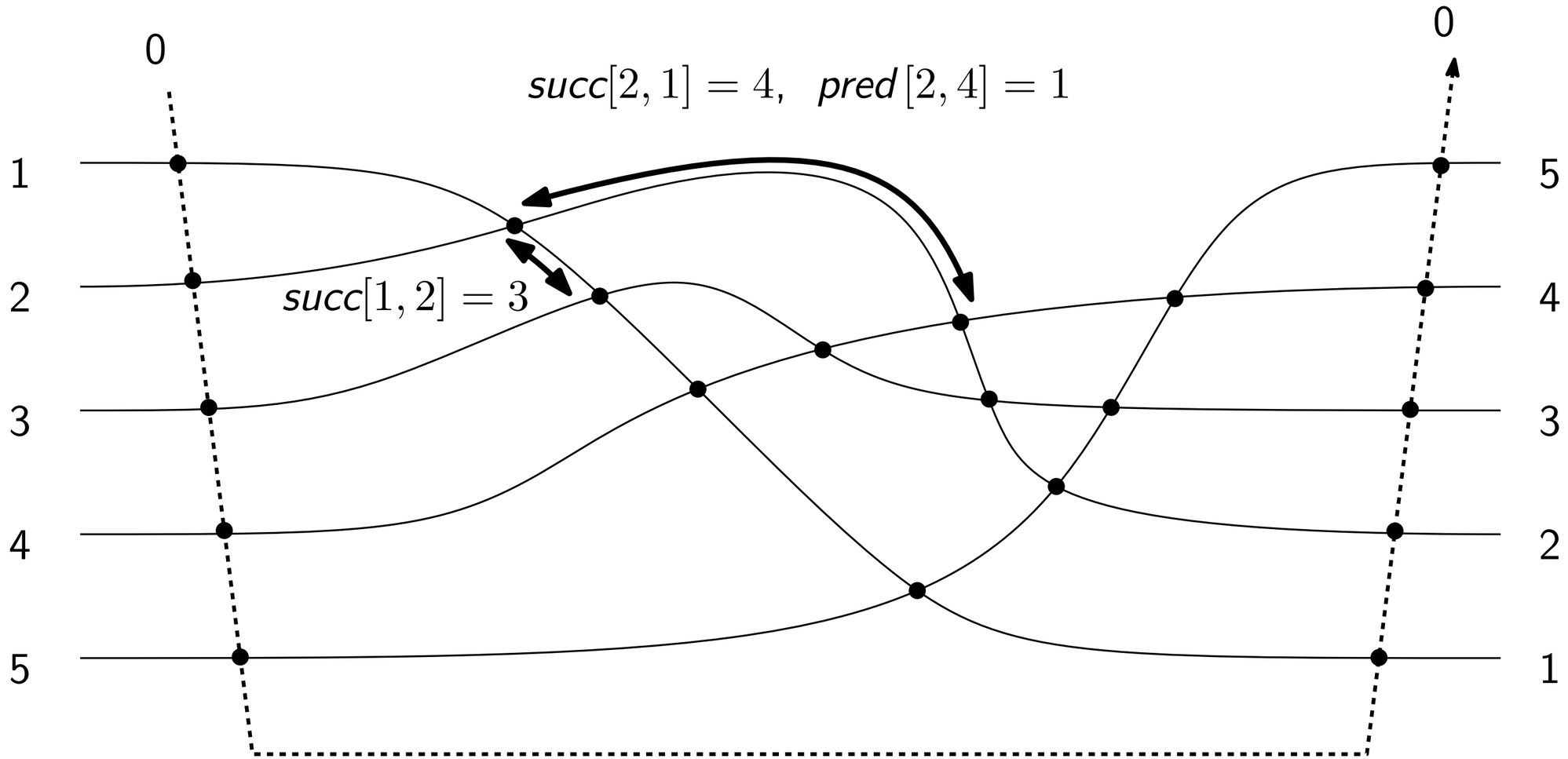
$$succ[2,1] = 4, \quad pred\,[2,4] = 1$$



# Plane graph data structure for PSLAs

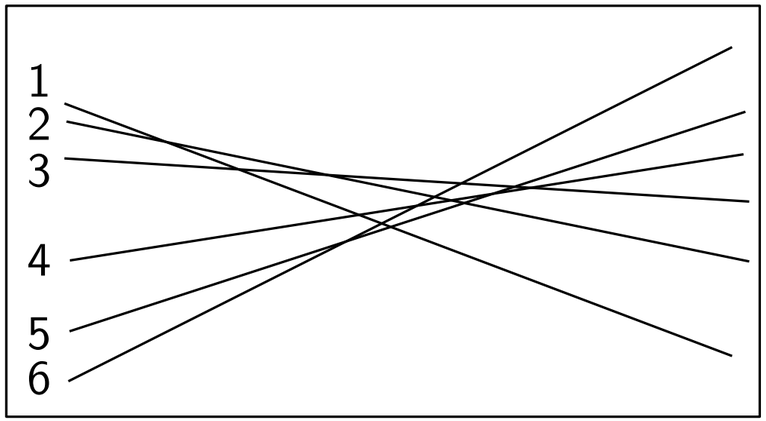


# Plane graph data structure for PSLAs



# Duality between lines and points

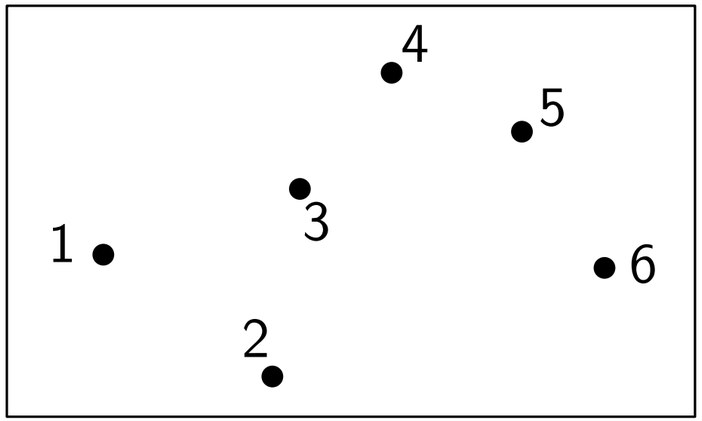
line arrangement



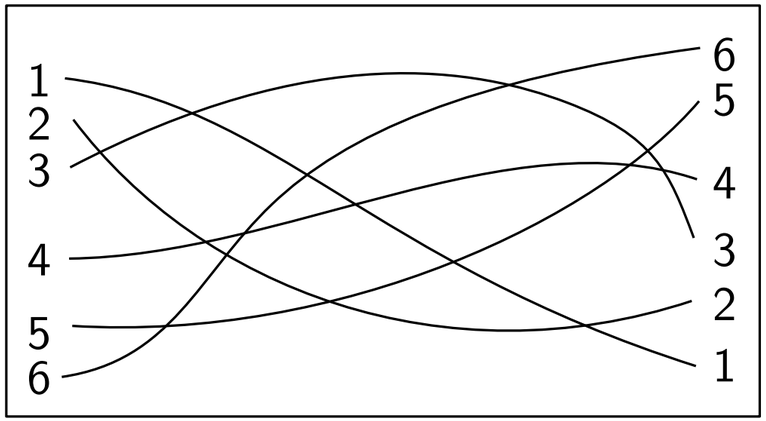

point-line duality

$y = ax - b$ $\qquad$ $(a, b)$

OT



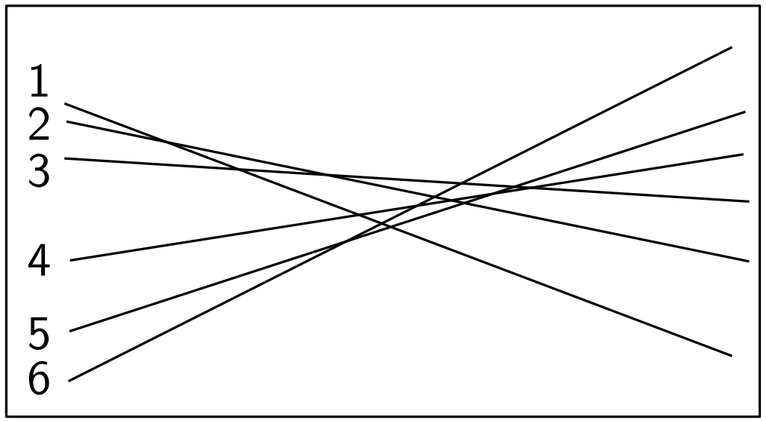

pseudoline arrangement (PSLA)




$n$ (pseudo)lines ordered by *slope* $a$ $\qquad$ $n$ points ordered by $x$-*coordinate* $a$

# Duality between lines and points

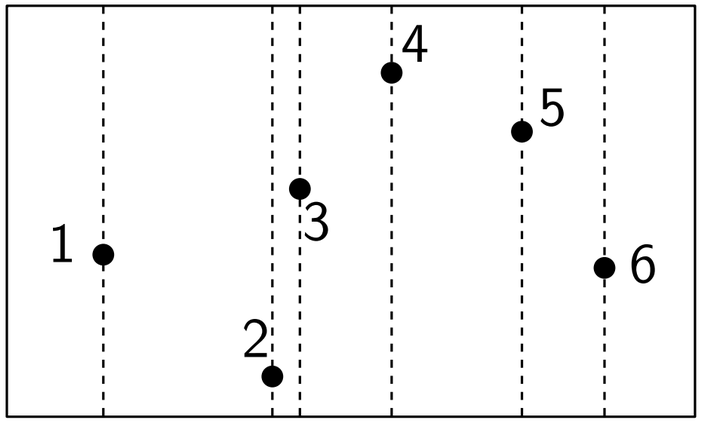
line arrangement



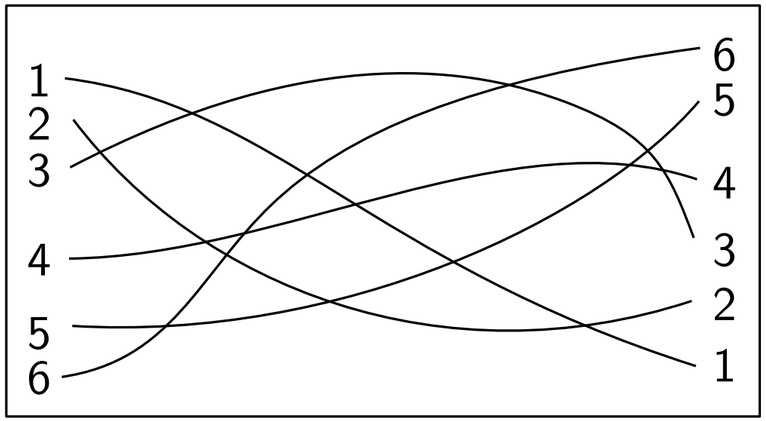

point-line duality

$y = ax - b$ $\qquad$ $(a, b)$

OT




pseudoline arrangement (PSLA)





extra point “0” at $-\infty$.
$\rightarrow n + 1$ points with a *marked* convex hull point.

$n$ (pseudo)lines ordered by *slope* $a$ $\qquad$ $n$ points ordered by $x$-*coordinate* $a$

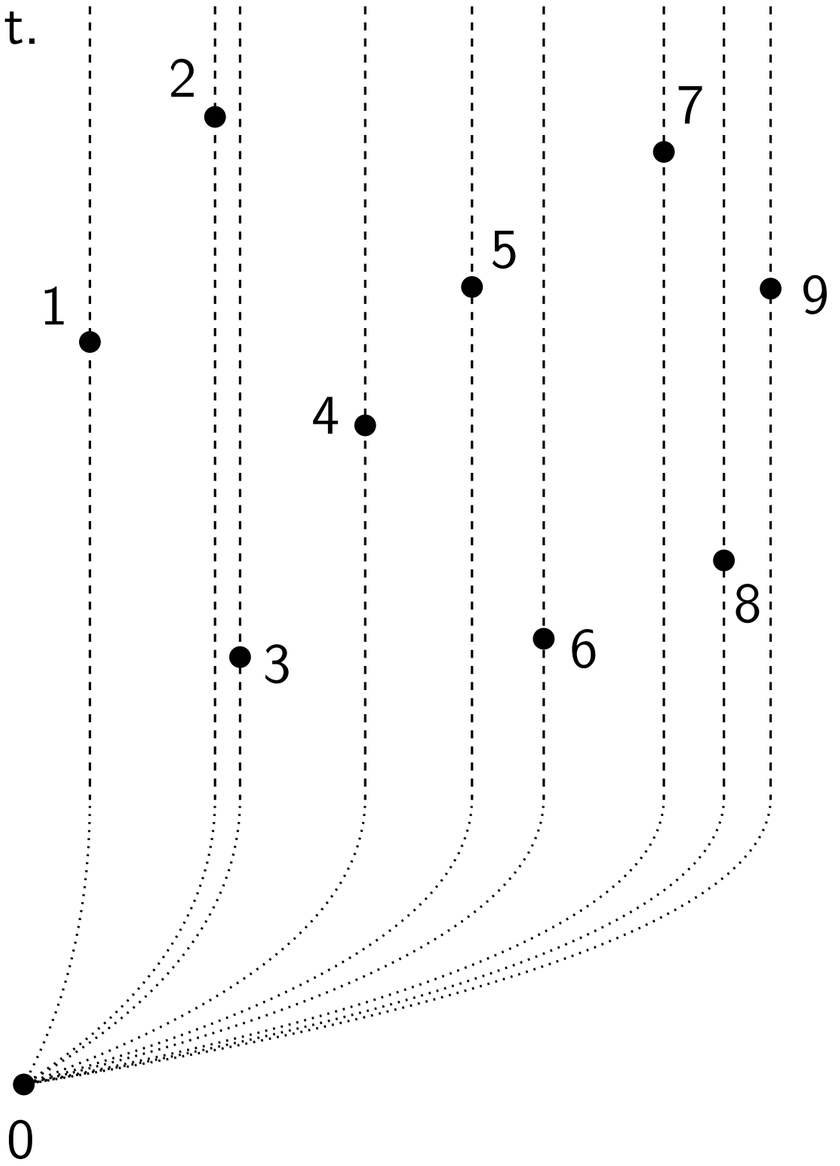
# From AOT to PSLA

PSLA with $n$ lines $\leftrightarrow$ $n+1$ points with a *marked* convex hull point.

Conversion from AOT $x$ to PSLA $y$:

- Start from an AOT $x$ with $n+1$ points
- Pick a hull vertex $v$ and an orientation ↻ or ↺
- Move $v$ to $-\infty$
- Dualize: $\rightarrow$ PSLA $y$ with $n$ pseudolines

$2h$ possibilities (not necessarily distinct)



# From AOT to PSLA

PSLA with $n$ lines $\leftrightarrow$ $n+1$ points with a *marked* convex hull point.

Conversion from AOT $x$ to PSLA $y$:

- Start from an AOT $x$ with $n+1$ points
- Pick a hull vertex $v$ and an orientation ↻ or ↺
- Move $v$ to $-\infty$
- Dualize: $\rightarrow$ PSLA $y$ with $n$ pseudolines

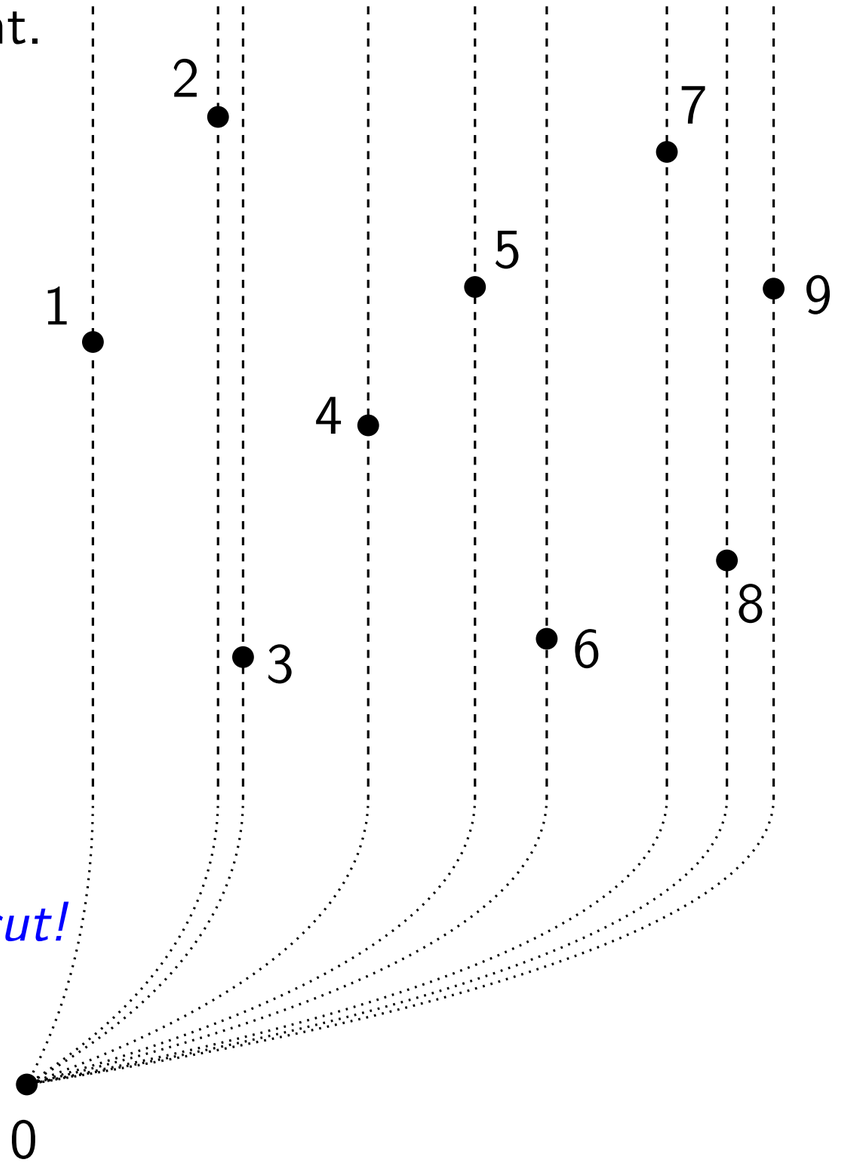
$2h$ possibilities (not necessarily distinct)

**Enumerating AOTs without duplication:**

For each PSLA $y$:
  compute the corresponding AOT $x$
  compute all PSLAs $y_1, y_2, \ldots, y_{2h}$ that correspond to $x$
  if the *representation* of $y$ is the lex-smallest among
    the representations of $y_1, y_2, \ldots, y_{2h}$:
  output $x$.

*shortcut!*

an $n \times n$ matrix



# The program

written in CWEB [Knuth and Levy]: *literate programming* in C

16 ¶⟨ Core subroutines for recursive generation 16 ⟩ ≡

```
void recursive_generate_PSLA_start(int n);

void recursive_generate_PSLA(int entering_edge, int k_right, int n)
{    /* The new line enters a face F from the bottom. The edge through which it crosses is part of line
        entering_edge, and its right endpoint is the crossing with k_right. */
  int j ← entering_edge;
  int j+ ← k_right;
  while (j+ > j) {    /* find right vertex of the current cell F */
    int j+_old ← j+;
    j+ ← SUCC(j+, j);
    j ← j+_old;
  }    /* the right vertex is the intersection of j and j+ */
  if (j+ = 0) {    /* F is unbounded */
```

# The program

written in CWEB [Knuth and Levy]: *literate programming* in C

16 ¶⟨ Core subroutines for recursive generation 16 ⟩ ≡
**void** *recursive_generate_PSLA_start*(**int** *n*);
**void** *recursive_generate_PSLA*(**int** *entering_edge*, **int** $k_{\text{right}}$, **int** *n*)

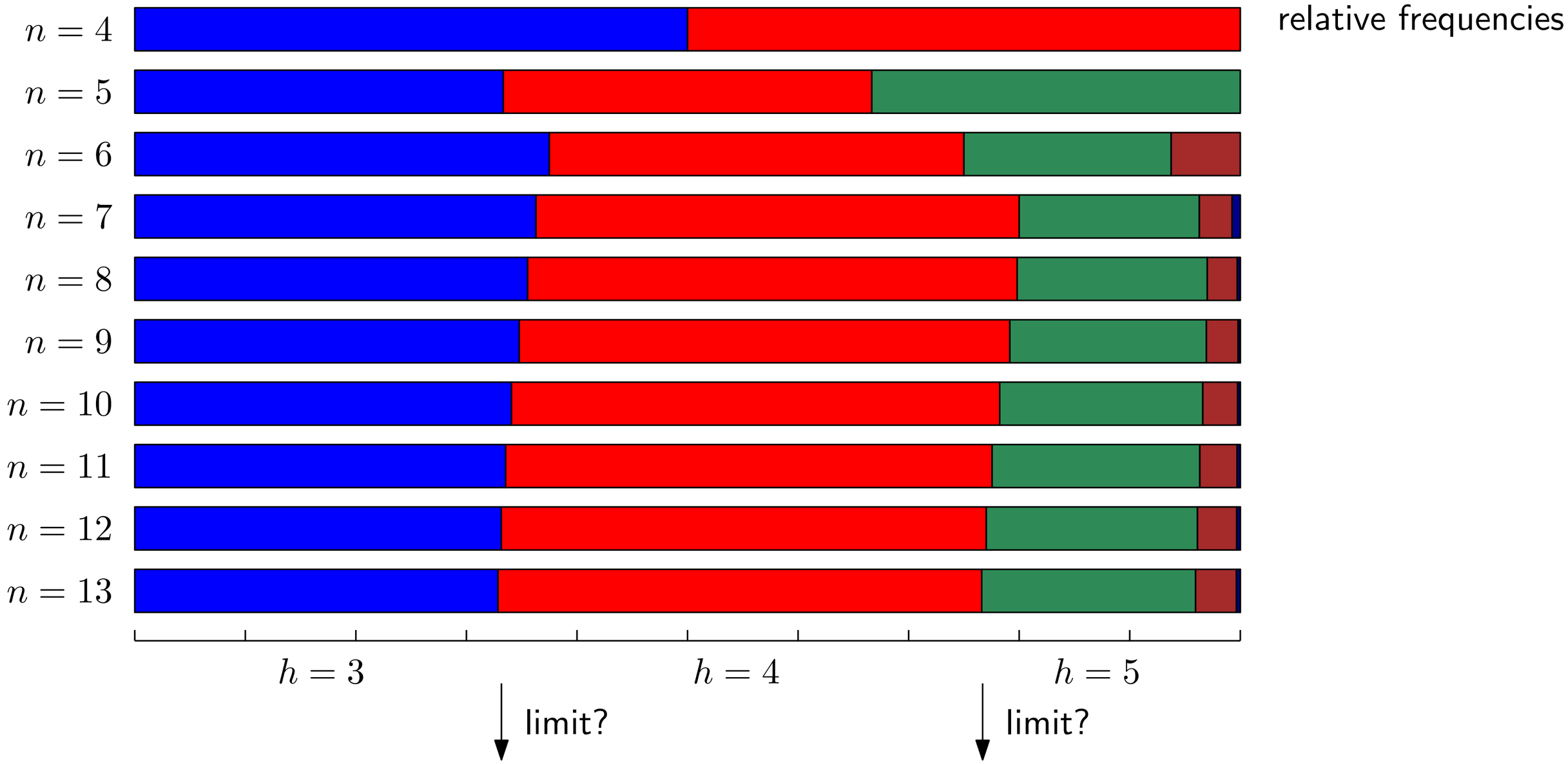
rough which it crosses is part of line
. */

## 4.1 Handling of a PSLA

19 ⟨ Process the PSLA; **return** if excluded 19 ⟩ ≡
⟨ Indicate Progress 20 ⟩;
**boolean** *is_excluded* ← *false*;
⟨ Check for exclusion and set the flag *is_excluded* 22 ⟩
**if** (*is_excluded*) **return**;
⟨ Gather statistics about the AOT, collect output 52 ⟩
⟨ Further processing of the AOT 58 ⟩
This code is used in chunk 16.

define your own job for processing the AOT

# Results: Number of hull vertices



# Numbers and runtimes

| $n$ | [OEIS A006247] #AOT | [A063666] #OT | $\Delta =$ #nonr. AOT | $\frac{\Delta}{\#\text{AOT}}$ | [OEIS A006245] #PSLA |
|---|---|---|---|---|---|
| 3 | 1 | 1 | 0 | 0 | 2 |
| 4 | 2 | 2 | 0 | 0 | 8 |
| 5 | 3 | 3 | 0 | 0 | 62 |
| 6 | 16 | 16 | 0 | 0 | 908 |
| 7 | 135 | 135 | 0 | 0 | 24,698 |
| 8 | 3,315 | 3,315 | 0 | 0 | 1,232,944 |
| 9 | 158,830 | 158,817 | 13 | 0,01 % | 112,018,190 |
| 10 | 14,320,182 | 14,309,547 | 10,635 | 0,07 % | 18,410,581,880 |
| 11 | 2,343,203,071 | 2,334,512,907 | 8,690,164 | 0,37 % | 5,449,192,389,984 |
| 12 | 691,470,685,682 | | | | 2,894,710,651,370,536 |
| 13 | 366,477,801,792,538 | | | | 2,752,596,959,306,389,652 |

the 2,334,512,907 order types of 11 points in 100 GBytes

[ The order-type database, Aichholzer–Aurenhammer–Krasser (2000–2007) ]

the 2,343,203,071 abstract order types of 11 points in 30 CPU minutes

the 691,470,685,682 abstract order types of 12 points in 200 CPU hours

the 366,477,801,792,538 abstract order types of 13 points in 3200 CPU days

# Stretchability, AOTs versus OTs

- Is an AOT an OT?
- Equivalently: Is a PSLA *stretchable*?

This is ETR-complete. [Mnëv 1985]

# The messy relation between all sorts of objects