

# Intersecții

Mihai-Sorin Stupariu

Sem. I, 2017-2018

# Motivație (Probleme geometrice în context 2D)

- **Problema 1.** Dată o listă (mulțime ordonată) de puncte  $\mathcal{P} = (P_1, P_2, \dots, P_m)$ , să se stabilească dacă ea reprezintă un poligon (linie poligonală fără autointersecții).

# Motivație (Probleme geometrice în context 2D)

- ▶ **Problema 1.** Dată o listă (mulțime ordonată) de puncte  $\mathcal{P} = (P_1, P_2, \dots, P_m)$ , să se stabilească dacă ea reprezintă un poligon (linie poligonală fără autointersecții).
- ▶ **Problema 2.** Date două poligoane  $\mathcal{P}$  și  $\mathcal{Q}$ , să se stabilească dacă se intersectează (interioarele lor se intersectează).

# Motivație (Probleme geometrice în context 2D)

- ▶ **Problema 1.** Dată o listă (mulțime ordonată) de puncte  $\mathcal{P} = (P_1, P_2, \dots, P_m)$ , să se stabilească dacă ea reprezintă un poligon (linie poligonală fără autointersecții).
- ▶ **Problema 2.** Date două poligoane  $\mathcal{P}$  și  $\mathcal{Q}$ , să se stabilească dacă se intersectează (interioarele lor se intersectează).
- ▶ **Problema 3.** Dată o mulțime  $\mathcal{S} = \{s_1, \dots, s_n\}$  de  $n$  segmente închise din plan, să se determine toate perechile care se intersectează.

# Motivație (Probleme geometrice în context 2D)

- ▶ **Problema 1.** Dată o listă (mulțime ordonată) de puncte  $\mathcal{P} = (P_1, P_2, \dots, P_m)$ , să se stabilească dacă ea reprezintă un poligon (linie poligonală fără autointersecții).
- ▶ **Problema 2.** Date două poligoane  $\mathcal{P}$  și  $\mathcal{Q}$ , să se stabilească dacă se intersectează (interioarele lor se intersectează).
- ▶ **Problema 3.** Dată o mulțime  $\mathcal{S} = \{s_1, \dots, s_n\}$  de  $n$  segmente închise din plan, să se determine toate perechile care se intersectează.
- ▶ **Problema 3'.** Dată o mulțime  $\mathcal{S} = \{s_1, \dots, s_n\}$  de  $n$  segmente închise din plan, să se determine toate punctele de intersecție dintre ele.

# Algoritmul trivial

- ▶ **Idee de lucru:** Sunt considerate toate perechile de segmente și se determină cele care se intersectează / se calculează punctele de intersecție.

# Algoritmul trivial

- ▶ **Idee de lucru:** Sunt considerate toate perechile de segmente și se determină cele care se intersectează / se calculează punctele de intersecție.
- ▶ **Complexitate:**

# Algoritmul trivial

- ▶ **Idee de lucru:** Sunt considerate toate perechile de segmente și se determină cele care se intersectează / se calculează punctele de intersecție.
- ▶ **Complexitate:**
  - ▶ timp:  $O(n^2)$



# Algoritmul trivial

- ▶ **Idee de lucru:** Sunt considerate toate perechile de segmente și se determină cele care se intersectează / se calculează punctele de intersecție.
- ▶ **Complexitate:**
  - ▶ timp:  $O(n^2)$
  - ▶ memorie:  $O(n)$

# Algoritmul trivial

- ▶ **Idee de lucru:** Sunt considerate toate perechile de segmente și se determină cele care se intersectează / se calculează punctele de intersecție.
- ▶ **Complexitate:**
  - ▶ timp:  $O(n^2)$
  - ▶ memorie:  $O(n)$
  - ▶ algebric: polinoame de gradul II (Problema 3), rapoarte de polinoame (Problema 3')

# Algoritmul trivial

- ▶ **Idee de lucru:** Sunt considerate toate perechile de segmente și se determină cele care se intersectează / se calculează punctele de intersecție.
- ▶ **Complexitate:**
  - ▶ timp:  $O(n^2)$
  - ▶ memorie:  $O(n)$
  - ▶ algebric: polinoame de gradul II (Problema 3), rapoarte de polinoame (Problema 3')
- ▶ **Comentariu:** În anumite cazuri: optim (dacă toate segmentele se intersectează).

# Algoritmul trivial

- ▶ **Idee de lucru:** Sunt considerate toate perechile de segmente și se determină cele care se intersectează / se calculează punctele de intersecție.
- ▶ **Complexitate:**
  - ▶ timp:  $O(n^2)$
  - ▶ memorie:  $O(n)$
  - ▶ algebric: polinoame de gradul II (Problema 3), rapoarte de polinoame (Problema 3')
- ▶ **Comentariu:** În anumite cazuri: optim (dacă toate segmentele se intersectează).
- ▶ Algoritmi mai eficienți (*output / intersection sensitive*)?