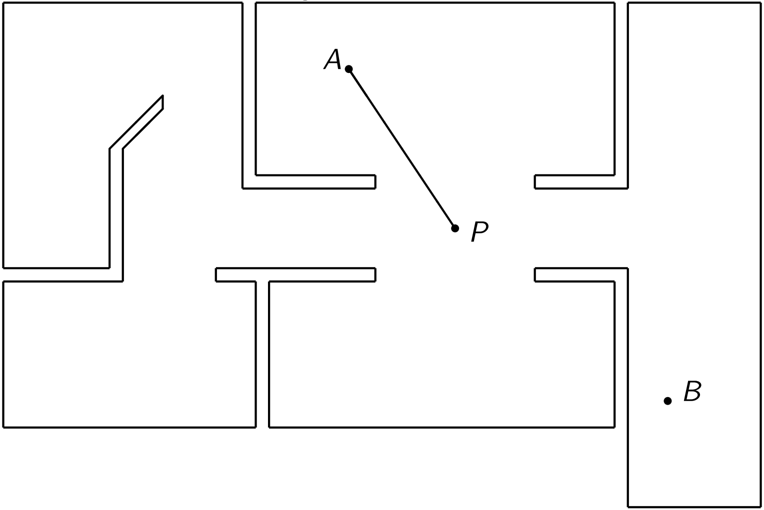
Triangulări

Mihai-Sorin Stupariu

Sem. I, 2019 - 2020

Supravegherea unei galerii de artă

Camera din P poate supraveghea A, dar nu B.



Formalizare

- O galerie de artă poate fi interpretată (în contextul acestei probleme) ca un poligon simplu $\mathcal{P}$ (adică un poligon fără autointersecții) având n vârfuri.

Formalizare

- O galerie de artă poate fi interpretată (în contextul acestei probleme) ca un poligon simplu $\mathcal{P}$ (adică un poligon fără autointersecții) având n vârfuri.
- O cameră video (vizibilitate 360^0) poate fi identificată cu un punct din interiorul lui $\mathcal{P}$; ea poate supraveghea acele puncte cu care poate fi unită printr-un segment inclus în interiorul poligonului.

Formalizare

- O galerie de artă poate fi interpretată (în contextul acestei probleme) ca un poligon simplu $\mathcal{P}$ (adică un poligon fără autointersecții) având n vârfuri.
- O cameră video (vizibilitate 360^0) poate fi identificată cu un punct din interiorul lui $\mathcal{P}$; ea poate supraveghea acele puncte cu care poate fi unită printr-un segment inclus în interiorul poligonului.
- **Problema galeriei de artă:** *câte camere video sunt necesare pentru a supraveghea o galerie de artă și unde trebuie amplasate acestea?*

Numărul de camere vs. forma poligonului

- Se dorește exprimarea numărului de camere necesare pentru supraveghere în funcție de n (sau controlarea acestuia de către n).

Numărul de camere vs. forma poligonului

- Se dorește exprimarea numărului de camere necesare pentru supraveghere în funcție de n (sau controlarea acestuia de către n).
- Pentru a supraveghea un spațiu având forma unui poligon convex, este suficientă o singură cameră.

Numărul de camere vs. forma poligonului

- Se dorește exprimarea numărului de camere necesare pentru supraveghere în funcție de n (sau controlarea acestuia de către n).
- Pentru a supraveghea un spațiu având forma unui poligon convex, este suficientă o singură cameră.
- Numărul de camere depinde și de forma poligonului: cu cât forma este mai "complexă", cu atât numărul de camere va fi mai mare.

Numărul de camere vs. forma poligonului

- Se dorește exprimarea numărului de camere necesare pentru supraveghere în funcție de n (sau controlarea acestuia de către n).
- Pentru a supraveghea un spațiu având forma unui poligon convex, este suficientă o singură cameră.
- Numărul de camere depinde și de forma poligonului: cu cât forma este mai "complexă", cu atât numărul de camere va fi mai mare.
- **Principiu:** Poligonul considerat: descompus în triunghiuri (triangulare).

Definiții

- Fie $\mathcal{P}$ un poligon plan.

Definiții

- Fie $\mathcal{P}$ un poligon plan.
- (i) O **diagonală** a lui $\mathcal{P}$ este un segment ce unește două vârfuri ale acestuia și care este situat în interiorul lui $\mathcal{P}$.

Definiții

- Fie $\mathcal{P}$ un poligon plan.
- (i) O **diagonală** a lui $\mathcal{P}$ este un segment ce unește două vârfuri ale acestuia și care este situat în interiorul lui $\mathcal{P}$.
- (ii) O **triangulare** $\mathcal{T}_P$ a lui $\mathcal{P}$ este o descompunere a lui $\mathcal{P}$ în triunghiuri, dată de o mulțime maximală de diagonale ce nu se intersectează.

Definiții

- Fie $\mathcal{P}$ un poligon plan.
- (i) O **diagonală** a lui $\mathcal{P}$ este un segment ce unește două vârfuri ale acestuia și care este situat în interiorul lui $\mathcal{P}$.
- (ii) O **triangulare** $\mathcal{T}_P$ a lui $\mathcal{P}$ este o descompunere a lui $\mathcal{P}$ în triunghiuri, dată de o mulțime maximală de diagonale ce nu se intersectează.
- **Teoremă.** *Orice poligon simplu admite o triangulare. Orice triangulare a unui poligon cu n vârfuri conține exact $n-2$ triunghiuri.*

Rezovlarea problemei galeriei de artă

- Amplasarea camerelor se poate face în vârfurile poligonului.

Rezovlarea problemei galeriei de artă

- Amplasarea camerelor se poate face în vârfurile poligonului.
- Dată o pereche $(\mathcal{P}, \mathcal{T}_P)$ se consideră o 3-colorare a acesteia: fiecărui vârf îi corespunde o culoare dintr-un set de 3 culori și pentru fiecare triunghi, cele 3 vârfuri au culori distincte.

Rezovlarea problemei galeriei de artă

- Amplasarea camerelor se poate face în vârfurile poligonului.
- Dată o pereche $(\mathcal{P}, \mathcal{T}_P)$ se consideră o 3-colorare a acesteia: fiecărui vârf îi corespunde o culoare dintr-un set de 3 culori și pentru fiecare triunghi, cele 3 vârfuri au culori distincte.
- **Observație.** Dacă $\mathcal{P}$ este simplu, o astfel de colorare există, deoarece graful asociat perechii $(\mathcal{P}, \mathcal{T}_P)$ este arbore.

Teorema galeriei de artă

- **Teoremă.** [Chvátal, 1975; Fisk, 1978] *Pentru un poligon cu n vârfuri,* $\left[\frac{n}{3}\right]$ *camere sunt* **uneori necesare** *și întotdeauna* **suficiente** *pentru ca fiecare punct al poligonului să fie vizibil din cel puțin una din camere.*

Teorema galeriei de artă

- **Teoremă.** [Chvátal, 1975; Fisk, 1978] *Pentru un poligon cu n vârfuri,* $\left[\frac{n}{3}\right]$ *camere sunt* **uneori necesare** *și întotdeauna* **suficiente** *pentru ca fiecare punct al poligonului să fie vizibil din cel puțin una din camere.*
- Despre Teorema Galeriei de Artă: J. O'Rourke, *Art Gallery Theorems and Algorithms*

Metode de triangulare: ear cutting / clipping / trimming

- Concepte pentru un poligon $\mathcal{P} = (P_1, P_2, \ldots, P_n)$:

Metode de triangulare: ear cutting / clipping / trimming

- Concepte pentru un poligon $\mathcal{P} = (P_1, P_2, \ldots, P_n)$:
  - **Vârf convex/concav("reflex"):** se stabilește cu testul de orientare.
  - **Vârf principal:** P_i este principal dacă $[P_{i-1}P_{i+1}]$ este diagonală (echivalent: nu există un alt vârf în interiorul sau pe laturile $\Delta P_{i-1}P_iP_{i+1}$).

Metode de triangulare: ear cutting / clipping / trimming

- Concepte pentru un poligon $\mathcal{P} = (P_1, P_2, \ldots, P_n)$:
  - **Vârf convex/concav("reflex"):** se stabilește cu testul de orientare.
  - **Vârf principal:** P_i este principal dacă $[P_{i-1}P_{i+1}]$ este diagonală (echivalent: nu există un alt vârf în interiorul sau pe laturile $\Delta P_{i-1}P_iP_{i+1}$).
  - **Ear (vârf / componentă de tip E):** este un vârf principal convex [Meisters, 1975]. Dacă P_i este componentă de tip E, atunci segmentul $[P_{i-1}P_{i+1}]$ nu intersectează laturile poligonului și este situat în interiorul acestuia, adică este "diagonală veritabilă", iar $\Delta P_{i-1}P_iP_{i+1}$ poate fi "eliminat".
  - **Mouth (vârf / componentă de tip M)**: este un vârf principal concav [Toussaint, 1991].

Metode de triangulare: ear cutting / clipping / trimming

- Concepte pentru un poligon $\mathcal{P} = (P_1, P_2, \ldots, P_n)$:
  - **Vârf convex/concav("reflex"):** se stabilește cu testul de orientare.
  - **Vârf principal:** P_i este principal dacă $[P_{i-1}P_{i+1}]$ este diagonală (echivalent: nu există un alt vârf în interiorul sau pe laturile $\Delta P_{i-1}P_iP_{i+1}$).
  - **Ear (vârf / componentă de tip E):** este un vârf principal convex [Meisters, 1975]. Dacă P_i este componentă de tip E, atunci segmentul $[P_{i-1}P_{i+1}]$ nu intersectează laturile poligonului și este situat în interiorul acestuia, adică este "diagonală veritabilă", iar $\Delta P_{i-1}P_iP_{i+1}$ poate fi "eliminat".
  - **Mouth (vârf / componentă de tip M)**: este un vârf principal concav [Toussaint, 1991].
- **Criterii de clasificare a vârfurilor:** (i) vârf convex/concav; (ii) vârf principal/nu.

Metode de triangulare: ear cutting / clipping / trimming

- Concepte pentru un poligon $\mathcal{P} = (P_1, P_2, \ldots, P_n)$:
  - **Vârf convex/concav("reflex"):** se stabilește cu testul de orientare.
  - **Vârf principal:** P_i este principal dacă $[P_{i-1}P_{i+1}]$ este diagonală (echivalent: nu există un alt vârf în interiorul sau pe laturile $\Delta P_{i-1}P_iP_{i+1}$).
  - **Ear (vârf / componentă de tip E):** este un vârf principal convex [Meisters, 1975]. <u>Dacă P_i este componentă de tip E, atunci segmentul $[P_{i-1}P_{i+1}]$ nu intersectează laturile poligonului și este situat în interiorul acestuia, adică este "diagonală veritabilă", iar $\Delta P_{i-1}P_iP_{i+1}$ poate fi "eliminat".</u>
  - **Mouth (vârf / componentă de tip M)**: este un vârf principal concav [Toussaint, 1991].
- **Criterii de clasificare a vârfurilor:** (i) vârf convex/concav; (ii) vârf principal/nu.
- **Teoremă.** (Two Ears Theorem [Meisters, 1975]) *Orice poligon cu cel puțin* 4 *vârfuri admite cel puțin două componente de tip E care nu se suprapun.*
- **Corolar.** *Orice poligon simplu admite (cel puțin) o diagonală.*

Metode de triangulare: ear cutting / clipping / trimming

- Concepte pentru un poligon $\mathcal{P} = (P_1, P_2, \ldots, P_n)$:
  - **Vârf convex/concav("reflex"):** se stabilește cu testul de orientare.
  - **Vârf principal:** P_i este principal dacă $[P_{i-1}P_{i+1}]$ este diagonală (echivalent: nu există un alt vârf în interiorul sau pe laturile $\Delta P_{i-1}P_iP_{i+1}$).
  - **Ear (vârf / componentă de tip E):** este un vârf principal convex [Meisters, 1975]. Dacă P_i este componentă de tip E, atunci segmentul $[P_{i-1}P_{i+1}]$ nu intersectează laturile poligonului și este situat în interiorul acestuia, adică este "diagonală veritabilă", iar $\Delta P_{i-1}P_iP_{i+1}$ poate fi "eliminat".
  - **Mouth (vârf / componentă de tip M)**: este un vârf principal concav [Toussaint, 1991].
- **Criterii de clasificare a vârfurilor:** (i) vârf convex/concav; (ii) vârf principal/nu.
- **Teoremă.** (Two Ears Theorem [Meisters, 1975]) *Orice poligon cu cel puțin* 4 *vârfuri admite cel puțin două componente de tip E care nu se suprapun.*
- **Corolar.** *Orice poligon simplu admite (cel puțin) o diagonală.*
- Găsirea unei componente de tip E: complexitate $O(n)$ [ElGindy, Everett, Toussaint, 1993]. Se bazează pe Two Ears Theorem!
- Algoritmul de triangulare bazat de metoda *ear cutting*: complexitate $O(n^2)$.

Metode de triangulare: ear cutting / clipping / trimming

- Concepte pentru un poligon $\mathcal{P} = (P_1, P_2, \ldots, P_n)$:
  - **Vârf convex/concav("reflex"):** se stabilește cu testul de orientare.
  - **Vârf principal:** P_i este principal dacă $[P_{i-1}P_{i+1}]$ este diagonală (echivalent: nu există un alt vârf în interiorul sau pe laturile $\Delta P_{i-1}P_iP_{i+1}$).
  - **Ear (vârf / componentă de tip E):** este un vârf principal convex [Meisters, 1975]. <u>Dacă P_i este componentă de tip E, atunci segmentul $[P_{i-1}P_{i+1}]$ nu intersectează laturile poligonului și este situat în interiorul acestuia, adică este "diagonală veritabilă", iar $\Delta P_{i-1}P_iP_{i+1}$ poate fi "eliminat".</u>
  - **Mouth (vârf / componentă de tip M)**: este un vârf principal concav [Toussaint, 1991].
- **Criterii de clasificare a vârfurilor:** (i) vârf convex/concav; (ii) vârf principal/nu.
- **Teoremă.** (Two Ears Theorem [Meisters, 1975]) *Orice poligon cu cel puțin* 4 *vârfuri admite cel puțin două componente de tip E care nu se suprapun.*
- **Corolar.** *Orice poligon simplu admite (cel puțin) o diagonală.*
- Găsirea unei componente de tip E: complexitate $O(n)$ [ElGindy, Everett, Toussaint, 1993]. Se bazează pe Two Ears Theorem!
- Algoritmul de triangulare bazat de metoda *ear cutting*: complexitate $O(n^2)$.
- Link despre triangulări
  Link pentru algoritmul *Ear cutting*

Metode de triangulare: descompunerea în poligoane monotone

- Concept: **poligon** y**-monoton**
- Algoritmi de triangulare eficienți: complexitate $O(n)$ pentru poligoane y-monotone [Garey et al., 1978] (algoritmul este descris pe slide-urile următoare).

Metode de triangulare: descompunerea în poligoane monotone

- Concept: **poligon y-monoton**
- Algoritmi de triangulare eficienți: complexitate $O(n)$ pentru poligoane y-monotone [Garey et al., 1978] (algoritmul este descris pe slide-urile următoare).
- Descompunerea unui poligon oarecare in componente y-monotone poate fi realizată cu un algoritm de complexitate $O(n \log n)$ [Lee, Preparata, 1977]. În concluzie, un poligon poate fi triangulat folosind un algoritm de complexitate $O(n \log n)$.

Metode de triangulare: descompunerea în poligoane monotone

- Concept: **poligon y-monoton**
- Algoritmi de triangulare eficienți: complexitate $O(n)$ pentru poligoane y-monotone [Garey et al., 1978] (algoritmul este descris pe slide-urile următoare).
- Descompunerea unui poligon oarecare in componente y-monotone poate fi realizată cu un algoritm de complexitate $O(n \log n)$ [Lee, Preparata, 1977]. În concluzie, un poligon poate fi triangulat folosind un algoritm de complexitate $O(n \log n)$.
- Există și alte clase de algoritmi mai rapizi; [Chazelle, 1990]: algoritm liniar.

Metode de triangulare: descompunerea în poligoane monotone

- Concept: **poligon y-monoton**
- Algoritmi de triangulare eficienți: complexitate $O(n)$ pentru poligoane y-monotone [Garey et al., 1978] (algoritmul este descris pe slide-urile următoare).
- Descompunerea unui poligon oarecare in componente y-monotone poate fi realizată cu un algoritm de complexitate $O(n \log n)$ [Lee, Preparata, 1977]. În concluzie, un poligon poate fi triangulat folosind un algoritm de complexitate $O(n \log n)$.
- Există și alte clase de algoritmi mai rapizi; [Chazelle, 1990]: algoritm liniar.
- Găsirea unui algoritm liniar "simplu" Problemă în *The Open Problems Project*

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$
4.   **do** **if** v_j și vârful din top al lui $\mathcal{S}$ sunt în lanțuri diferite

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$
4.   **do** **if** v_j și vârful din top al lui $\mathcal{S}$ sunt în lanțuri diferite
5.   **then** extrage toate vârfurile din $\mathcal{S}$

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$
4.   **do** **if** v_j și vârful din top al lui $\mathcal{S}$ sunt în lanțuri diferite
5.   **then** extrage toate vârfurile din $\mathcal{S}$
6.   inserează diagonale de la v_j la vf. extrase, exceptând ultimul

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$
4.   **do** **if** v_j și vârful din top al lui $\mathcal{S}$ sunt în lanțuri diferite
5.   **then** extrage toate vârfurile din $\mathcal{S}$
6.   inserează diagonale de la v_j la vf. extrase, exceptând ultimul
7.   inserează v_{j-1} și v_j în $\mathcal{S}$

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$
4.   **do** **if** v_j și vârful din top al lui $\mathcal{S}$ sunt în lanțuri diferite
5.   **then** extrage toate vârfurile din $\mathcal{S}$
6.   inserează diagonale de la v_j la vf. extrase, exceptând ultimul
7.   inserează v_{j-1} și v_j în $\mathcal{S}$
8.   **else** extrage un vârf din $\mathcal{S}$

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$
4. **do** **if** v_j și vârful din top al lui $\mathcal{S}$ sunt în lanțuri diferite
5. **then** extrage toate vârfurile din $\mathcal{S}$
6. inserează diagonale de la v_j la vf. extrase, exceptând ultimul
7. inserează v_{j-1} și v_j în $\mathcal{S}$
8. **else** extrage un vârf din $\mathcal{S}$
9. extrage celelalte vârfuri din $\mathcal{S}$ dacă diagonalele formate cu v_j sunt în interiorul lui $\mathcal{P}$; inserează aceste diagonale; inserează înapoi ultimul vârf extras

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \dots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$
4.   **do** **if** v_j și vârful din top al lui $\mathcal{S}$ sunt în lanțuri diferite
5.   **then** extrage toate vârfurile din $\mathcal{S}$
6.   inserează diagonale de la v_j la vf. extrase, exceptând ultimul
7.   inserează v_{j-1} și v_j în $\mathcal{S}$
8.   **else** extrage un vârf din $\mathcal{S}$
9.   extrage celelalte vârfuri din $\mathcal{S}$ dacă diagonalele formate cu v_j sunt în interiorul lui $\mathcal{P}$; inserează aceste diagonale; inserează înapoi ultimul vârf extras
10.   inserează v_j în $\mathcal{S}$

Triangularea poligoanelor monotone

Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for** $j = 3$ **to** $n - 1$
4.   **do** **if** v_j și vârful din top al lui $\mathcal{S}$ sunt în lanțuri diferite
5.   **then** extrage toate vârfurile din $\mathcal{S}$
6.   inserează diagonale de la v_j la vf. extrase, exceptând ultimul
7.   inserează v_{j-1} și v_j în $\mathcal{S}$
8.   **else** extrage un vârf din $\mathcal{S}$
9.   extrage celelalte vârfuri din $\mathcal{S}$ dacă diagonalele formate cu v_j sunt în interiorul lui $\mathcal{P}$; inserează aceste diagonale; inserează înapoi ultimul vârf extras
10.   inserează v_j în $\mathcal{S}$
11. adaugă diagonale de la v_n la vf. stivei (exceptând primul și ultimul)

Problematizare

- Triangularea unui poligon convex (listă ordonată de puncte $(P_1, P_2, \ldots, P_n)$.

Problematizare

- Triangularea unui poligon convex (listă ordonată de puncte $(P_1, P_2, \ldots, P_n)$.
- Are sens să vorbim de triangulare pentru mulțimea $\{P_1, P_2, \ldots, P_n\}$?

Problematizare

- Triangularea unui poligon convex (listă ordonată de puncte $(P_1, P_2, \ldots, P_n)$.
- Are sens să vorbim de triangulare pentru <u>mulțimea</u> $\{P_1, P_2, \ldots, P_n\}$?
- **Comentariu:** Triangulările mulțimilor de puncte sunt esențiale în grafica pe calculator.

Problematizare

- Triangularea unui poligon convex (listă ordonată de puncte $(P_1, P_2, \ldots, P_n)$.
- Are sens să vorbim de triangulare pentru <u>mulțimea</u> $\{P_1, P_2, \ldots, P_n\}$?
- **Comentariu:** Triangulările mulțimilor de puncte sunt esențiale în grafica pe calculator.
- **Definiție.** O **triangulare** a unei mulțimi $\mathcal{P}$ din plan este o subdivizare maximală a acoperirii convexe $\mathrm{Conv}(\mathcal{P})$ a lui $\mathcal{P}$ cu triunghiuri ale căror vârfuri sunt elemente ale lui $\mathcal{P}$ (fără autointersecții!)

Problematizare

- Triangularea unui poligon convex (listă ordonată de puncte $(P_1, P_2, \ldots, P_n)$.
- Are sens să vorbim de triangulare pentru mulțimea $\{P_1, P_2, \ldots, P_n\}$?
- **Comentariu:** Triangulările mulțimilor de puncte sunt esențiale în grafica pe calculator.
- **Definiție.** O **triangulare** a unei mulțimi $\mathcal{P}$ din plan este o subdivizare maximală a acoperirii convexe $\mathrm{Conv}(\mathcal{P})$ a lui $\mathcal{P}$ cu triunghiuri ale căror vârfuri sunt elemente ale lui $\mathcal{P}$ (fără autointersecții!)
- Trebuie făcută distincție între triangulare a unui poligon $(P_1, P_2, \ldots, P_n)$ și triangulare a mulțimii subdiacente $\{P_1, P_2, \ldots, P_n\}$ (coincid dacă poligonul este convex!)

Elemente ale unei triangulări

- Dată o mulțime de puncte $\mathcal{P}$ și o triangulare $\mathcal{T}_P$ a sa: **vârfuri, muchii, triunghiuri**.

Elemente ale unei triangulări

- Dată o mulțime de puncte $\mathcal{P}$ și o triangulare $\mathcal{T}_P$ a sa: **vârfuri, muchii, triunghiuri**.
- Legătură între aceste elemente?

Elemente ale unei triangulări

- Dată o mulțime de puncte $\mathcal{P}$ și o triangulare $\mathcal{T}_P$ a sa: **vârfuri, muchii, triunghiuri**.
- Legătură între aceste elemente?
- **Propoziție.** *Fie $\mathcal{P}$ o mulțime de n puncte din plan nesituate toate pe o aceeași dreaptă. Notăm cu k numărul de puncte de pe frontiera acoperirii convexe $\mathrm{Conv}(\mathcal{P})$. Orice triangulare a lui $\mathcal{P}$ are $(2n - k - 2)$ triunghiuri și $(3n - k - 3)$ muchii.*

Elemente ale unei triangulări

- Dată o mulțime de puncte $\mathcal{P}$ și o triangulare $\mathcal{T}_P$ a sa: **vârfuri, muchii, triunghiuri**.
- Legătură între aceste elemente?
- **Propoziție.** *Fie* $\mathcal{P}$ *o mulțime de* n *puncte din plan nesituate toate pe o aceeași dreaptă. Notăm cu* k *numărul de puncte de pe frontiera acoperirii convexe* $\mathrm{Conv}(\mathcal{P})$. *Orice triangulare a lui* $\mathcal{P}$ *are* $(2n-k-2)$ *triunghiuri și* $(3n-k-3)$ *muchii.*
- **Demonstrație:** Se bazează pe formula lui Euler / numărul de incidențe.

Elemente ale unei triangulări

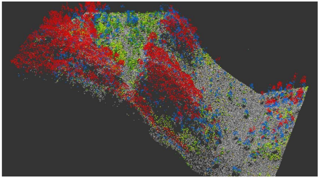
- Dată o mulțime de puncte $\mathcal{P}$ și o triangulare $\mathcal{T}_P$ a sa: **vârfuri, muchii, triunghiuri**.
- Legătură între aceste elemente?
- **Propoziție.** *Fie $\mathcal{P}$ o mulțime de n puncte din plan nesituate toate pe o aceeași dreaptă. Notăm cu k numărul de puncte de pe frontiera acoperirii convexe $\mathrm{Conv}(\mathcal{P})$. Orice triangulare a lui $\mathcal{P}$ are $(2n-k-2)$ triunghiuri și $(3n-k-3)$ muchii.*
- **Demonstrație:** Se bazează pe formula lui Euler / numărul de incidențe.
- **Exemplu:** Cazul unui poligon convex cu n vârfuri ($k=n$): $n-2$ triunghiuri și $2n-3$ muchii.

Problematizare

- **Problemă.** Se fac măsurători ale altitidinii pentru un teren. Se dorește reprezentarea tridimensională (cât mai sugestivă). Alternativ: se dorește generarea unui teren pentru o aplicație.

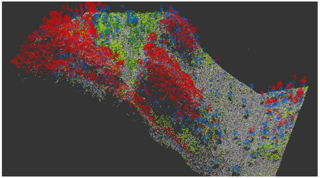
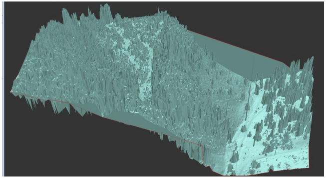
Problematizare

- **Problemă.** Se fac măsurători ale altitidinii pentru un teren. Se dorește reprezentarea tridimensională (cât mai sugestivă). Alternativ: se dorește generarea unui teren pentru o aplicație.



Problematizare

- **Problemă.** Se fac măsurători ale altitidinii pentru un teren. Se dorește reprezentarea tridimensională (cât mai sugestivă). Alternativ: se dorește generarea unui teren pentru o aplicație.

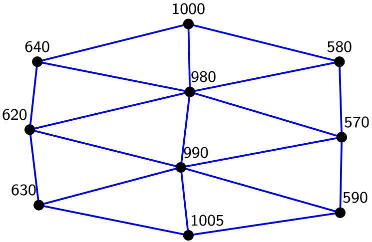



Problematizare - continuare

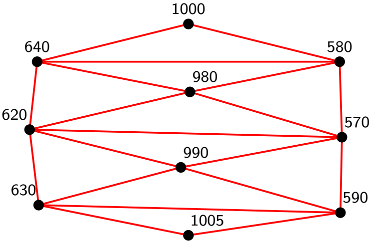
- **Problemă (reformulată).** Cum "comparăm triangulările" unei mulțimi de puncte fixate?

Problematizare - continuare

- **Problemă (reformulată).** Cum "comparăm triangulările" unei mulțimi de puncte fixate?
- **Exemplu.** Măsurători ale altitudinii.



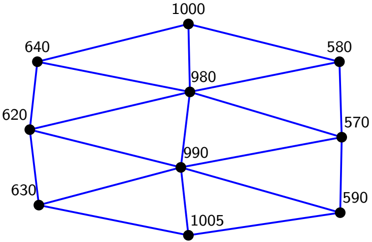
Triangulare 1



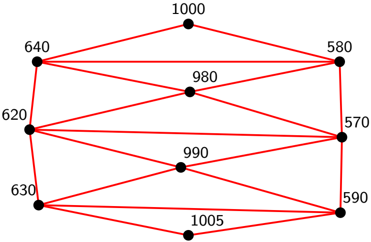
Triangulare 2

Problematizare - continuare

- **Problemă (reformulată).** Cum ”comparăm triangulările” unei mulțimi de puncte fixate?
- **Exemplu.** Măsurători ale altitudinii.



Triangulare 1



Triangulare 2

- **Întrebări naturale:** (i) Există o triangulare “convenabilă” a unei mulțimi de puncte? (ii) Cum poate fi determinată eficient o astfel de triangulare?

Terminologie, triangulări legale

- Fixată: o mulțime de puncte $\mathcal{P}$.

Terminologie, triangulări legale

- Fixată: o mulțime de puncte $\mathcal{P}$.
- Fie $\mathcal{T}$ o triangulare a lui $\mathcal{P}$ cu m triunghiuri. Fie $\alpha_1, \alpha_2, \ldots, \alpha_{3m}$ unghiurile lui $\mathcal{T}$, ordonate crescător. **Vectorul unghiurilor lui $\mathcal{T}$ este** $A(\mathcal{T}) = (\alpha_1, \alpha_2, \ldots, \alpha_{3m})$.

Terminologie, triangulări legale

- Fixată: o mulțime de puncte $\mathcal{P}$.
- Fie $\mathcal{T}$ o triangulare a lui $\mathcal{P}$ cu m triunghiuri. Fie $\alpha_1, \alpha_2, \ldots, \alpha_{3m}$ unghiurile lui $\mathcal{T}$, ordonate crescător. **Vectorul unghiurilor lui $\mathcal{T}$ este** $A(\mathcal{T}) = (\alpha_1, \alpha_2, \ldots, \alpha_{3m})$.
- **Relație de ordine pe mulțimea triangulărilor lui $\mathcal{P}$:** ordinea lexicografică pentru vectorii unghiurilor. Fie $\mathcal{T}$ și $\mathcal{T}'$ două triangulări ale lui $\mathcal{P}$. Atunci $A(\mathcal{T}) > A(\mathcal{T}')$ dacă $\exists i$ astfel ca $\alpha_j = \alpha'_j$, $\forall 1 \leq j < i$ și $\alpha_i > \alpha'_i$.

Terminologie, triangulări legale

- Fixată: o mulțime de puncte $\mathcal{P}$.
- Fie $\mathcal{T}$ o triangulare a lui $\mathcal{P}$ cu m triunghiuri. Fie $\alpha_1, \alpha_2, \ldots, \alpha_{3m}$ unghiurile lui $\mathcal{T}$, ordonate crescător. **Vectorul unghiurilor lui $\mathcal{T}$ este** $A(\mathcal{T}) = (\alpha_1, \alpha_2, \ldots, \alpha_{3m})$.
- **Relație de ordine pe mulțimea triangulărilor lui $\mathcal{P}$:** ordinea lexicografică pentru vectorii unghiurilor. Fie $\mathcal{T}$ și $\mathcal{T}'$ două triangulări ale lui $\mathcal{P}$. Atunci $A(\mathcal{T}) > A(\mathcal{T}')$ dacă $\exists i$ astfel ca $\alpha_j = \alpha'_j$, $\forall 1 \leq j < i$ și $\alpha_i > \alpha'_i$.
- **Triangulare unghiular optimă:** $\mathcal{T}$ astfel ca $A(\mathcal{T}) \geq A(\mathcal{T}')$, pentru orice triangulare $\mathcal{T}'$.

Terminologie, triangulări legale

- Fixată: o mulțime de puncte $\mathcal{P}$.
- Fie $\mathcal{T}$ o triangulare a lui $\mathcal{P}$ cu m triunghiuri. Fie $\alpha_1, \alpha_2, \ldots, \alpha_{3m}$ unghiurile lui $\mathcal{T}$, ordonate crescător. **Vectorul unghiurilor lui $\mathcal{T}$ este** $A(\mathcal{T}) = (\alpha_1, \alpha_2, \ldots, \alpha_{3m})$.
- **Relație de ordine pe mulțimea triangulărilor lui $\mathcal{P}$:** ordinea lexicografică pentru vectorii unghiurilor. Fie $\mathcal{T}$ și $\mathcal{T}'$ două triangulări ale lui $\mathcal{P}$. Atunci $A(\mathcal{T}) > A(\mathcal{T}')$ dacă $\exists i$ astfel ca $\alpha_j = \alpha'_j$, $\forall 1 \leq j < i$ și $\alpha_i > \alpha'_i$.
- **Triangulare unghiular optimă:** $\mathcal{T}$ astfel ca $A(\mathcal{T}) \geq A(\mathcal{T}')$, pentru orice triangulare $\mathcal{T}'$.
- **Exemplu:** Cazul unui patrulater convex.

Muchii ilegale

- **Conceptul de muchie ilegală.** Fie $A, B, C, D \in \mathbb{R}^2$ fixate astfel ca $ABCD$ să fie un patrulater convex; fie $\mathcal{T}_{AC}$, $\mathcal{T}_{BD}$ triangulările date de diagonalele AC, respectiv BD. Muchia AC este **ilegală** dacă
$$\min A(\mathcal{T}_{AC}) < \min A(\mathcal{T}_{BD}).$$

Muchii ilegale

- **Conceptul de muchie ilegală.** Fie $A, B, C, D \in \mathbb{R}^2$ fixate astfel ca $ABCD$ să fie un patrulater convex; fie $\mathcal{T}_{AC}$, $\mathcal{T}_{BD}$ triangulările date de diagonalele AC, respectiv BD. Muchia AC este **ilegală** dacă
$$\min A(\mathcal{T}_{AC}) < \min A(\mathcal{T}_{BD}).$$
- **Criteriu geometric** pentru a testa dacă o muchie este legală: muchia AC, adiacentă cu triunghiurile ΔACB și ΔACD este ilegală dacă și numai dacă punctul D este situat în interiorul cercului circumscris ΔABC.

Muchii ilegale

- **Conceptul de muchie ilegală.** Fie $A, B, C, D \in \mathbb{R}^2$ fixate astfel ca $ABCD$ să fie un patrulater convex; fie $\mathcal{T}_{AC}$, $\mathcal{T}_{BD}$ triangulările date de diagonalele AC, respectiv BD. Muchia AC este **ilegală** dacă
$$\min A(\mathcal{T}_{AC}) < \min A(\mathcal{T}_{BD}).$$
- **Criteriu geometric** pentru a testa dacă o muchie este legală: muchia AC, adiacentă cu triunghiurile ΔACB și ΔACD este ilegală dacă și numai dacă punctul D este situat în interiorul cercului circumscris ΔABC.
- **Criteriu numeric / analitic** pentru a testa dacă o muchie este ilegală.
  - Pentru puncte $A = (x_A, y_A), B = (x_B, y_B), C = (x_C, y_C), D = (x_D, y_D)$:

$$\Theta(A, B, C, D) = \begin{vmatrix} x_A & y_A & x_A^2 + y_A^2 & 1 \\ x_B & y_B & x_B^2 + y_B^2 & 1 \\ x_C & y_C & x_C^2 + y_C^2 & 1 \\ x_D & y_D & x_D^2 + y_D^2 & 1 \end{vmatrix}$$

Muchii ilegale

- **Conceptul de muchie ilegală.** Fie $A, B, C, D \in \mathbb{R}^2$ fixate astfel ca $ABCD$ să fie un patrulater convex; fie $\mathcal{T}_{AC}$, $\mathcal{T}_{BD}$ triangulările date de diagonalele AC, respectiv BD. Muchia AC este **ilegală** dacă
$$\min A(\mathcal{T}_{AC}) < \min A(\mathcal{T}_{BD}).$$
- **Criteriu geometric** pentru a testa dacă o muchie este legală: muchia AC, adiacentă cu triunghiurile ΔACB și ΔACD este ilegală dacă și numai dacă punctul D este situat în interiorul cercului circumscris ΔABC.
- **Criteriu numeric / analitic** pentru a testa dacă o muchie este ilegală.
  - Pentru puncte $A = (x_A, y_A), B = (x_B, y_B), C = (x_C, y_C), D = (x_D, y_D)$:
$$\Theta(A, B, C, D) = \begin{vmatrix} x_A & y_A & x_A^2 + y_A^2 & 1 \\ x_B & y_B & x_B^2 + y_B^2 & 1 \\ x_C & y_C & x_C^2 + y_C^2 & 1 \\ x_D & y_D & x_D^2 + y_D^2 & 1 \end{vmatrix}$$
  - (i) Punctele A, B, C, D sunt conciclice $\Leftrightarrow \Theta(A, B, C, D) = 0$.
(ii) Fie A, B, C astfel ca ABC să fie un viraj la stânga. Un punct D este situat în interiorul cercului circumscris $\Delta ABC \Leftrightarrow \Theta(A, B, C, D) > 0$.

Muchii ilegale, triangulări legale

- **Concluzie:** Muchia AC este ilegală dacă, printr-un *flip* (înlocuirea ei cu BD), cel mai mic unghi poate fi mărit (local).

Muchii ilegale, triangulări legale

- **Concluzie:** Muchia AC este ilegală dacă, printr-un *flip* (înlocuirea ei cu BD), cel mai mic unghi poate fi mărit (local).
- **Concluzie (reformulare):** Fie $\mathcal{T}$ o triangulare cu o muchie ilegală e, fie $\mathcal{T}'$ triangularea obținută din $\mathcal{T}$ prin *flip*-ul muchiei e. Atunci $A(\mathcal{T}') > A(\mathcal{T})$.

Muchii ilegale, triangulări legale

- **Concluzie:** Muchia AC este ilegală dacă, printr-un *flip* (înlocuirea ei cu BD), cel mai mic unghi poate fi mărit (local).
- **Concluzie (reformulare):** Fie $\mathcal{T}$ o triangulare cu o muchie ilegală e, fie $\mathcal{T}'$ triangularea obținută din $\mathcal{T}$ prin *flip*-ul muchiei e. Atunci $A(\mathcal{T}') > A(\mathcal{T})$.
- **Triangulare legală:** nu are muchii ilegale. **Fapt:** O triangulare legală poate fi determinată printr-un algoritm incremental randomizat, cu complexitate-timp medie $O(n \log n)$.

Triangulări unghiular optime vs. triangulări legale

- **Propoziție.** *Fie $\mathcal{P}$ o mulțime de puncte din plan.*
  (i) *Orice triangulare unghiular optimă este legală.*
  (ii) *Dacă $\mathcal{P}$ este în poziție generală (oricare patru puncte nu sunt conciclice), atunci există o unică triangulare legală, iar aceasta este unghiular optimă.*

Triangulări unghiular optime vs. triangulări legale

- **Propoziție.** *Fie $\mathcal{P}$ o mulțime de puncte din plan.*
  (i) *Orice triangulare unghiular optimă este legală.*
  (ii) *Dacă $\mathcal{P}$ este în poziție generală (oricare patru puncte nu sunt conciclice), atunci există o unică triangulare legală, iar aceasta este unghiular optimă.*
- **Teoremă.** *Fie $\mathcal{P}$ o mulțime de n puncte din plan, în poziție generală. Triangularea unghiular optimă poate fi construită, folosind un algoritm incremental randomizat, în timp mediu $O(n \log n)$, folosind $O(n)$ memorie medie.*