Algoritmi avansați

C9 - Triangularea poligoanelor

Mihai-Sorin Stupariu

Sem. al II-lea, 2024 - 2025

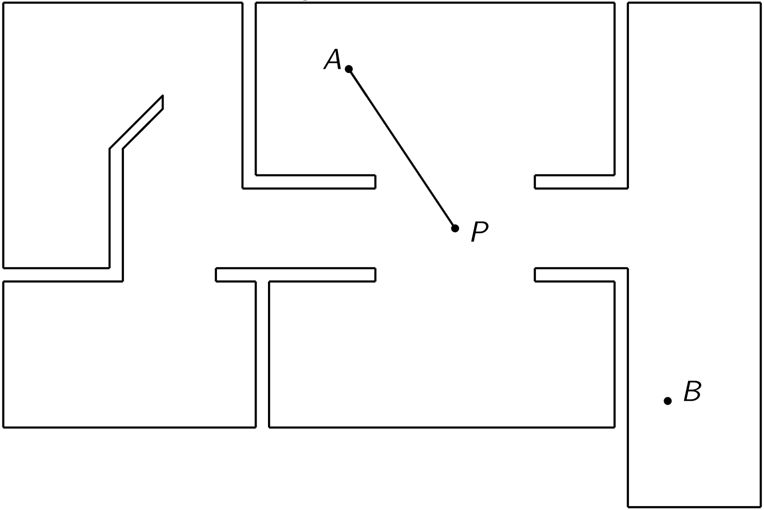
Problema galeriei de artă

Algoritmi de triangulare - “Ear clipping”

Triangularea poligoanelor - un algoritm eficient

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 $\mathcal{P}$ (adică o linie poligonală fără autointersecții) având n vârfuri.

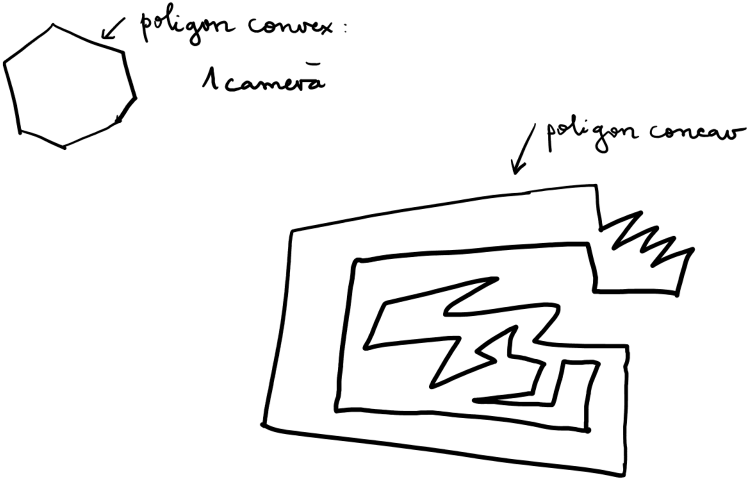
Formalizare

- O galerie de artă poate fi interpretată (în contextul acestei probleme) ca un poligon $\mathcal{P}$ (adică o linie poligonală 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 $\mathcal{P}$ (adică o linie poligonală 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?*

Comentarii



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.

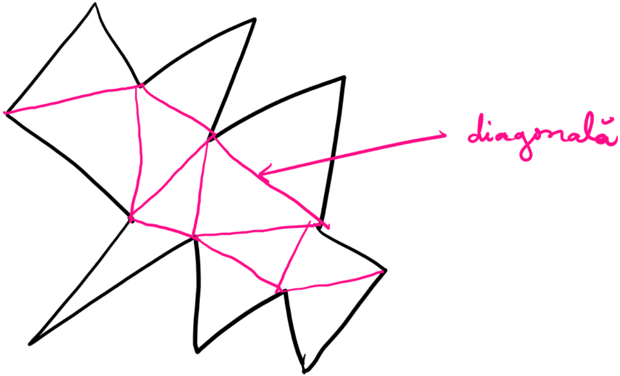
Numărul de camere vs. forma poligonului

- 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).

Despre triangulări



Definiție formală

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

Definiție formală

- 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 $\mathcal{P}$.

Definiție formală

- 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.

Rezultate

- **Lemă.** Orice poligon admite o diagonală.

Rezultate

- **Lemă.** Orice poligon admite o diagonală.
- **Teoremă.** *Orice poligon admite o triangulare. Orice triangulare a unui poligon cu n vârfuri conține exact $n-2$ triunghiuri.*

Rezolvarea problemei galeriei de artă

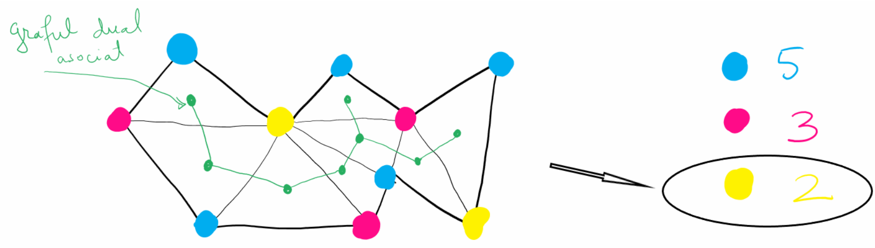
- Amplasarea camerelor se poate face în vârfurile poligonului.

Rezolvarea 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 corepunde o culoare dintr-un set de 3 culori și pentru fiecare triunghi, cele 3 vârfuri au culori distincte.

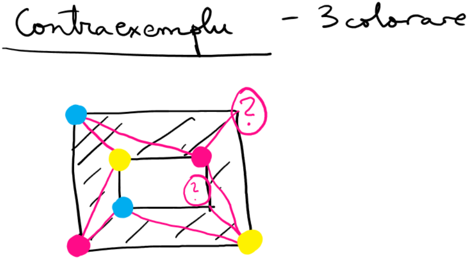
Rezolvarea problemei galeriei de artă

- 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 corepunde o culoare dintr-un set de 3 culori și pentru fiecare triunghi, cele 3 vârfuri au culori distincte.
- **Observație.** Dacă $\mathcal{P}$ este linie poligonală fără autointersecții o astfel de colorare există, deoarece graful dual asociat perechii $(\mathcal{P}, \mathcal{T}_P)$ este arbore.



Rezolvarea 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 corepunde o culoare dintr-un set de 3 culori și pentru fiecare triunghi, cele 3 vârfuri au culori distincte.
- **Observație.** Dacă $\mathcal{P}$ este linie poligonală fără autointersecții o astfel de colorare există, deoarece graful dual 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

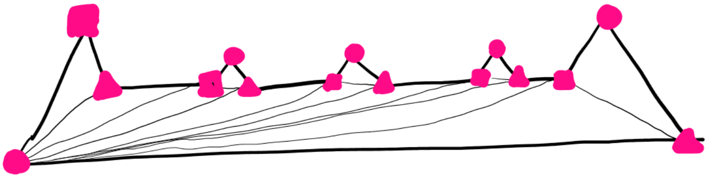
- **Teoremă.** [Chvátal, 1975; Fisk, 1978] *Pentru un poligon cu n vârfuri,* $\left[\frac{n}{3}\right]$ *camere sunt* **uneori necesare** *și întotdeauna* **suficiente** *pentru ca fiecare punct al poligonului să fie vizibil din cel puțin una din camere.*
- Despre Teorema Galeriei de Artă: J. O'Rourke,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

Teorema galeriei de artă

- **Teoremă.** [Chvátal, 1975; Fisk, 1978] *Pentru un poligon cu n vârfuri,* $\left\lfloor \frac{n}{3} \right\rfloor$ *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*; J. Urrutia, *Art Gallery and Illumination Problems*
- Despre numărul de culori utilizat: L. Erickson, S. LaValle, *A chromatic art gallery problem*, Fekete et al., *On the Chromatic Art Gallery Problem*

Teorema galeriei de artă - justificare, exemplu

- **“uneori necesare”**



Teorema galeriei de artă - justificare, exemplu

- **“întotdeauna suficiente”**

Teorema galeriei de artă - justificare, exemplu

- **“întotdeauna suficiente”**

Notăm cu n_1, n_2, n_3 numărul de vârfuri colorate cu cele 3 culori, are loc relația $n_1 + n_2 + n_3 = n$.

Teorema galeriei de artă - justificare, exemplu

- **"întotdeauna suficiente"**

Notăm cu n_1, n_2, n_3 numărul de vârfuri colorate cu cele 3 culori, are loc relația $n_1 + n_2 + n_3 = n$.

Presupunem prin absurd ca nu sunt suficiente $\left[\frac{n}{3}\right]$ camere. Aceasta înseamnă că:

Teorema galeriei de artă - justificare, exemplu

- **"întotdeauna suficiente"**

Notăm cu n_1, n_2, n_3 numărul de vârfuri colorate cu cele 3 culori, are loc relația $n_1 + n_2 + n_3 = n$.

Presupunem prin absurd ca nu sunt suficiente $\left[\frac{n}{3}\right]$ camere. Aceasta înseamnă că:

$$n_1 > \left[\frac{n}{3}\right] \Rightarrow n_1 > \frac{n}{3} \quad \text{(folosind definiția părții întregi)}$$

$$n_2 > \left[\frac{n}{3}\right] \Rightarrow n_2 > \frac{n}{3} \quad \text{(folosind definiția părții întregi)}$$

$$n_3 > \left[\frac{n}{3}\right] \Rightarrow n_3 > \frac{n}{3} \quad \text{(folosind definiția părții întregi)}$$

Teorema galeriei de artă - justificare, exemplu

- **“întotdeauna suficiente”**

Notăm cu n_1, n_2, n_3 numărul de vârfuri colorate cu cele 3 culori, are loc relația $n_1 + n_2 + n_3 = n$.

Presupunem prin absurd ca nu sunt suficiente $\left[\frac{n}{3}\right]$ camere. Aceasta înseamnă că:

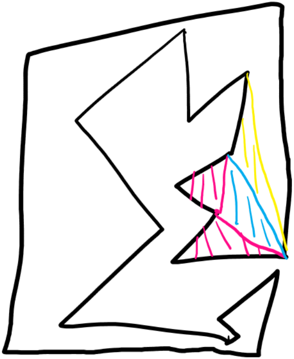
$$n_1 > \left[\frac{n}{3}\right] \Rightarrow n_1 > \frac{n}{3} \quad \text{(folosind definiția părții întregi)}$$

$$n_2 > \left[\frac{n}{3}\right] \Rightarrow n_2 > \frac{n}{3} \quad \text{(folosind definiția părții întregi)}$$

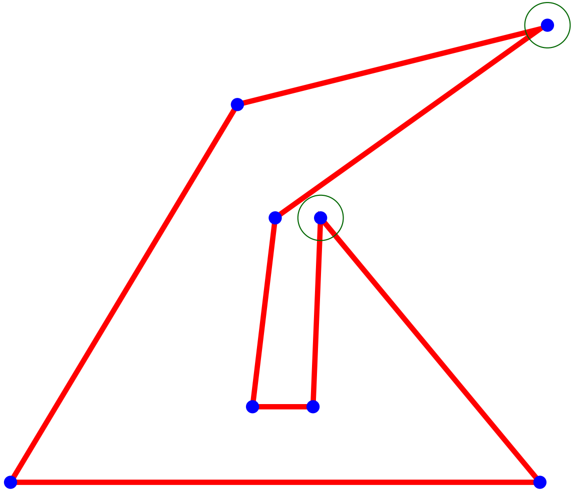
$$n_3 > \left[\frac{n}{3}\right] \Rightarrow n_3 > \frac{n}{3} \quad \text{(folosind definiția părții întregi)}$$

Adunând relațiile obținute se ajunge la relația $n_1 + n_2 + n_3 > n$, ceea ce reprezintă o contradicție. În concluzie, există i astfel ca $n_i \leq \left[\frac{n}{3}\right]$.

Triangularea unui poligon - intuiție

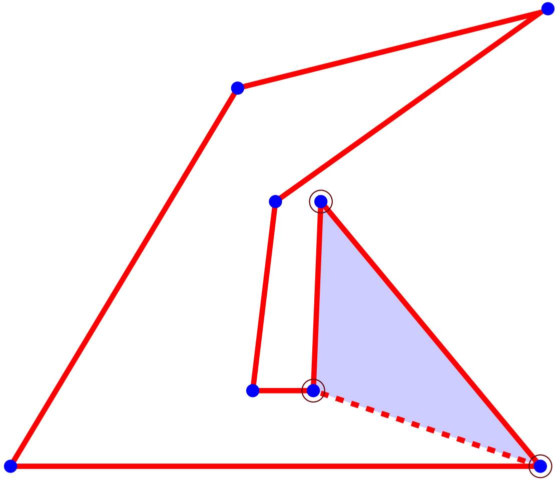


Triangularea unui poligon - algoritmul “Ear clipping”



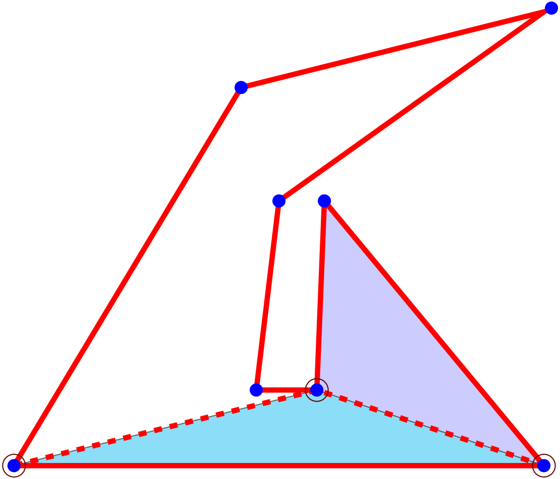
Vârfuri care pot fi selectate pentru start.

Triangularea unui poligon - algoritmul “Ear clipping”



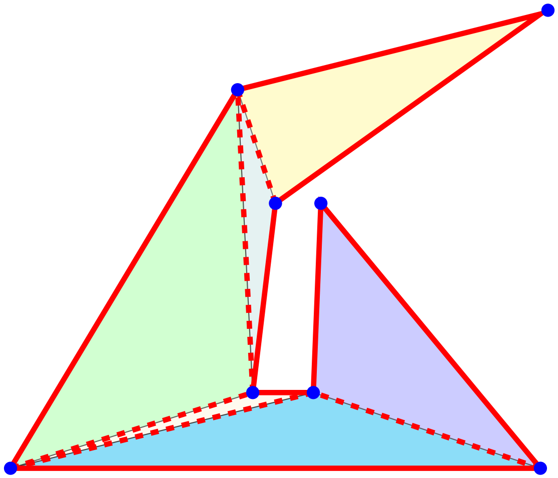
Ales un vârf, este considerat triunghiul determinat cu predecesorul și succesorul.

Triangularea unui poligon - algoritmul “Ear clipping”



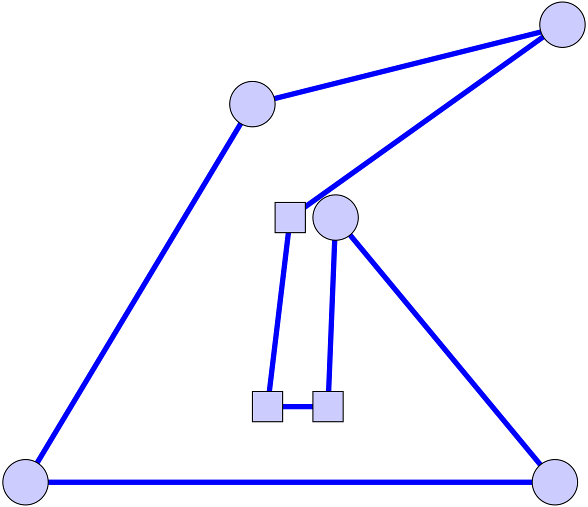
Procedura continuă....

Triangularea unui poligon - algoritmul “Ear clipping”



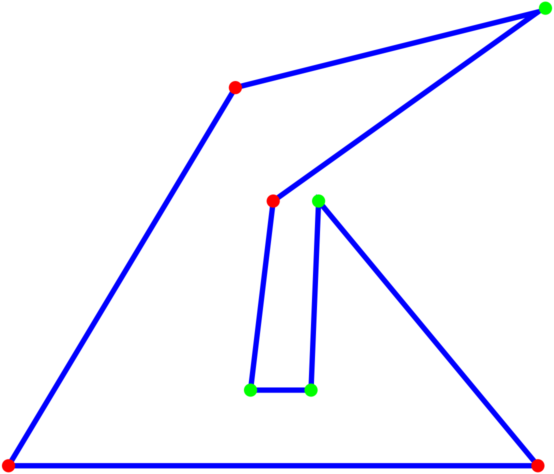
... se obține o triangulare a poligonului.

Clasificarea vârfurilor unui poligon - convexe/concave



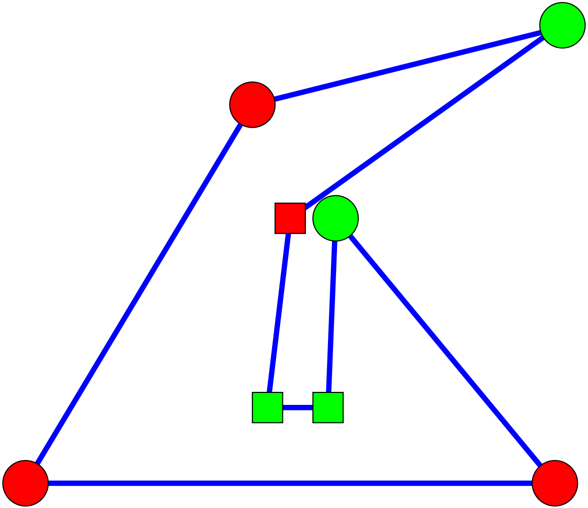
Vârfurile convexe (cerc) / concave (pătrat).

Clasificarea vârfurilor unui poligon - principale/neprincipale



Vârfurile principale (verde) / neprincipale (roșu).

Clasificarea vârfurilor unui poligon



Patru tipuri de vârfuri.

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. Un vârf este convex $\Leftrightarrow$ are același tip de viraj ca vârful “cel mai din stânga”.
  - **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. Un vârf este convex $\Leftrightarrow$ are același tip de viraj ca vârful “cel mai din stânga”.
  - **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. Un vârf este convex $\Leftrightarrow$ are același tip de viraj ca vârful "cel mai din stânga".
  - **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. Un vârf este convex $\Leftrightarrow$ are același tip de viraj ca vârful "cel mai din stânga".
  - **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 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. Un vârf este convex $\Leftrightarrow$ are același tip de viraj ca vârful "cel mai din stânga".
  - **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 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. Un vârf este convex $\Leftrightarrow$ are același tip de viraj ca vârful "cel mai din stânga".
  - **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 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

- 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

- 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, avem următoarea
  Teoremă. *Un poligon poate fi triangulat folosind un algoritm de complexitate* $O(n \log n)$.

Metode de triangulare: descompunerea în poligoane monotone

- 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, avem următoarea
  Teoremă. *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

- 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, avem următoarea
  Teoremă. *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*

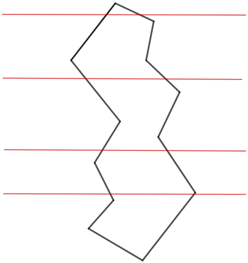
Metode de triangulare: descompunerea în poligoane monotone

- Concept: **poligon y-monoton**

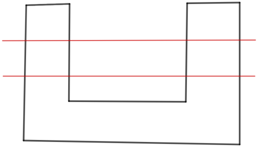
Metode de triangulare: descompunerea în poligoane monotone

- Concept: **poligon** y**-monoton**

Exemplul 1



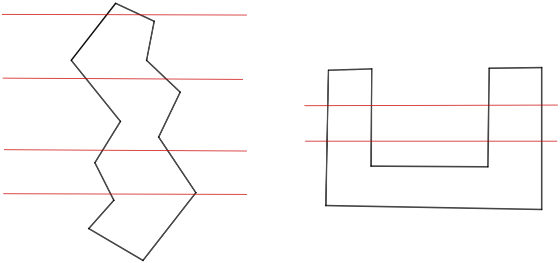
Exemplul 2



Metode de triangulare: descompunerea în poligoane monotone

- Concept: **poligon** y**-monoton**

Exemplul 1 Exemplul 2



- Poligonul din Exemplul 1 **este** y**-monoton**: (i) poate fi parcurs de sus în jos în exact două moduri, fără întoarceri - există două lanțuri pe care se poate realiza parcurgerea, (ii) orice dreaptă orizontală intersectează reuniunea dintre poligon și interior după o mulțime conexă ($\emptyset$, punct sau segment). Poligonul din Exemplul 2 **nu este** y**-monoton**.

Metoda - paradigma dreptei de baleiere (*line sweep*)

- Se consideră o dreaptă de baleiere orizontală. Algoritmul reține o serie de informații legate de structura geometrică analizată.

Metoda - paradigma dreptei de baleiere (*line sweep*)

- Se consideră o dreaptă de baleiere orizontală. Algoritmul reține o serie de informații legate de structura geometrică analizată.
- **Statut** al dreptei de baleiere: stivă a vârfurilor deja întâlnite, dar care "mai au nevoie de diagonale" / "mai pot să apară în triunghiuri. (Clarificare. **Q:** Când este eliminat un vârf? **A:** Când a fost trasată o diagonală situată "mai jos de acesta").

Metoda - paradigma dreptei de baleiere (*line sweep*)

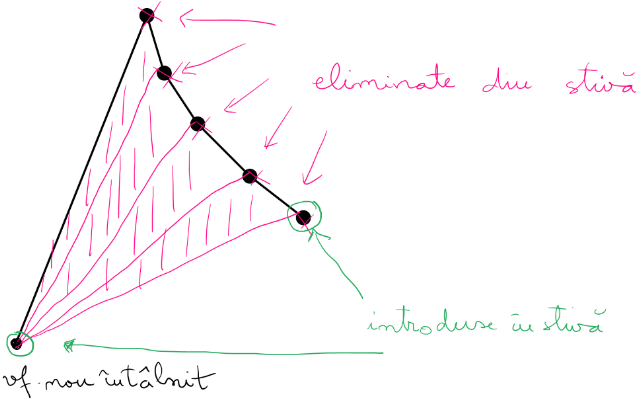
- Se consideră o dreaptă de baleiere orizontală. Algoritmul reține o serie de informații legate de structura geometrică analizată.
- **Statut** al dreptei de baleiere: stivă a vârfurilor deja întâlnite, dar care "mai au nevoie de diagonale" / "mai pot să apară în triunghiuri. (Clarificare. **Q:** Când este eliminat un vârf? **A:** Când a fost trasată o diagonală situată "mai jos de acesta").
- **Evenimente:** modificarea statutului. Sunt vârfurile poligonului, în prealabil ordonate după y; pentru fiecare vârf știm dacă este pe lanțul din stânga sau pe cel din dreapta.

Metoda - paradigma dreptei de baleiere (*line sweep*)

- Se consideră o dreaptă de baleiere orizontală. Algoritmul reține o serie de informații legate de structura geometrică analizată.
- **Statut** al dreptei de baleiere: stivă a vârfurilor deja întâlnite, dar care "mai au nevoie de diagonale" / "mai pot să apară în triunghiuri. (Clarificare. **Q:** Când este eliminat un vârf? **A:** Când a fost trasată o diagonală situată "mai jos de acesta").
- **Evenimente:** modificarea statutului. Sunt vârfurile poligonului, în prealabil ordonate după y; pentru fiecare vârf știm dacă este pe lanțul din stânga sau pe cel din dreapta.
- **Invariant:** "pâlnie" (*funnel*) în care (i) vârful de sus este convex; (ii) pe o parte: o muchie; (iii) pe cealaltă parte: muchie / succesiune de vârfuri concave.

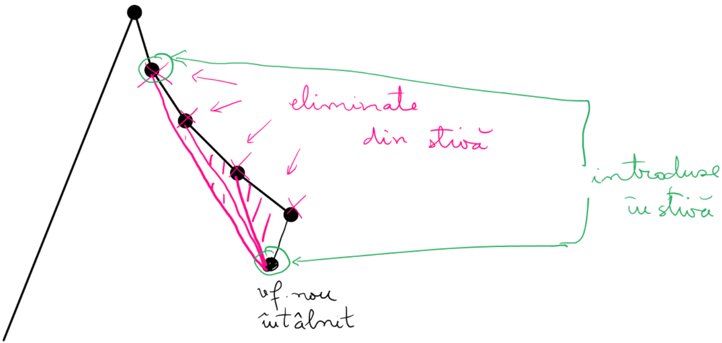
Evenimente - cazul 1

1. Vârful nou întâlnit este pe lanțul opus ultimului vârf din stivă.



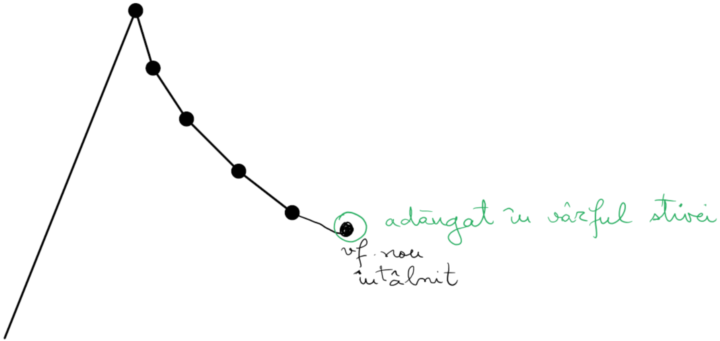
Evenimente - cazul 2a

2a. Vârful nou întâlnit este pe același lanț cu ultimul vârf din stivă.

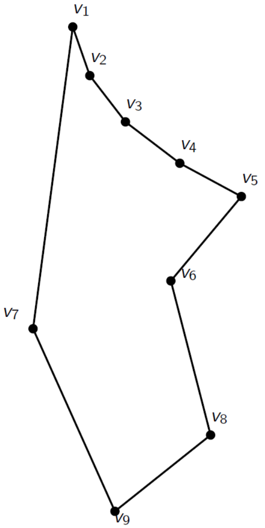


Evenimente - cazul 2b

2b. Vârful nou întâlnit este pe același lanț cu ultimul vârf din stivă.

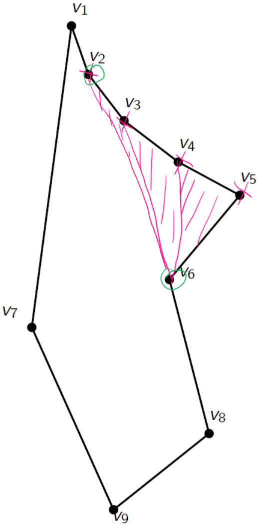


Exemplu



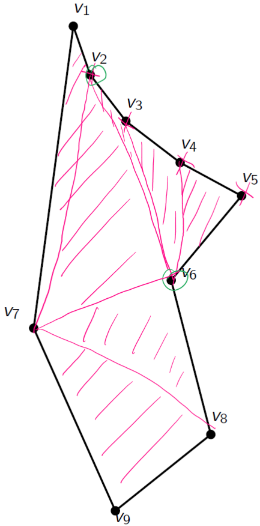
Eveniment	Statut
	v_2 v_1
v_3 (caz 2b)	v_3 v_2 v_1
v_4 (caz 2b)	v_4 v_3 v_2 v_1

Exemplu



Eveniment	Statut
v_5 (caz 2b)	v_5, v_4, v_3, v_2, v_1
v_6 (caz 2a)	~~v_5~~, ~~v_4~~, ~~v_3~~, ~~v_2~~, v_1 ; v_6, v_2, v_1

Exemplu



Eveniment	Statut
v_7 (caz 1)	$\cancel{v_6}$, $\cancel{v_2}$, $\cancel{v_1}$ → v_7, v_6
v_8 (Caz 1)	$\cancel{v_7}$, $\cancel{v_6}$ → v_8, v_7
v_9	Δ

La fiecare pas: diagonale și Δ în mod adecvat.

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 $\mathcal{P}$.

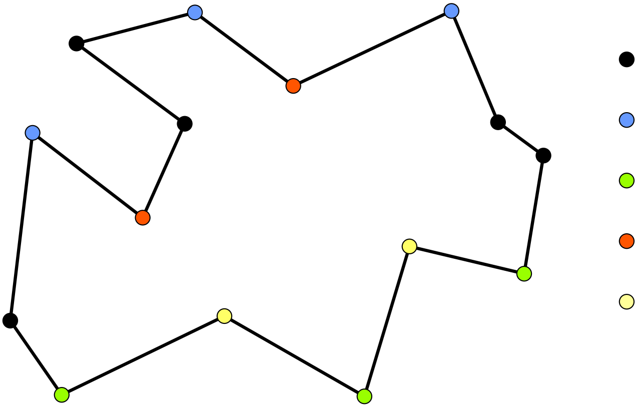
1. Lanțul vârfurilor din partea stângă și al celor din partea dreaptă sunt unite într-un singur șir, ordonat descrescător, dupa y (dacă ordonata este egală, se folosește abscisa). Fie $v_1, v_2, \ldots, v_n$ șirul ordonat.
2. Inițializează o stivă vidă $\mathcal{S}$ și inserează v_1, v_2.
3. **for**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

Triangularea poligoanelor monotone

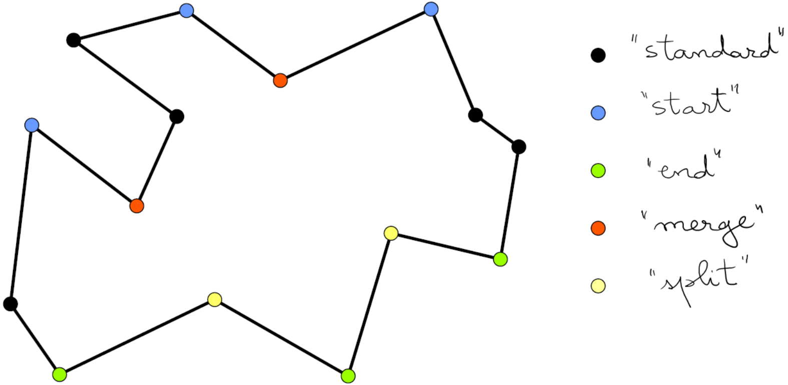
Input: Un poligon y-monoton $\mathcal{P}$.
Output: O triangulare a lui $\mathcal{P}$.

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)

Tipuri de vârfuri



Tipuri de vârfuri



Rezultate

- Folosind un algoritm bazat pe paradigma dreptei de baleiere și clasificarea vârfurilor indicată, un poligon cu n vârfuri poate fi descompus în poligoane y-monotone cu un algoritm având complexitatea-timp $O(n \log n)$.

Rezultate

- Folosind un algoritm bazat pe paradigma dreptei de baleiere și clasificarea vârfurilor indicată, un poligon cu n vârfuri poate fi descompus în poligoane y-monotone cu un algoritm având complexitatea-timp $O(n \log n)$.
- Conform algoritmului descris, un poligon y-monoton poate fi triangulat în timp liniar.

Rezultate

- Folosind un algoritm bazat pe paradigma dreptei de baleiere și clasificarea vârfurilor indicată, un poligon cu n vârfuri poate fi descompus în poligoane y-monotone cu un algoritm având complexitatea-timp $O(n \log n)$.
- Conform algoritmului descris, un poligon y-monoton poate fi triangulat în timp liniar.
- **Teoremă (rezultatul principal)** *Un poligon cu n vârfuri poate fi triangulat cu complexitatea-timp* $O(n \log n)$ *și complexitatea-spațiu* $O(n)$.

Scurt rezumat

- Teorema galeriei de artă, exemple de aplicare (și la seminar).

Scurt rezumat

- Teorema galeriei de artă, exemple de aplicare (și la seminar).
- Clasificarea vârfurilor unui poligon (convexe/concave, principale/neprincipale).

Scurt rezumat

- Teorema galeriei de artă, exemple de aplicare (și la seminar).
- Clasificarea vârfurilor unui poligon (convexe/concave, principale/neprincipale).
- Algoritm de triangulare (*ear clipping*) cu complexitate-timp pătratică.

Scurt rezumat

- Teorema galeriei de artă, exemple de aplicare (și la seminar).
- Clasificarea vârfurilor unui poligon (convexe/concave, principale/neprincipale).
- Algoritm de triangulare (*ear clipping*) cu complexitate-timp pătratică.
- Paradigma dreptei de baleiere.

Scurt rezumat

- Teorema galeriei de artă, exemple de aplicare (și la seminar).
- Clasificarea vârfurilor unui poligon (convexe/concave, principale/neprincipale).
- Algoritm de triangulare (*ear clipping*) cu complexitate-timp pătratică.
- Paradigma dreptei de baleiere.
- Algoritm de triangulare liniar pentru poligoane monotone. Aplicație la algoritm de triangulare cu complexitate timp $O(n \log n)$.