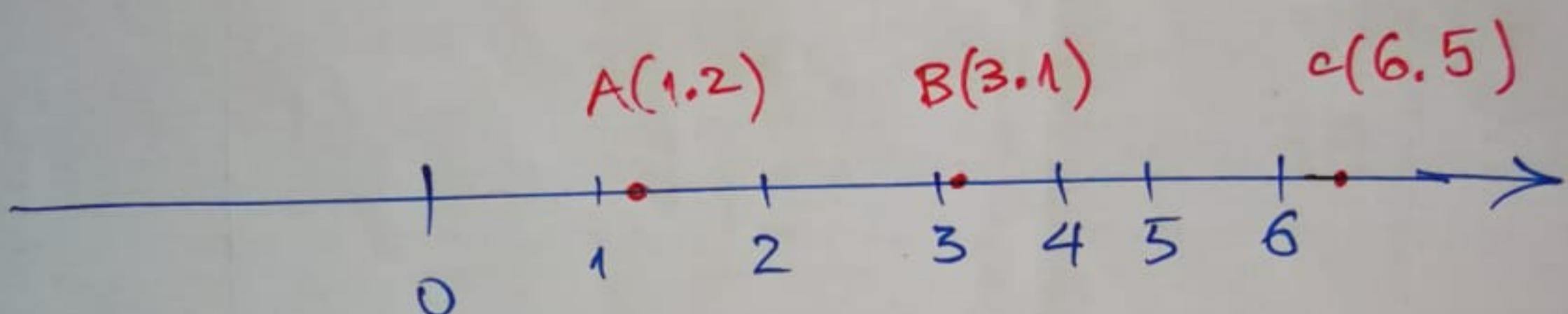


Relații între puncte

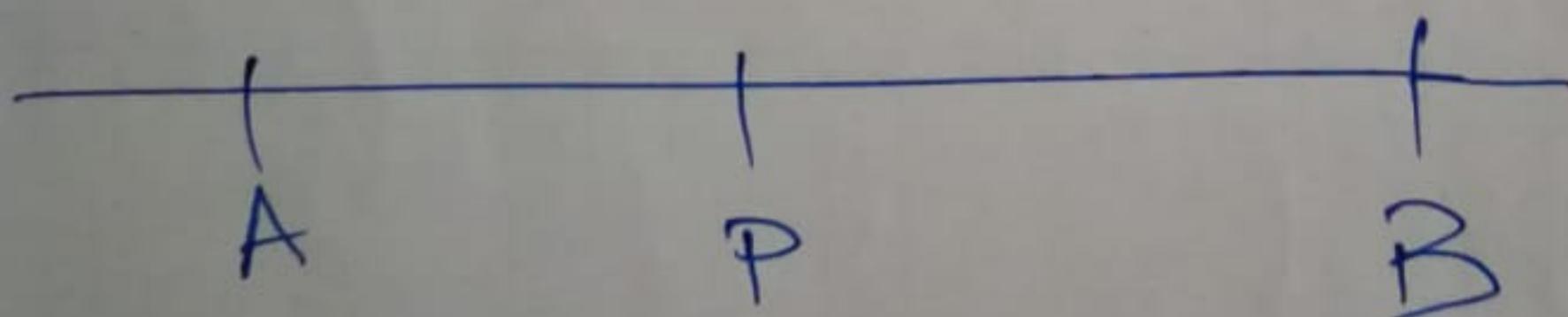
Context 1D

- Coordonate (pe dreaptă)



B este între A și C.

- Fără sistem de coordonate pe dreaptă care conține punctele



Obs. Punctele A, P, B sunt coliniare \Leftrightarrow
 $(A \neq B, P \neq B)$

$\exists n \in \mathbb{R}$ a.i. $\boxed{\overrightarrow{AP} = n \overrightarrow{PB}}$ (i.e. vectorii \overrightarrow{AP} ,
 \overrightarrow{PB} coliniari).
 n s.m. raportul punctelor A, P, B

Convenție / Notație

$\text{Ie } A, B \in \mathbb{R}^d; \overrightarrow{AB} \stackrel{\text{def}}{=} B - A$

De exemplu, în \mathbb{R}^3 :

$$A = (x_A, y_A, z_A); \quad B = (x_B, y_B, z_B);$$

$$\overrightarrow{AB} = (x_B - x_A, y_B - y_A, z_B - z_A).$$

Exemplu.

$$A = (1, 2, 3); \quad B = (2, 1, -1); \quad C = (0, 3, 7).$$

$$n(A, C, B) = ?$$

$$\overrightarrow{AC} = C - A = (0, 3, 7) - (1, 2, 3) = (-1, 1, 4)$$

$$\overrightarrow{CB} = B - C = (2, -2, -8)$$

$$\begin{aligned} \overrightarrow{AC} &= \frac{1}{2} \cdot \overrightarrow{CB} \\ &= n(A, C, B). \end{aligned}$$

Combinări liniare, affine și convexe
 ↓
 vectori puncte
 În \mathbb{R}^d .
 • $v_1, v_2, \dots, v_p \in \mathbb{R}^d$ vectori; o combinație liniară este
 un vector de forma $\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_p v_p$
 $\alpha_1, \alpha_2, \dots, \alpha_p \in \mathbb{R}$

• $A_1, A_2, \dots, A_p \in \mathbb{R}^d$ puncte
 combinație afină (baricentrică) este un punct
 de forma $\lambda_1 A_1 + \lambda_2 A_2 + \dots + \lambda_p A_p$
 $\lambda_1, \lambda_2, \dots, \lambda_p \in \mathbb{R}; \lambda_1 + \dots + \lambda_p = 1.$

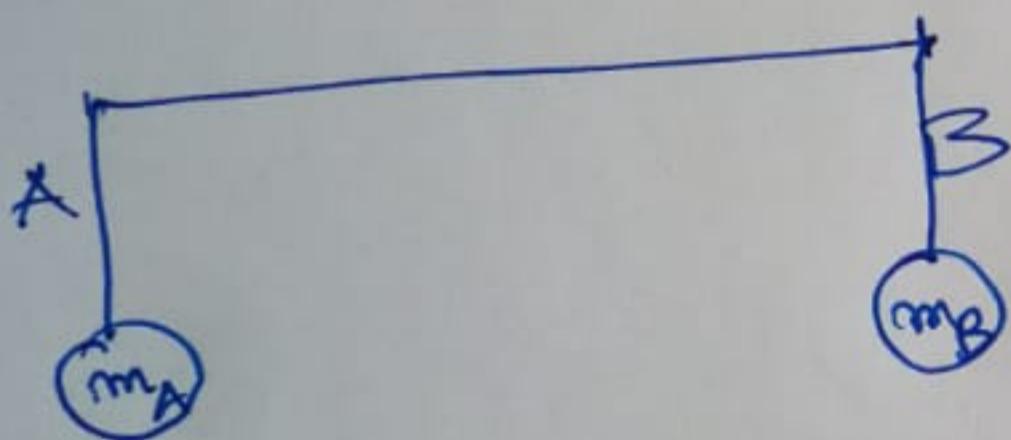
• $A_1, A_2, \dots, A_p \in \mathbb{R}^d$ puncte
 combinație convexă este un punct de forma
 $\lambda_1 A_1 + \lambda_2 A_2 + \dots + \lambda_p A_p$
 $\lambda_1, \lambda_2, \dots, \lambda_p \in [0, 1]$
 $\lambda_1 + \lambda_2 + \dots + \lambda_p = 1.$

Fie A și B două puncte distincte.

O combinație afină este de forma

$$\lambda A + \mu B = (1-\alpha)A + \alpha B$$
$$\lambda + \mu = 1$$
$$\lambda, \mu \in \mathbb{R}$$
$$\alpha \in \mathbb{R}$$

→ situat pe dreapta AB, iar o combinație convexă pe segmentul [AB].



centrul de greutate:

$$\frac{m_A}{m_A + m_B} \cdot A + \frac{m_B}{m_A + m_B} \cdot B$$

combinare convexă a punctelor A și B.

Exemplu:

$$A = (1, 2, 3)$$

$$B = (2, 1, -1)$$

$$C = (0, 3, 7)$$

$$r(A, C, B) = -\frac{1}{2}.$$

(exemplul anterior).

$$\vec{AC} = -\frac{1}{2} \vec{CB} \quad (\text{stiu calculul anterior}) \Rightarrow$$

$$\Rightarrow \vec{CA} - \frac{1}{2} \vec{CB} = 0 = \vec{CC} \quad | \cdot 2$$

$$2\cancel{\vec{CA}} - \cancel{\vec{CB}} = 0 = \cancel{\vec{CC}} \quad \triangleleft$$

$$\underline{\underline{C = 2A - B}}$$

Concluzie: dacă A, P, B coliniare (distingute)

putem caracteriza poziția lor relativă:

• folosind raportul $r = r(A, P, B)$

• folosind combinații affine ($P = (1-\alpha)A + \alpha B$)

• \mathbb{R}^3 (abordare numerică)

$$v = (v_1, v_2, v_3); w = (w_1, w_2, w_3); v, w \in \mathbb{R}^3$$

$$v \times w = ? (\in \mathbb{R}^3)$$

$$v \times w = \begin{vmatrix} v_1 & w_1 & e_1 \\ v_2 & w_2 & e_2 \\ v_3 & w_3 & e_3 \end{vmatrix}$$

determinant
formal

⚠ Deosebit de produsul scalar (dot product)

$$\langle v, w \rangle = v \cdot w = v_1 w_1 + v_2 w_2 + v_3 w_3.$$

Exemplu: $v = (4, 1, 0); w = (0, 2, 1)$

$$\langle v, w \rangle = v \cdot w = 4 \cdot 0 + 1 \cdot 2 + 0 \cdot 1 = 2$$

$$v \times w = \begin{vmatrix} 4 & 0 & e_1 \\ 1 & 2 & e_2 \\ 0 & 1 & e_3 \end{vmatrix}$$

dezvoltare după
ultima coloană

$$\begin{aligned}
 &= \begin{vmatrix} 1 & 2 \\ 0 & 1 \end{vmatrix} e_1 - \begin{vmatrix} 4 & 0 \\ 0 & 1 \end{vmatrix} e_2 + \begin{vmatrix} 4 & 0 \\ 1 & 2 \end{vmatrix} e_3 = \\
 &= 1 \cdot e_1 - 4 \cdot e_2 + 8 \cdot e_3 = (1, -4, 8) (= v \times w).
 \end{aligned}$$

observatie (importantă)

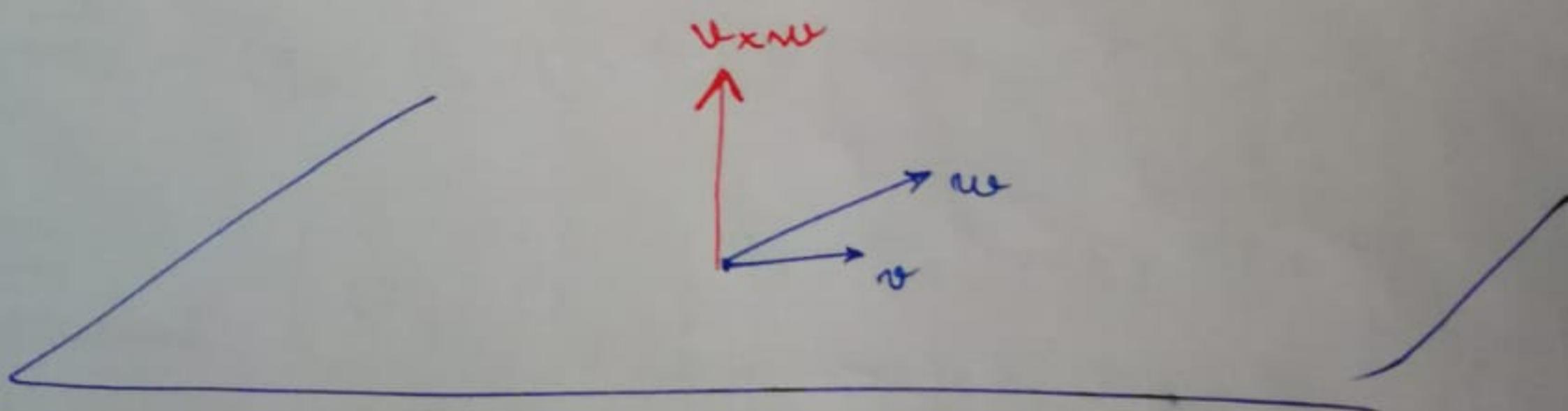
cum se calculează produsul vectorial și ce rezultat se obține pentru doi vectori "orizontali"?

$$v = (v_1, v_2, 0)$$

$$w = (w_1, w_2, 0).$$

$$v \times w = \begin{vmatrix} v_1 & w_1 & e_1 \\ v_2 & w_2 & e_2 \\ 0 & 0 & e_3 \end{vmatrix} = (0, 0, \begin{vmatrix} v_1 & w_1 \\ w_2 & w_2 \end{vmatrix}).$$

Intuitiv:



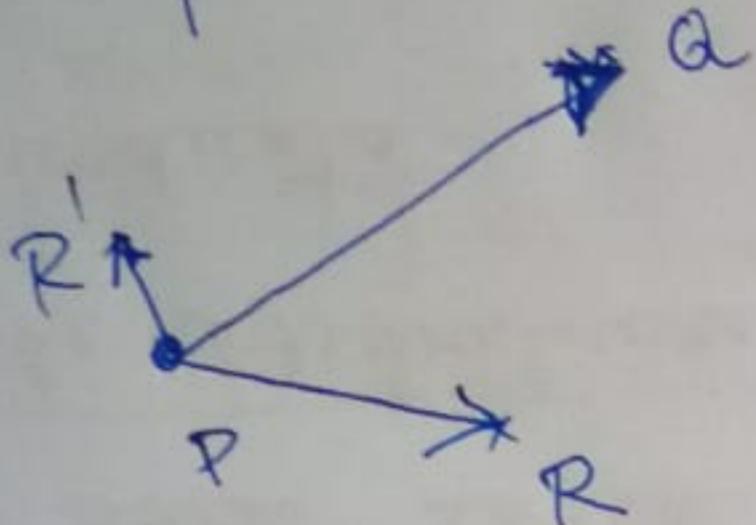
planul orizontal
v, w orizontali

v x w : vertical.

- $\in \mathbb{R}^+$ (abordare numerică)

Problematizare

In plan:



? cantitate numerică
prin care să exprimăm
faptul că "R este în dreapta
lui \vec{PQ} , iar R' este
în stânga lui \vec{PQ} ".

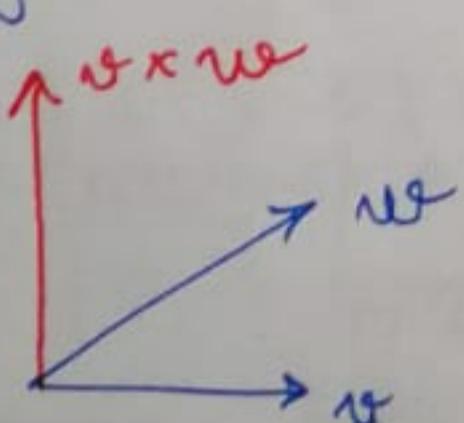
Produs vectorial (cross product)

- geometric

dati v și w necoliniari, produsul vectorial $v \times w$
este un vector:

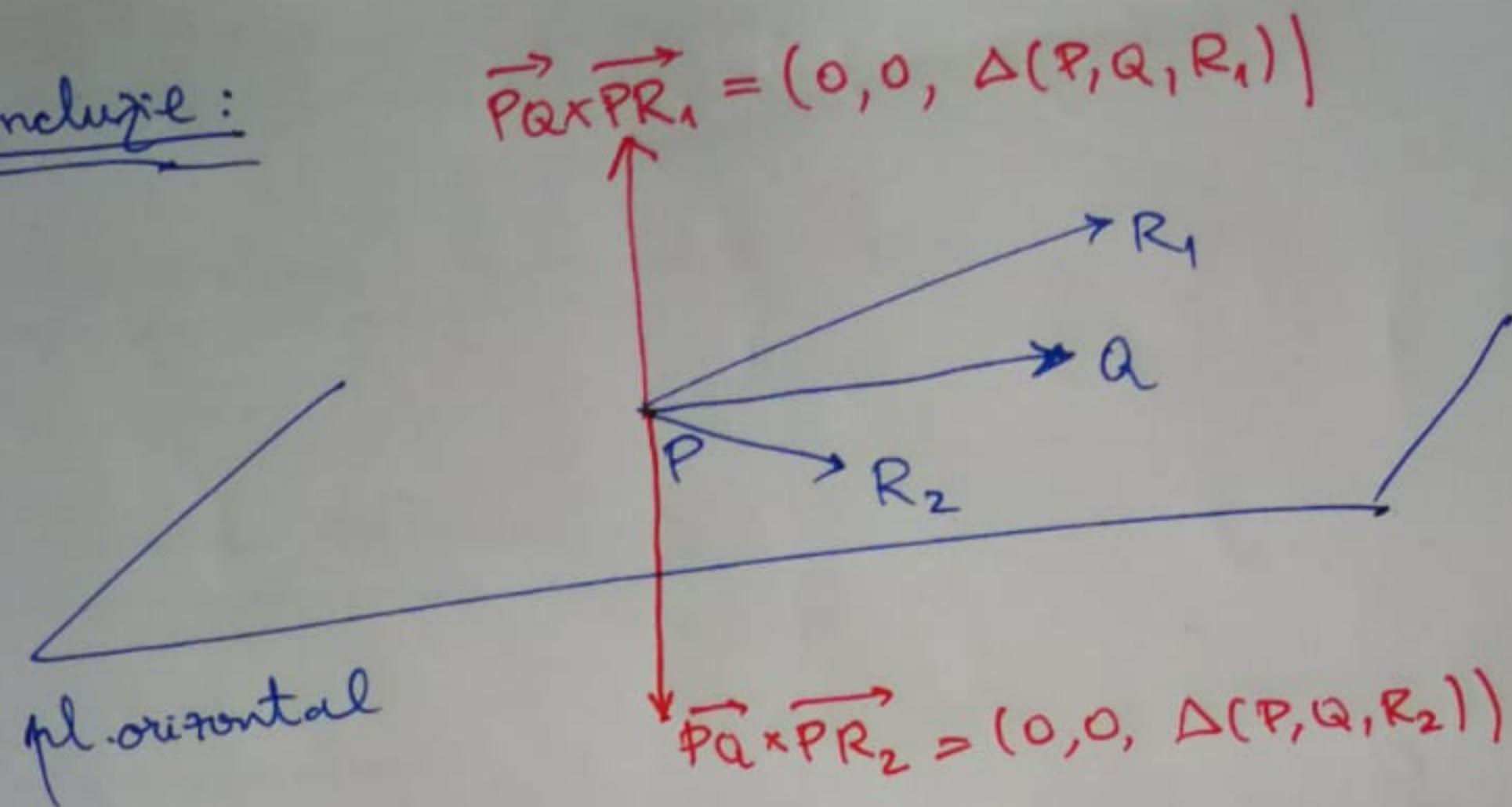
- perpendicular pe v și w ;
- care are sensul dat de regula surubului

(burghieulic) drept:



- are lungimea dată de o anumită
formulă (aria paralelogramului determinat
de v și w) .

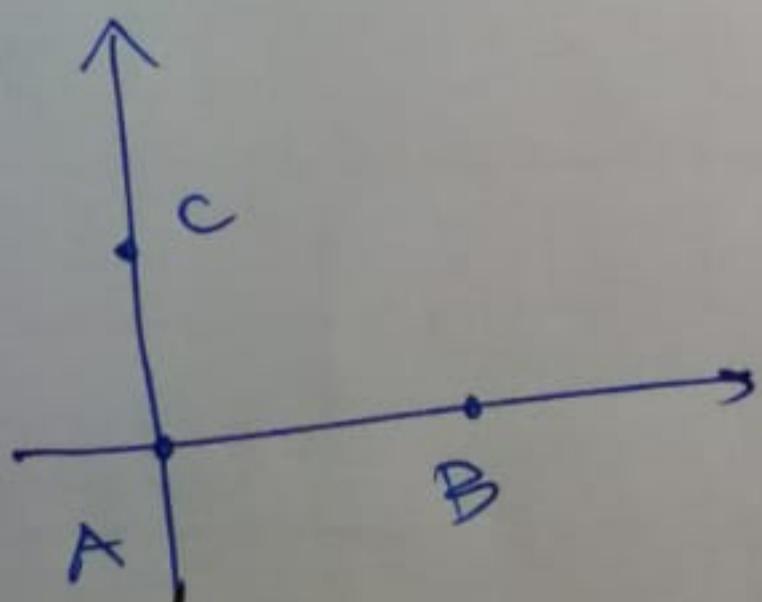
Concluzie:



Pt. R_1 (stânga lui \overrightarrow{PQ}): $\Delta(P, Q, R_1) > 0$.

R_2 (dreapta lui \overrightarrow{PQ}): $\Delta(P, Q, R_2) < 0$.

Exemplu (simpler)



$$A = (0, 0)$$

$$B = (1, 0)$$

$$C = (0, 1)$$

$$\begin{aligned} \Delta(A, B, C) &= \\ &= \begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} = 1 > 0 \end{aligned}$$

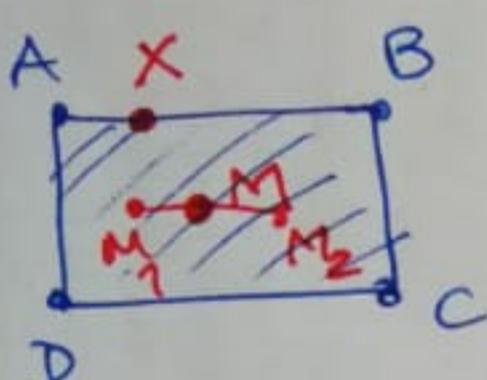
în stânga!

2. Acoperiri conexe

Algoritmi lenti (naivi)

1) Puncte extreme

Exemplul 1



- $M \text{ nu}$ este punct extreム
(găsim M_1, M_2 ca în figura
a.i. $M \in [M_1, M_2]$).
- $X \text{ nu}$ este punct extreム
($x \in [AB]$)
- A, B, C, D sunt puncte
extreム (nu putem găsi P, Q
pe dreptunghi / nu interiorul său
a.i. $A/B/C/D \in [PQ]$).

Exemplul 2 .

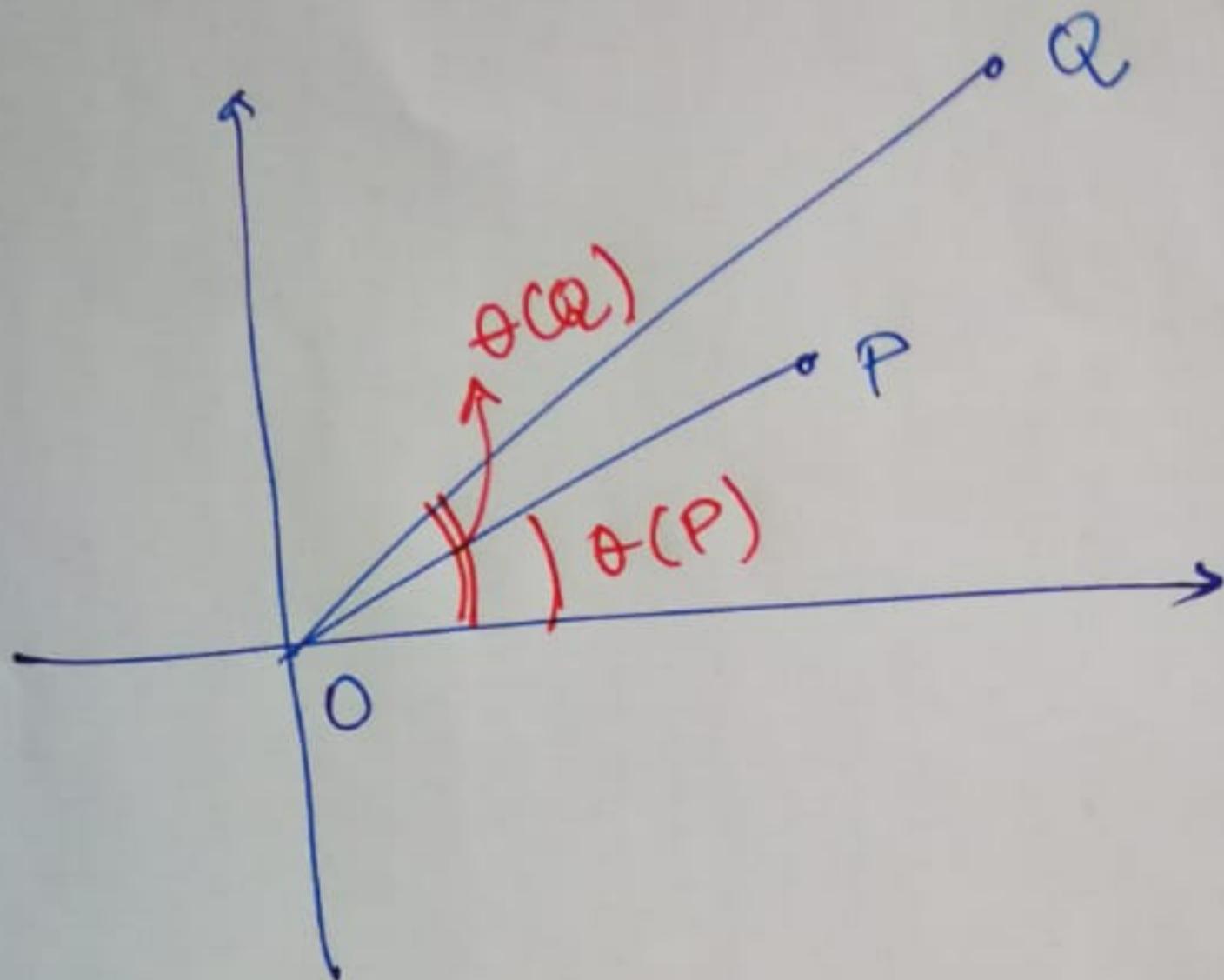


Disc = cerc \cup interiorul său

Toate punctele cercului
sunt extreム !

Comentariu:

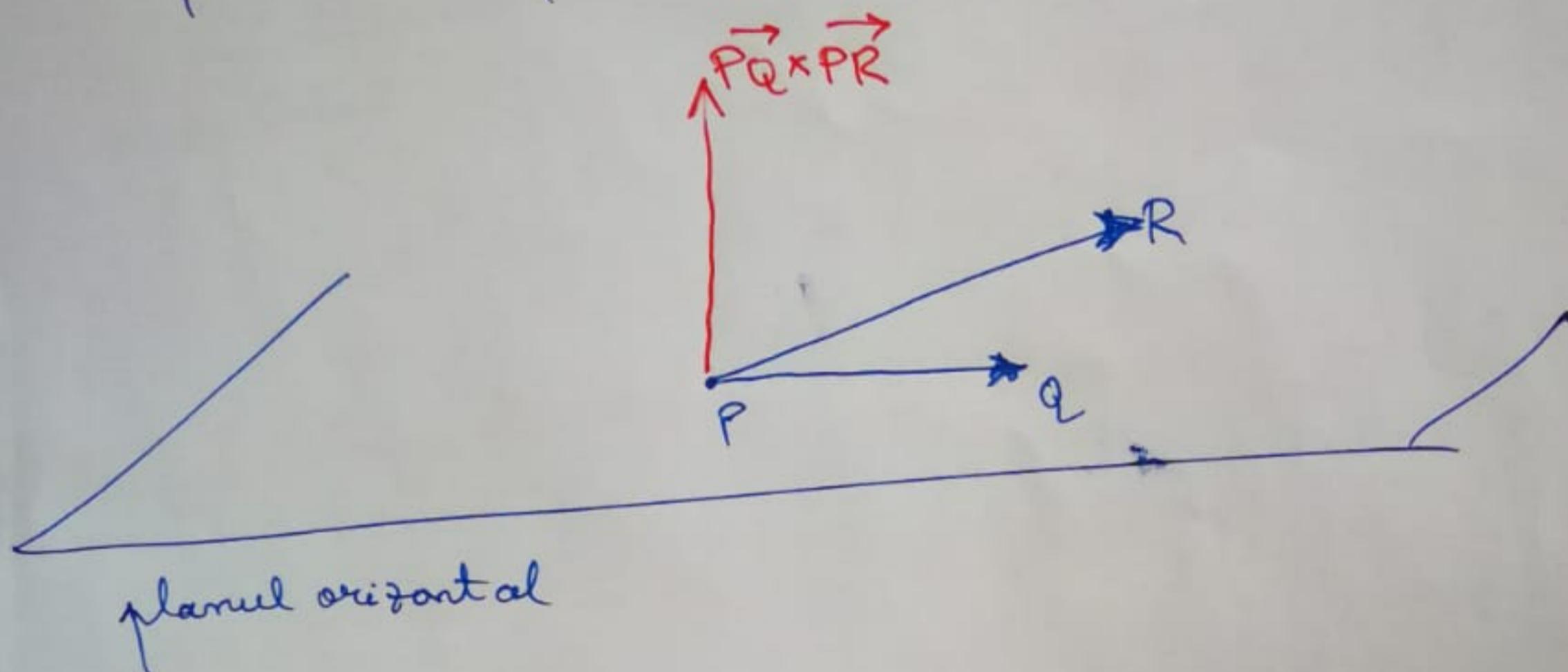
ordonarea după unghiul polar



$\theta(Q) > \theta(P) \Leftrightarrow Q$ este situat la stânga
muchiei orientate \overrightarrow{OP} .
(v-testul de orientare).

Afirmatie (Lemnă din suportul de curs).

Île $P = (P_1, P_2, 0)$; $Q = (q_1, q_2, 0)$; $R = (r_1, r_2, 0)$
(punkte din planul orizontal).



$$\vec{PQ} \times \vec{PR} = (q_1 - P_1, q_2 - P_2, 0) \times (r_1 - P_1, r_2 - P_2, 0) =$$

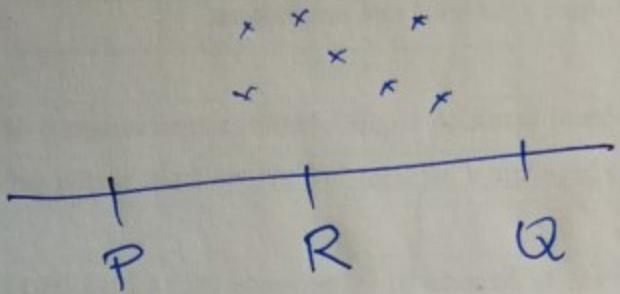
obs.
antrenare $\left(0, 0, \begin{vmatrix} q_1 - P_1 & r_1 - P_1 \\ q_2 - P_2 & r_2 - P_2 \end{vmatrix} \right) =$

$$= \left(0, 0, \begin{vmatrix} 1 & 1 & 1 \\ P_1 & q_1 & r_1 \\ P_2 & q_2 & r_2 \end{vmatrix} \right) =$$

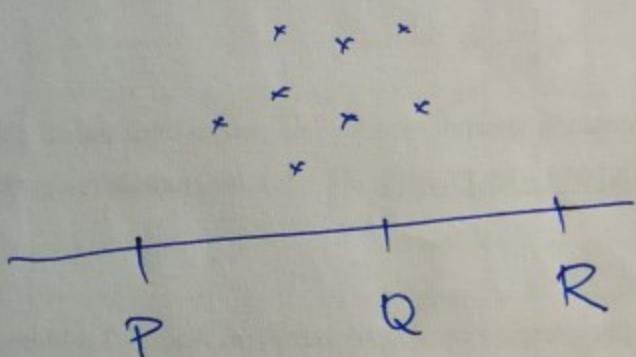
c.f. notatie curs $\left(0, 0, \Delta(P, Q, R) \right)$. 23

Algoritmul lent 2 - Comentarii

tratarea cazurilor degenerate

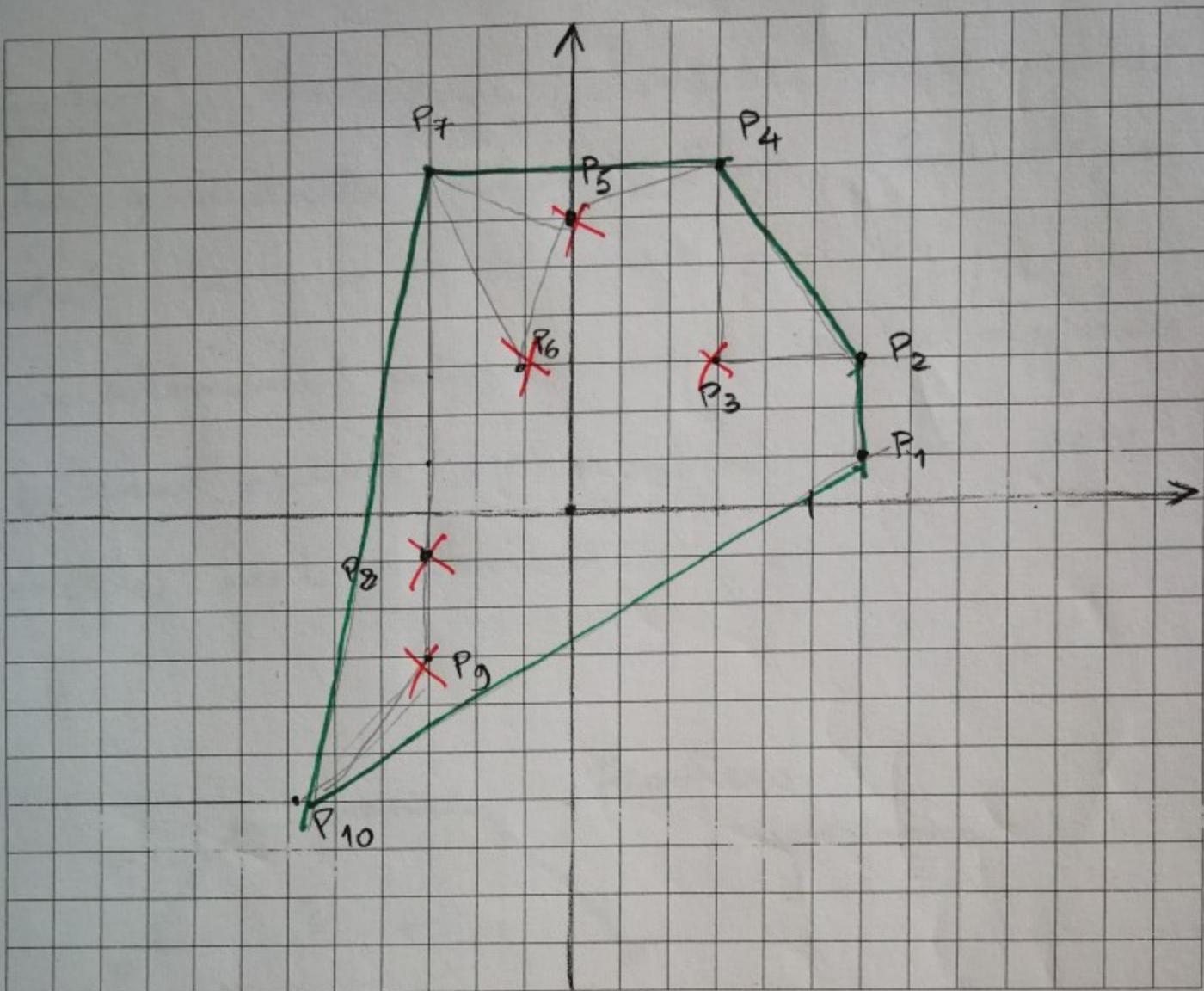


\vec{PQ} este muchie a
frontierei
 $\chi(P, R, Q) > 0$



\vec{PQ} nu este
muchie a
frontierei
 $\chi(P, R, Q) < 0$

Graham's scan



$$P_1 = (6, 1); \quad P_2 = (6, 3); \quad P_3 = (3, 3); \quad P_4 = (3, 7)$$

$$P_5 = (0, 6); \quad P_6 = (-1, 3); \quad P_7 = (-3, 7); \quad P_8 = (-3, -1)$$

$$P_9 = (-3, -3); \quad P_{10} = (-6, -6).$$

$L: P_1, P_2, \cancel{P_3}, \cancel{P_4}, \cancel{P_5}, \cancel{P_6}, \cancel{P_7}, \cancel{P_8}, \cancel{P_9}, P_{10}, P_1$

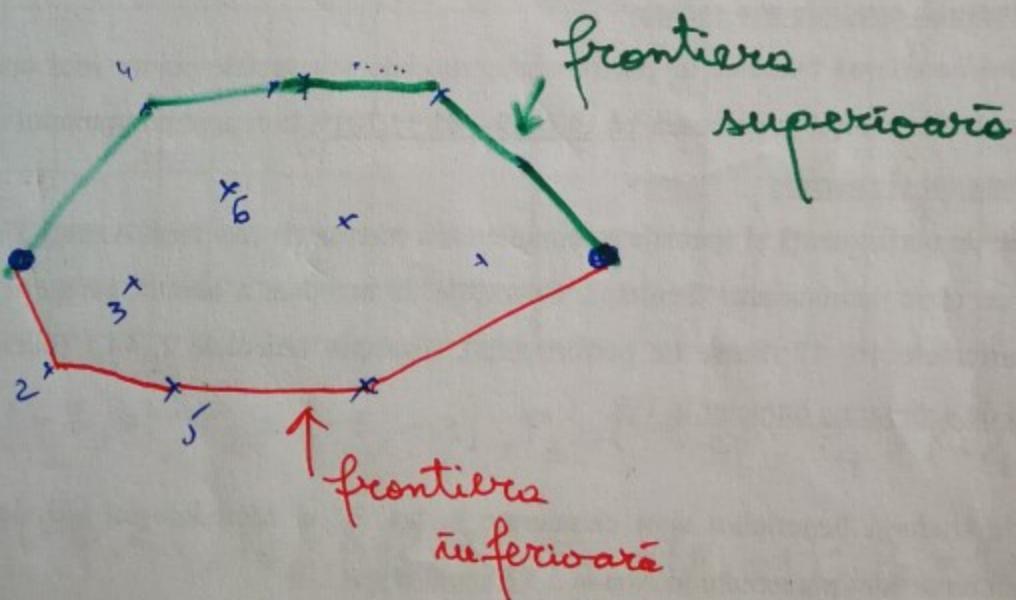
$\boxed{P_1, P_2, P_4, P_7, P_{10}, P_1} \rightarrow$ frontiera acoperirii convexe

(23)

3/2

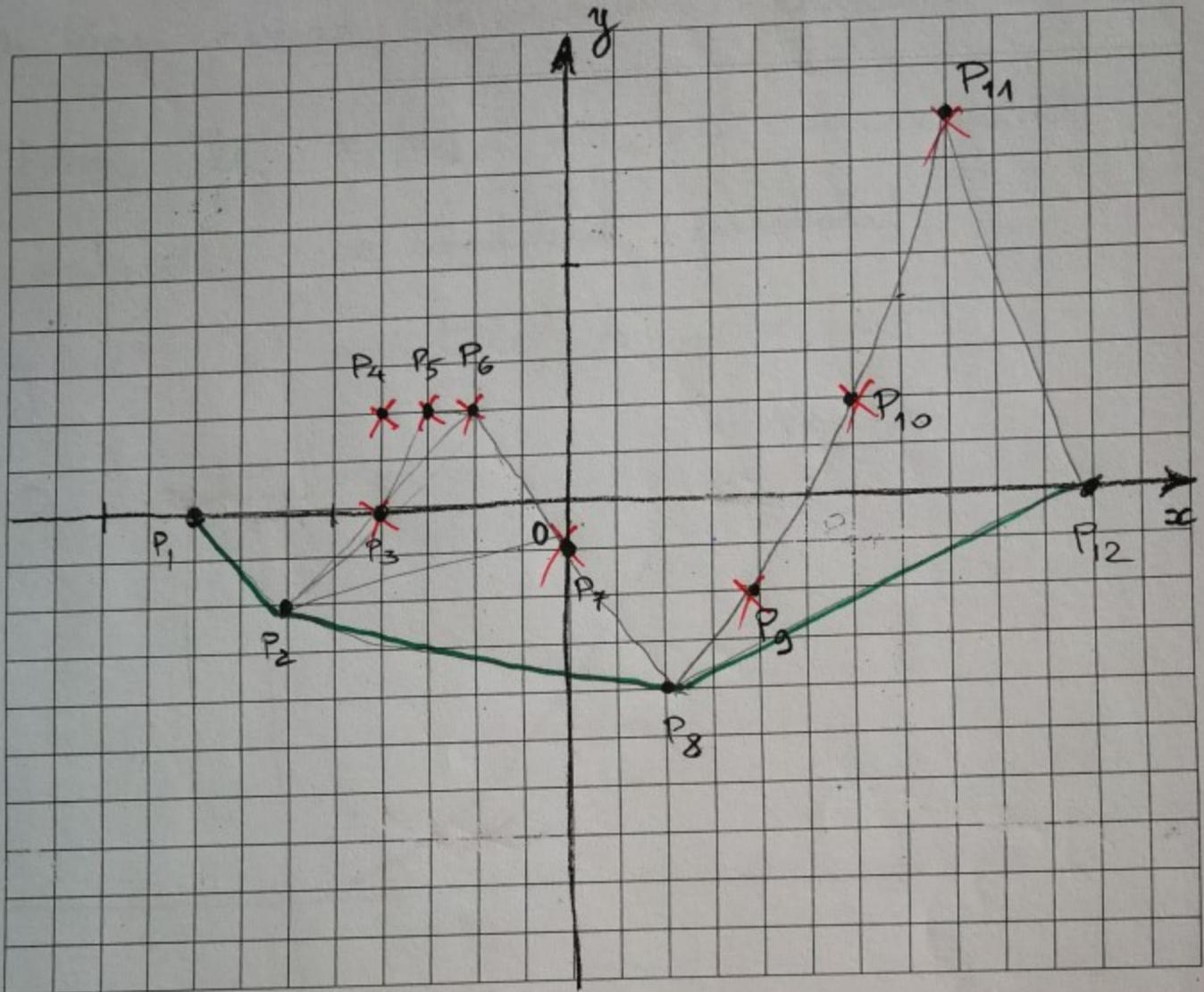
Graham's scan - varianta Andrew.

- sortarea: lexicografic (\cong sortarea după unghiul polar și distanța polară atunci când alegem polul la $-\infty$ în lungul axei Oy)
- se determină "cel mai mic" și "cel mai mare" (în sensul ordinii lexicografice) punct, apoi punctele sunt renumerotate.



- Principiul: același ca la Graham's scan, fiind reținute doar virajele la stânga.

Graham's scan - varianta Andrew



$$P_1 = (-8, 0); P_2 = (-6, -2); P_3 = (-4, 0); P_4 = (-4, 2)$$

$$P_5 = (-3, 2); P_6 = (-2, 2); P_7 = (0, -1); P_8 = (2, -4)$$

$$P_9 = (4, -2); P_{10} = (6, -2); P_{11} = (8, 8); P_{12} = (10, 9)$$

frontiera inf:

~~P₁ P₂ P₃ P₄ P₅ P₆ P₇ P₈ P₉ P₁₀ P₁₁ P₁₂~~

\downarrow P₃ colinear cu P₂ și P₅

$\boxed{P_1 P_2 P_8 P_{12}}$ \rightarrow frontiera inferioară.

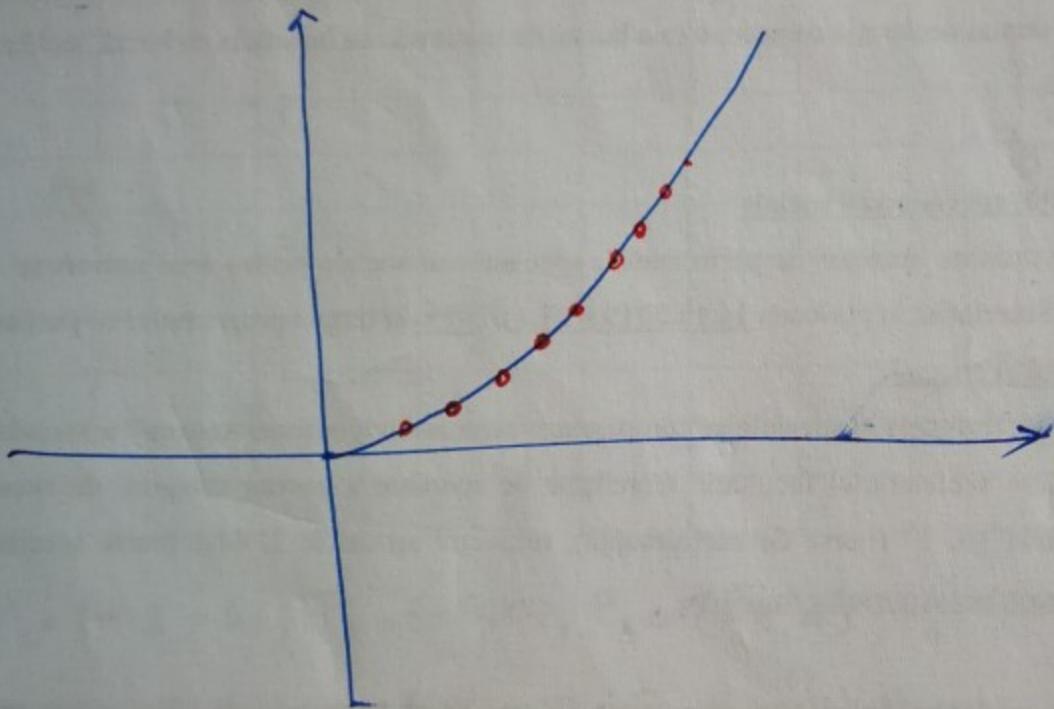
(23)

3/4

Teorema. Problema sortării poate fi transformată,
în timp liniar, în problema acoperirii convexe.

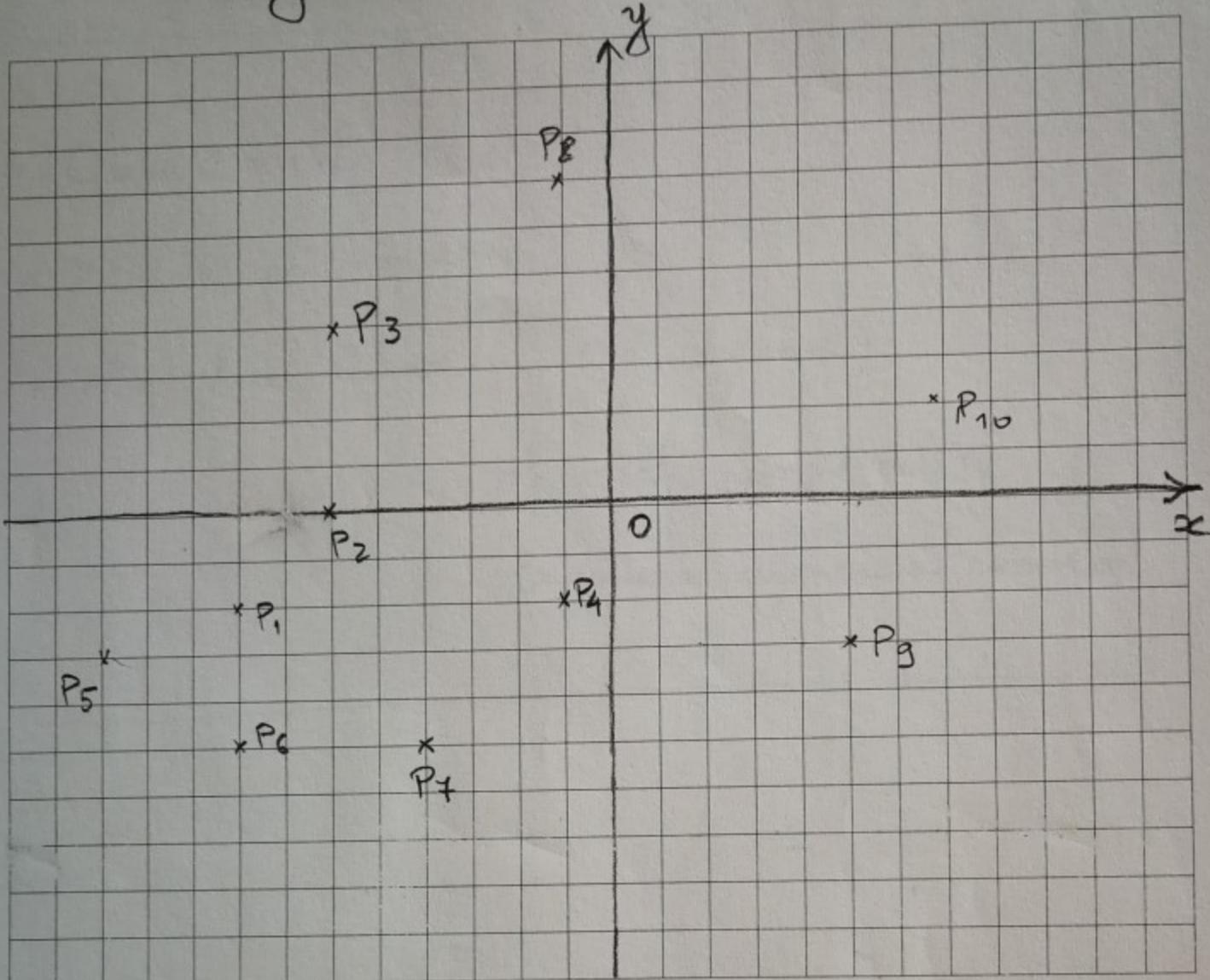
Dem. Fie $x_1, x_2, \dots, x_n > 0$, nu neapărat
ordonate. Se consideră parabola

$y = x^2$ și punctele $A_i = (x_i, x_i^2)$
 $\forall i = \overline{1, n}$.



A sorta numerele x_1, x_2, \dots, x_n este echivalent
cu a determina frontieră acoperirii convexe
pentru $\{A_1, A_2, \dots, A_n\}$.

Jarvis' march



$$P_1 = (-8, -2); P_2 = (-6, 0); P_3 = (-6, 4); P_4 = (-1, -2); P_5 = (-11, -3)$$

$$P_6 = (-8, -5); P_7 = (-4, -5); P_8 = (-1, 7); P_9 = (5, -3); P_{10} = (7, 2)$$

- P_5 cel mai "mic" (lexicografic) - apartine frontierei acoperirii convexe

- determinarea succesorului lui P_5 :

$P_5 P_1 P_2 \rightarrow$ "pivot"
 $P_5 P_1 P_3$
 $P_5 P_1 P_4$
 $P_5 P_4 P_6$
 $P_5 P_6 P_7$
 $P_5 P_6 P_8$
 $P_5 P_6 P_9$
 $P_5 P_6 P_{10}$

P_4 "la dreapta" lui $\overrightarrow{P_5 P_1}$

$\Rightarrow P_6$ este succesorul lui P_5 (23)

3/6

Alți algoritmi:

- Quick Hull
- Divide et Impera
- Alg. lui Chan : $O(n \log h)$

$h = \#$ puncte de pe
frontiera acoperirii convexe.

Aplicație pentru acoperiri convexe

Diametruul unei multimi de puncte

Problema. Date o multime de n puncte din plan, notata cu M , să se determine x și y care sunt cele mai îndepărtate.

(distanța dintre aceste puncte: $\text{diam}(M)$)

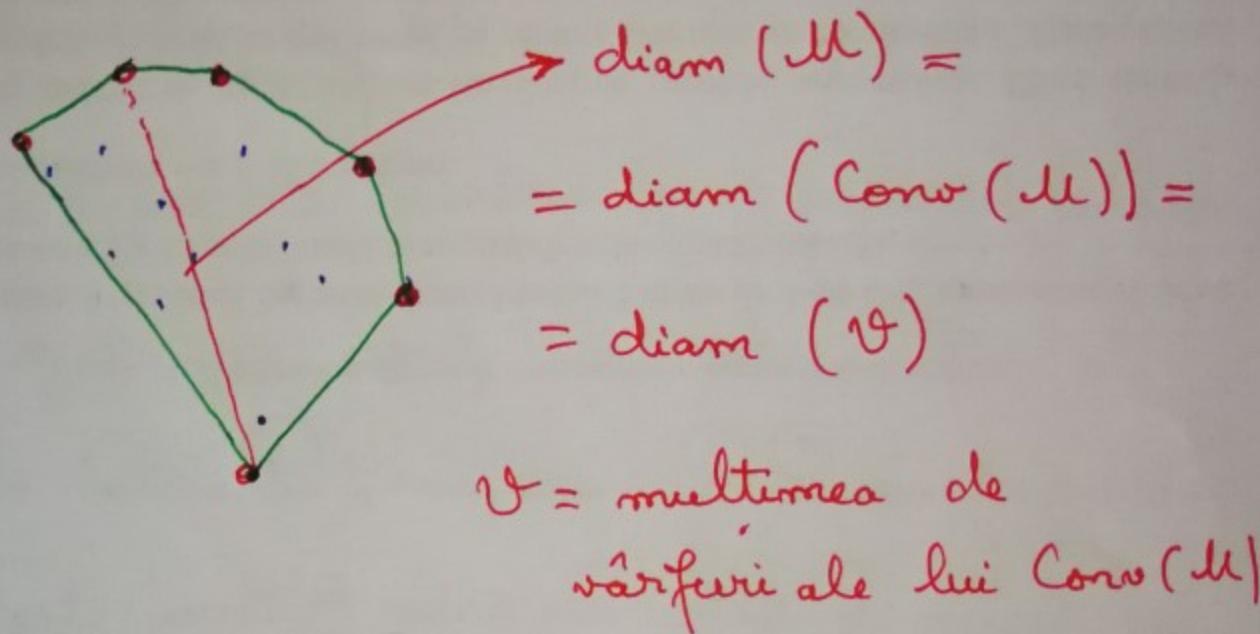
⊗ Abordare 1 (naivă): comparații între perechi de puncte \rightarrow complexitate $O(n^2)$.

⊗ Abordare 2

Rezultate:

Prop. Fie M o multime cu n elemente. Determinarea $\text{diam}(M)$ necesită $\Omega(n \log n)$ (cu \leq) operații.

Prop. Diametrul unei multimi finite de puncte este egal cu diametrul multimii date de varfurile acoperirii sale convexă.

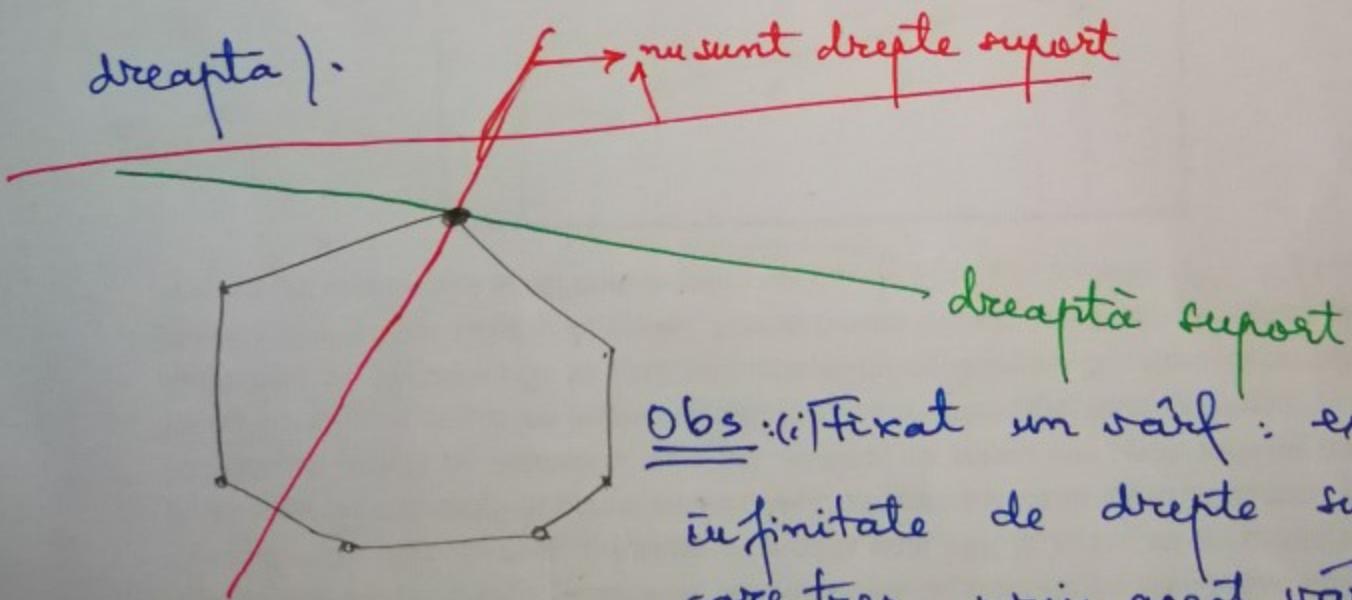


Teorema Diametrul unei poligoane convexe (inclusiv varfurile cele mai indepartate) poate fi determinat in timp liniar.

Corolar : Diametrul unei multimii cu n puncte (inclusiv varfurile care il determină) poate fi determinat cu complexitate $\Theta(n \log n)$.

Idea de dem. a teoremei + algoritm + exemplu

- Diametrul unei multimi de puncte din \mathbb{R}^2 este egal cu cea mai mare distanta dintre toate dreptele suport.
- Dreapta suport pentru un poligon convex: trece printr-un varf al poligonalui si are toate celelalte varfurile de aceasi parte a sa (cel mult intre un varf e situat pe acea dreapta).



Obs: (i) Fixat un varf: există o infinitate de drepte suport care trece prin acest varf.

(ii) Analogie cu tangenta într-un punct al unui cerc C la C.

- Pereche de varfură antipodale: varfură distincte care adună suport paralele.

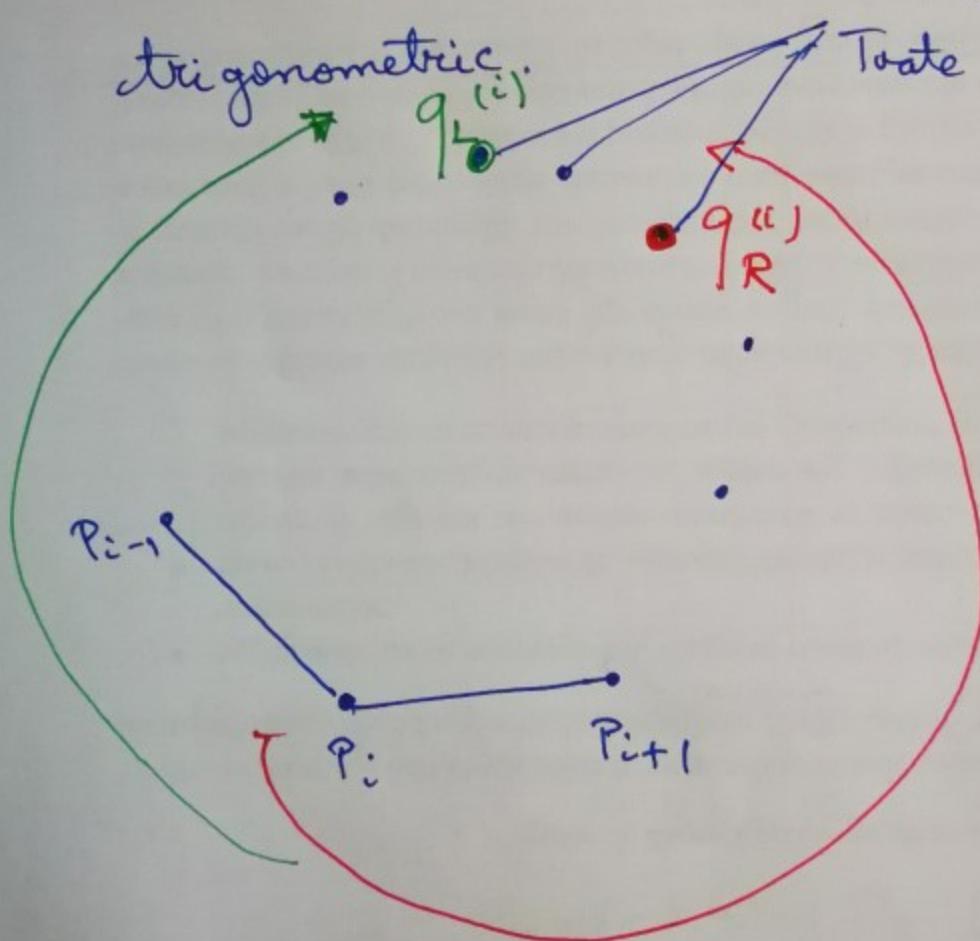
Obs: p, q care dă un diametru \Rightarrow antipodale.

\leftarrow NUT

Idea algoritmului în timp linear pentru determinarea
vârfurilor antipodale

Fixăm un vârf P_i al unui poligon convex P
 și P_{i-1} și P_{i+1} predecesorul / succesorul
 lui P_i , atunci când P este parcurs în sens

trigonometric

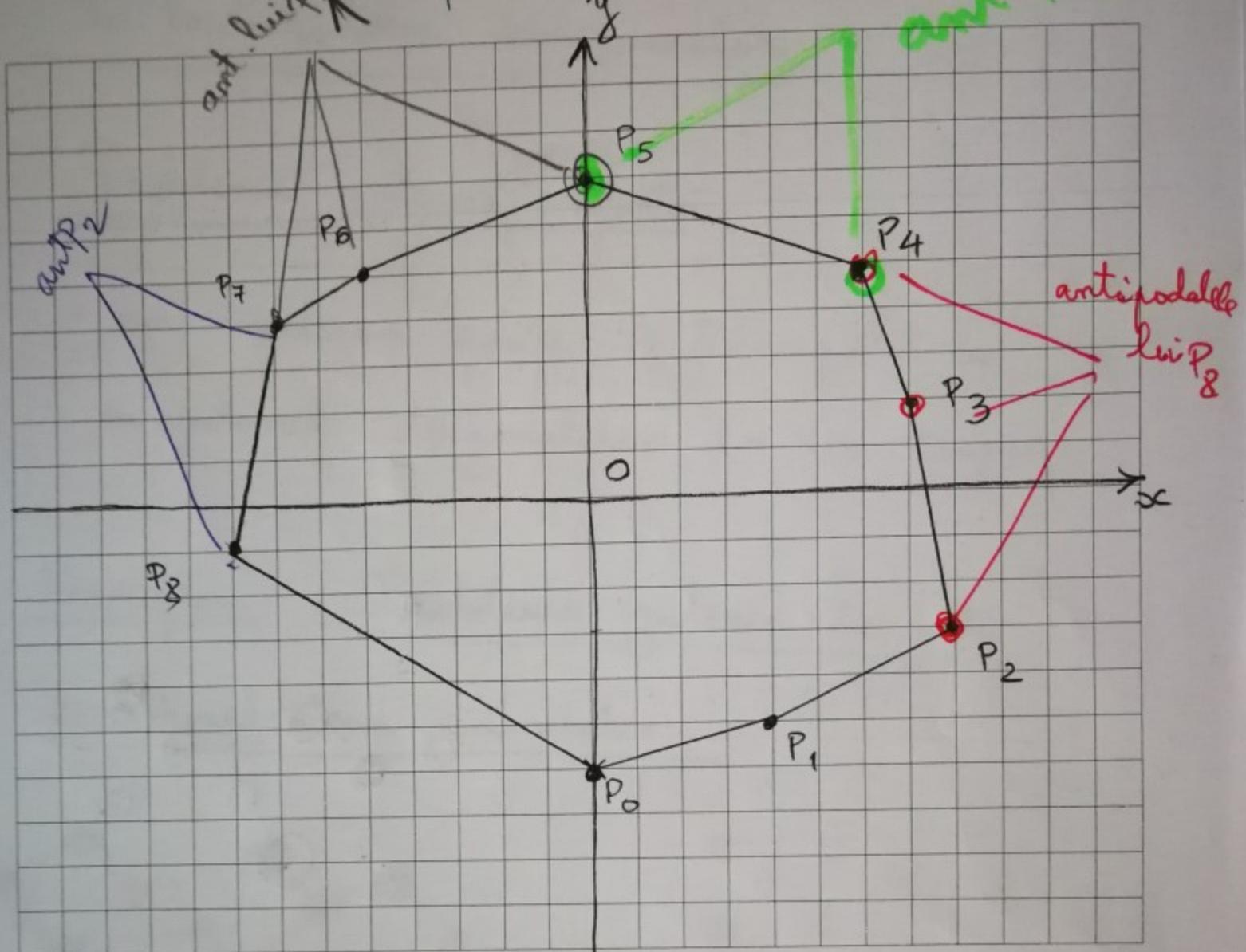


Toate aceste vârfuri
 sunt antipodale
 pentru P_i

$q_L^{(i)}$: cel mai "îndepărtat" de $\overrightarrow{P_i P_{i+1}}$ în sens orar

$q_R^{(i)}$: cel mai "îndepărtat" de $\overrightarrow{P_{i-1} P_i}$ în sens
 trigonometric.

Determinarea punctelor antipodale



$$P_0 = (0, -6); \quad P_1 = (4, -5); \quad P_2 = (8, -3);$$

$$P_3 = (7, 2); \quad P_4 = (6, 5); \quad P_5 = (0, 7);$$

$$P_6 = (-5, 5); \quad P_7 = (-7, 4); \quad P_8 = (-8, -1)$$

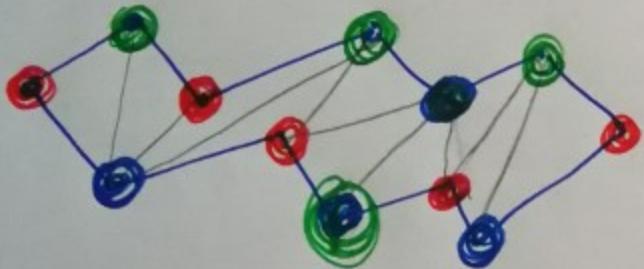
3. Triangularea poligoanelor

Problema galeriei de artă

- un poligon poate fi triangulat cu ajutorul diagonalelor (v. mai târziu).

Exemplu. Problema galeriei de artă

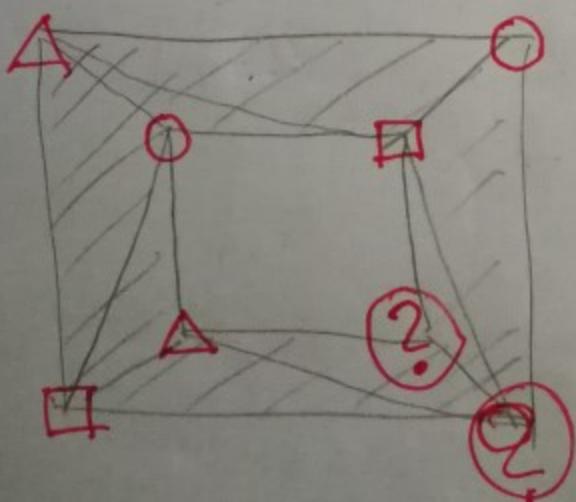
Amplasarea camerelor



- : 5
- : 4
- ◐ : 3

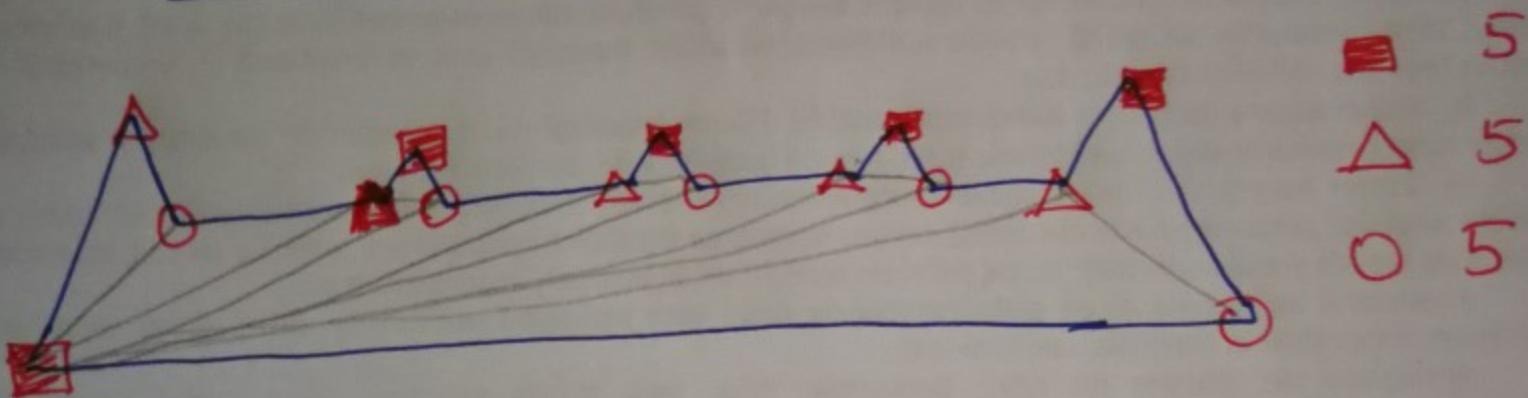
"3-colorare"

Contraexemplu pentru 3-colorare



Teorema Galeriei de Artă

- unori necesare



- intotdeauna suficiente

notăm cu n_1, n_2, n_3 numărul de varfuri

colorate cu cele 3 culori: $n_1 + n_2 + n_3 = n$.

Bp. abs. $n_1 > \left[\frac{n}{3} \right] \Rightarrow n_1 > \frac{n}{3}$ (proprietățile

$$n_2 > \left[\frac{n}{3} \right] \Rightarrow n_2 > \frac{n}{3}$$

$$n_3 > \left[\frac{n}{3} \right] \Rightarrow n_3 > \frac{n}{3}$$

părții
intregi :
partea întreagă

a unui nr. e
cel mai mare
nr. întreg \leq nr.
respectiv)

$n_1 + n_2 + n_3 > n$ contradicție!

\Rightarrow Fie a.i. $n_i \leq \left[\frac{n}{3} \right]$.

Triangulararea poligoanelor (algoritmi)

a) "Ear cutting" / "Ear clipping"

În $P = (P_1, P_2, \dots, P_n)$ un poligon.

Terminologie

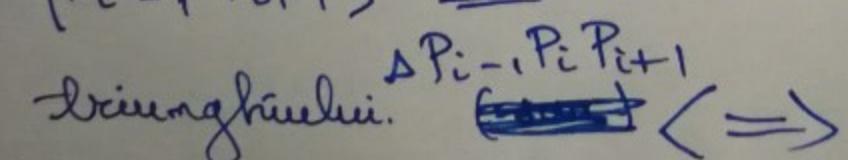
- vârfuri convexe / concave ("reflexe") ale unui poligon \rightarrow verajă \rightarrow s. testul de orientare!

"af. cel mai "mic" lexicografic \rightarrow convex.

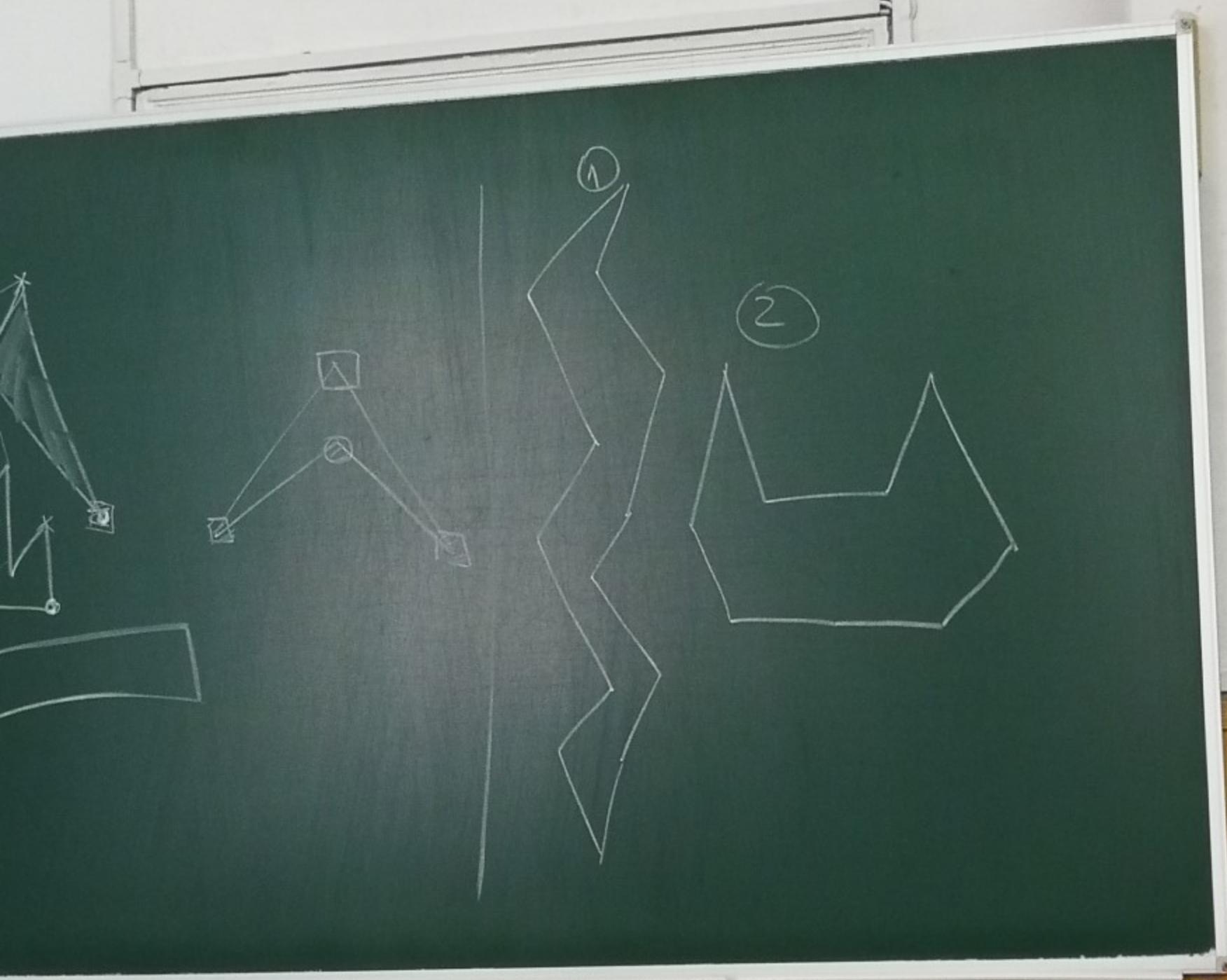
- vârf principal:

P_i este vârf principal dacă $[P_{i-1}, P_{i+1}]$ este diagonală, i.e.

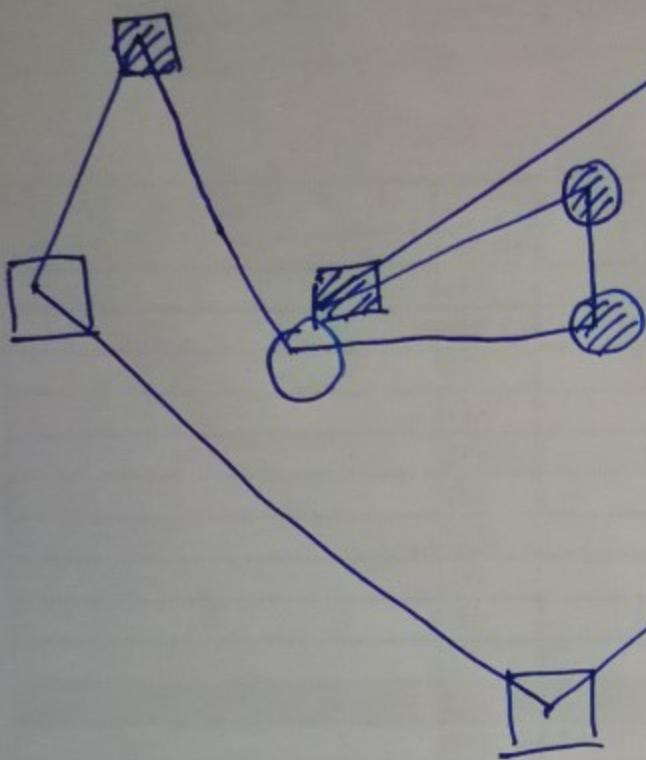
• $[P_{i-1}, P_{i+1}]$ nu intersectă laturile triunghiului $\Delta P_{i-1} P_i P_{i+1}$



• nu există alt vârf în interiorul sau pe laturile $\Delta P_{i-1} P_i P_{i+1}$.



Exemple



~~E~~ → nonprincipal

E

M

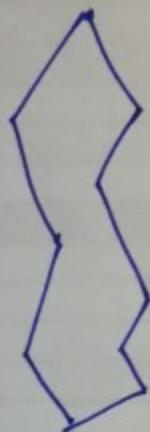
→ f. convex
nonprincipal

→ f. concave
nonprincipal

b) Triangulararea poligoanelor y -monotone

Conceptul de poligon y -monoton

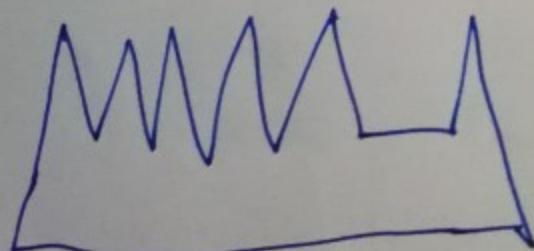
Exemplul 1: Poligon y -monoton



două descrieri:

- poligonul poate fi parcurs de sus în jos - în două moduri (pe două direcții) fără întoarcere în sus;
- orice dreaptă orizontală intersectează poligonul ^{în interiorul} "după o multime conexă" (\emptyset , punct, segment).

Exemplul 2: Poligon care nu este y -monoton



descrieri alternative

→ sunt negate cele
de mai sus:

- nu poate fi parcurs decât cu întoarceri în sus"
- există drepte orizontale care \cap pol. după multimi care nu sunt conexe.

Convenim că "există" laturile orizontale.

Triangulararea poligoanelor y-monotone

- paradigma dreptei / liniei de baleiere
(line sweep)
- în acest caz: dreapta este orizontală.

- **"Statut"** al dreptei de baleiere: stivă a vârfurilor deja întâlnite, dar care "mai au nevoie de diagonale" / "mai pot să apară în triunghiuri".

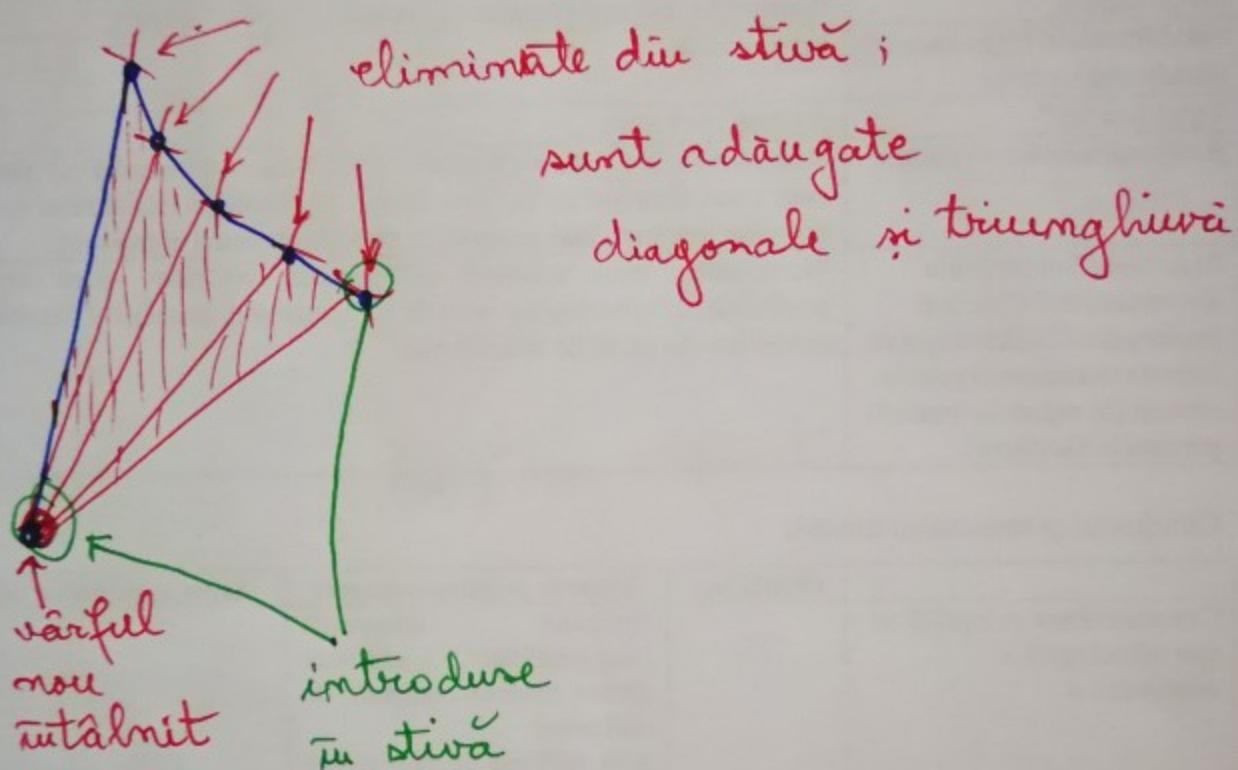
(Clarificare: când este eliminat un vârf?: când a fost traseată o diagonală situată "mai jîs de acela".)

- **"Evenimente"**: modificarea statutului vârfurile poligonului, în prealabil ordonate după y ; pentru fiecare vârf suntem dacă este pe lantul din stînga sau pe cel din dreapta.
- **"Invariant"**: "formă de pâlnie". În core:
 - vârful de sus: convex
 - pe o parte: o muchie
 - cealaltă parte: muchie/succesiune de vârfuri concave

Ce se întâmplă la evenimente?

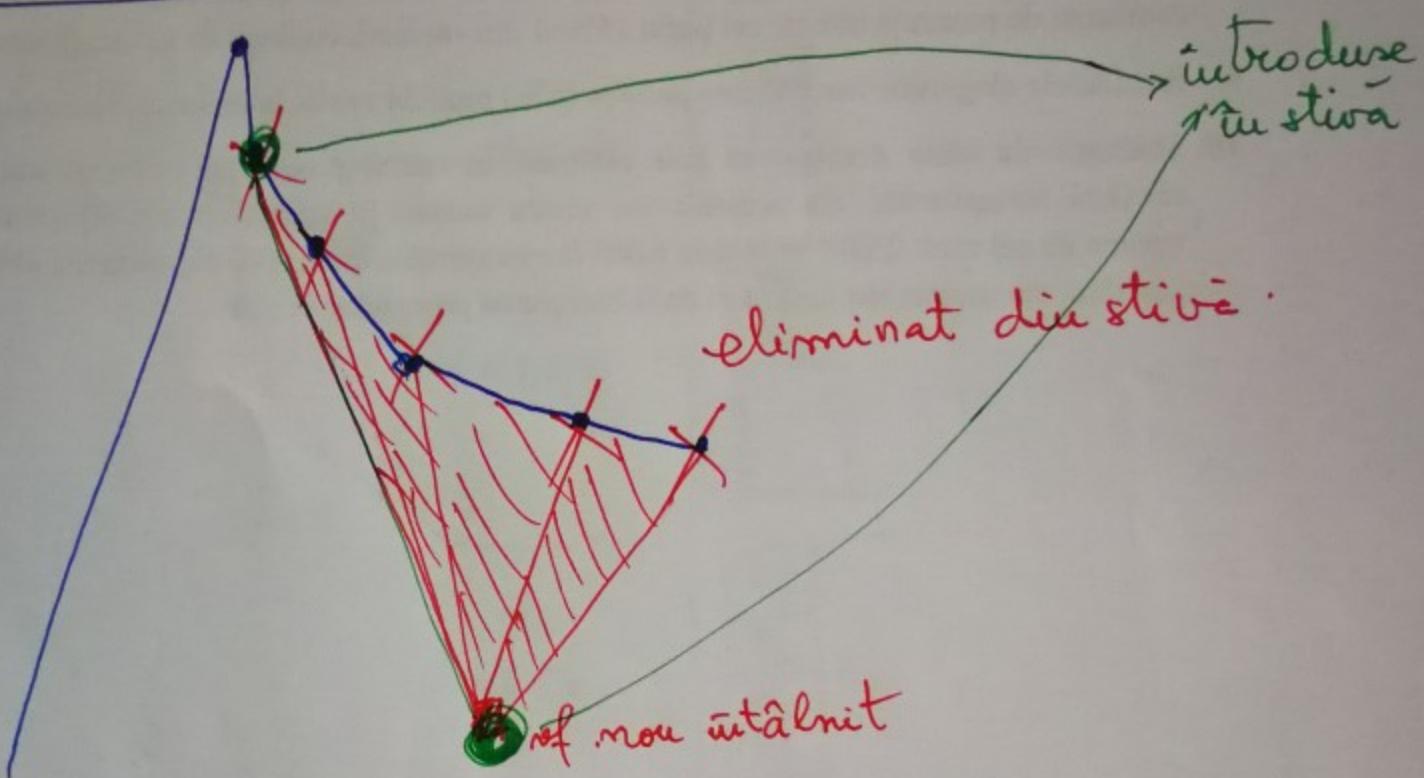
(când întâlnim un nou vîrf).

Cazul 1. Vîrful nou întâlnit este pe lantul open
ultimului vîrf din stivă.

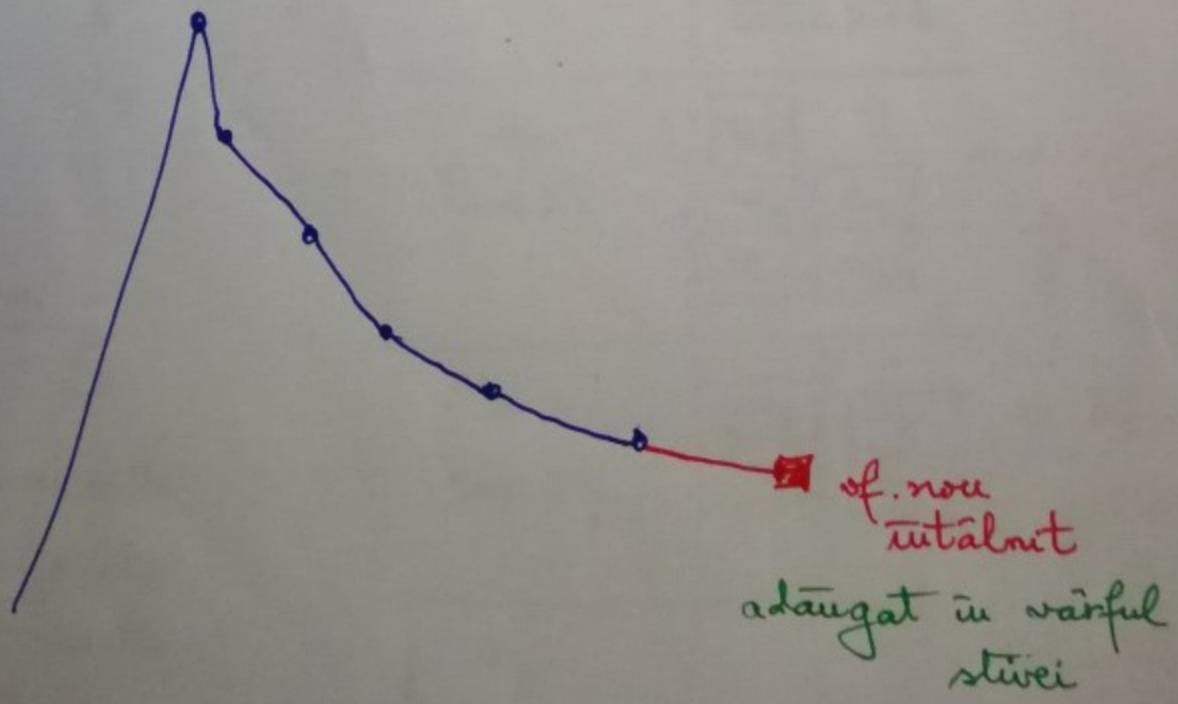


Cazul 2 sf. nou întâlnit: pe același lant.

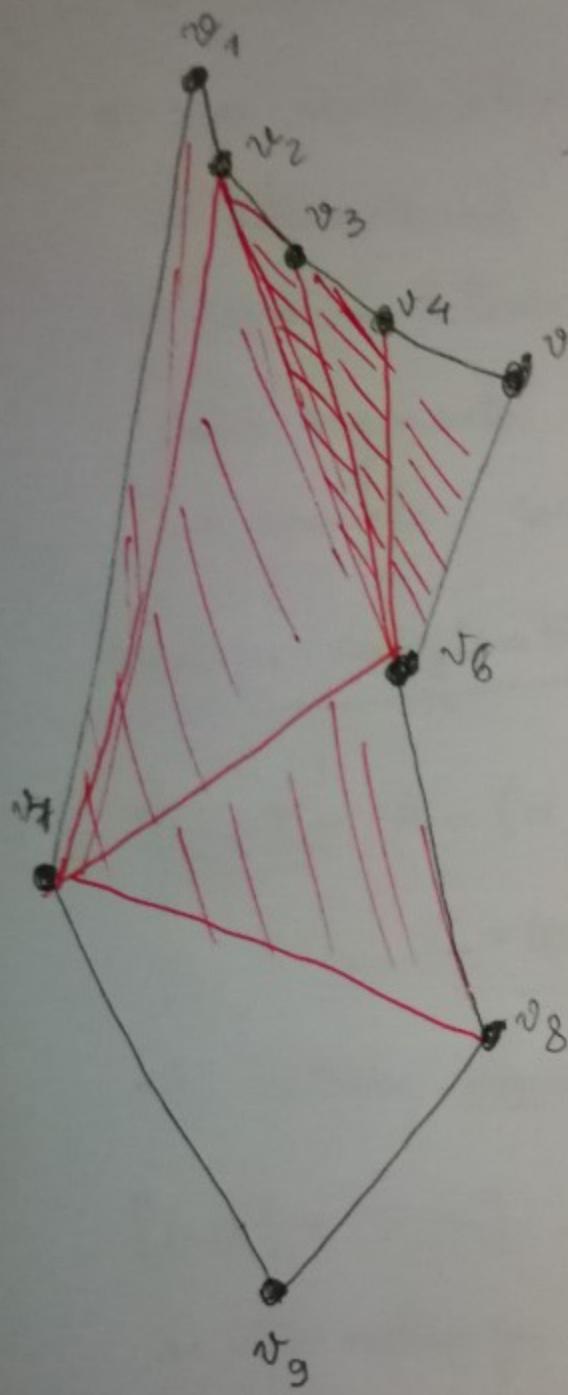
varianta a)



varianta b).



Exemplu (Triangulararea poligoanelor monotone)



Eveniment

	Stiva
v_3 (caz 2b)	v_2 v_1
v_4 (caz 2b)	v_3 v_2 v_1
v_5 (caz 2b)	v_4 v_3 v_2 v_1
v_6 (caz 2a)	v_5 v_4 v_3 v_2 v_1
v_7 (caz 1)	$\cancel{v_5}$ $\cancel{v_4}$ $\cancel{v_3}$ $\cancel{v_2}$ v_1
v_8 (caz 1')	$\cancel{v_6}$ $\cancel{v_2}$ v_7 v_6
v_9	—

+ diagonale
și triunghiuri

+ diagonale
și triunghiuri

+ triunghiuri
diagonale

Exercitii

Scrieți la fiecare pas ce diagonale au fost trase și ce Δ au fost eliminate.

4. Intersecție de segmente ($\subset \mathbb{R}^2$)

- a) cum se stabilește dacă două segmente se intersectează?
- b) cum se determină punctul de intersecție dintre două segmente (se presupune că există...)
- La ambele: care este complexitatea algebrică?

Analiza complexității algebrice

Input: $A = (x_A, y_A); B = (x_B, y_B);$

$C = (x_C, y_C); D = (x_D, y_D).$

a) Testăm dacă se intersectează:

$[AB] \cap [CD]$ nu sunt inclusă în aceeași dreaptă
se intersectează $\Leftrightarrow A$ și B sunt de o
parte și de alta a lui $\frac{[C, D]}{CD}$ și C și D

sunt de o parte și de alta a lui $\frac{[AB]}{AB}$.

Aceasta revine la testul de orientare

\rightarrow polinom de gradul II.

\Rightarrow complexitate algebrică polinom de gradul II.

Obs. Cum procedăm dacă segmentele sunt situate
pe aceeași dreaptă?

b) Determinarea explicită a punctului de intersecție dintre segmentele $[AB]$ și $[CD]$

$$M \in [AB] \Leftrightarrow \exists \lambda \in [0,1] \text{ a.i. } M = (1-\lambda)A + \lambda B$$

$$(M \in AB \Leftrightarrow \exists \lambda \in \mathbb{R} \text{ a.i. } M = (1-\lambda)A + \lambda B)$$

$$N \in [CD] \Leftrightarrow \exists \mu \in [0,1] \text{ a.i. } N = (1-\mu)C + \mu D$$

$$(N \in CD \Leftrightarrow \exists \mu \in \mathbb{R} \text{ a.i. } N = (1-\mu)C + \mu D).$$

$$M \in [AB] \cap [CD] \Leftrightarrow \exists \lambda, \mu \in [0,1] \text{ a.i.}$$

$$\underline{\underline{M = (1-\lambda)A + \lambda B = (1-\mu)C + \mu D}}$$

In coordonate:

$$\begin{cases} x_M = (1-\lambda)x_A + \lambda x_B = (1-\mu)x_C + \mu x_D \\ y_M = (1-\lambda)y_A + \lambda y_B = (1-\mu)y_C + \mu y_D \end{cases}$$

Sistem de 2 ecuații cu 2 necunoscute

$$(\lambda, \mu).$$

$\Delta x_A, y_A, x_B, y_B, x_C, y_C, x_D, y_D$: cunoscute.

Sistemul :

$$(*) \begin{cases} (x_B - x_A) \lambda + (x_D - x_C) \mu = x_C - x_A \\ (y_B - y_A) \lambda + (y_D - y_C) \mu = y_C - y_A \end{cases}$$

$$\Delta = \begin{vmatrix} x_B - x_A & x_D - x_C \\ y_B - y_A & y_D - y_C \end{vmatrix} \rightarrow \begin{array}{l} \text{polinom} \\ \text{de gradul II.} \end{array}$$

Pp. $\Delta \neq 0$:

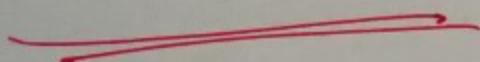
$$\lambda = \frac{1}{\Delta} \begin{vmatrix} x_c - x_A & x_D - x_C \\ y_c - y_A & y_D - y_C \end{vmatrix}$$

analog $\mu \Rightarrow$

calculul soluțiilor λ și μ ale sistemului (*)

revine la evaluarea unor rapoarte de forma

$$\frac{\text{Pol } \underline{\text{II}}}{\text{Pol } \underline{\text{II}}}.$$



In final: prin înlocuire

$$\frac{\text{Pol } \underline{\text{III}}}{\text{Pol } \underline{\text{II}}}.$$



Analiza complexității algebrice - completări

Obs. (cu notatiile din cursul anterior)

(ii) Dreptele AB și CD se intersectează
într-un singur punct $\Leftrightarrow \Delta \neq 0$.

(iii) Pp. $\Delta \neq 0$. Cele două segmente se intersectează
 $\Leftrightarrow 0 \leq \lambda, \mu \leq 1$.
 $(\lambda, \mu \in [0, 1])$.

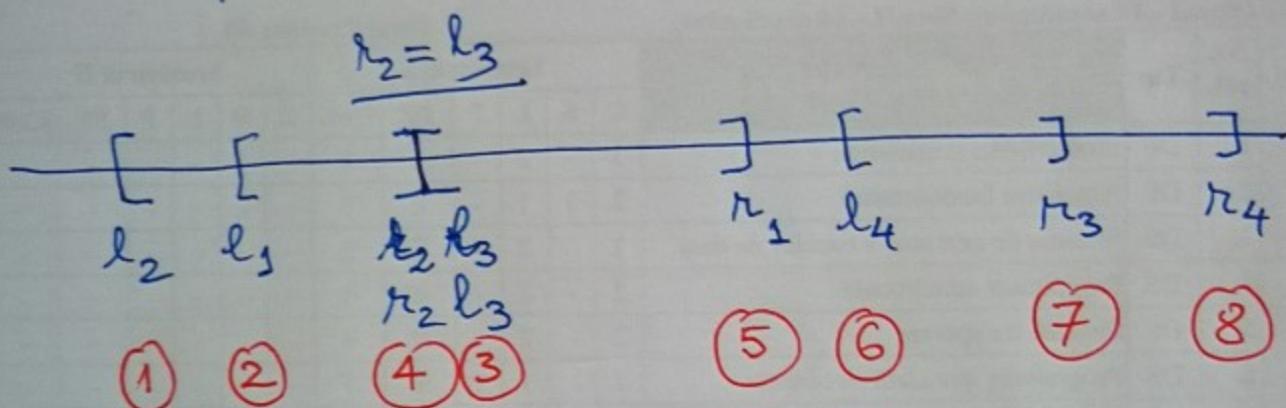
(iv) Pentru a stabili dacă două segmente se
intersectează abordând folosind λ și μ nu
este necesar să le calculăm efectiv, este
suficient să efectuăm comparații de polinoame.

Exemplu : Var. 1 Statutul = multime

Rezolvarea problemei intersecției de segmente
în context 1D

Segmentele $[l_i, r_i], i = \overline{1, n}$ sunt
parcurse după ordonarea extremităților.

Exemplu .



Comentariu : aceasta ordonare corespunde ordonării
lexicograficea multimii:

$$\{(l_i, 0) \mid i = \overline{1, n}\} \cup \{(r_i, 1) \mid i = \overline{1, n}\}.$$

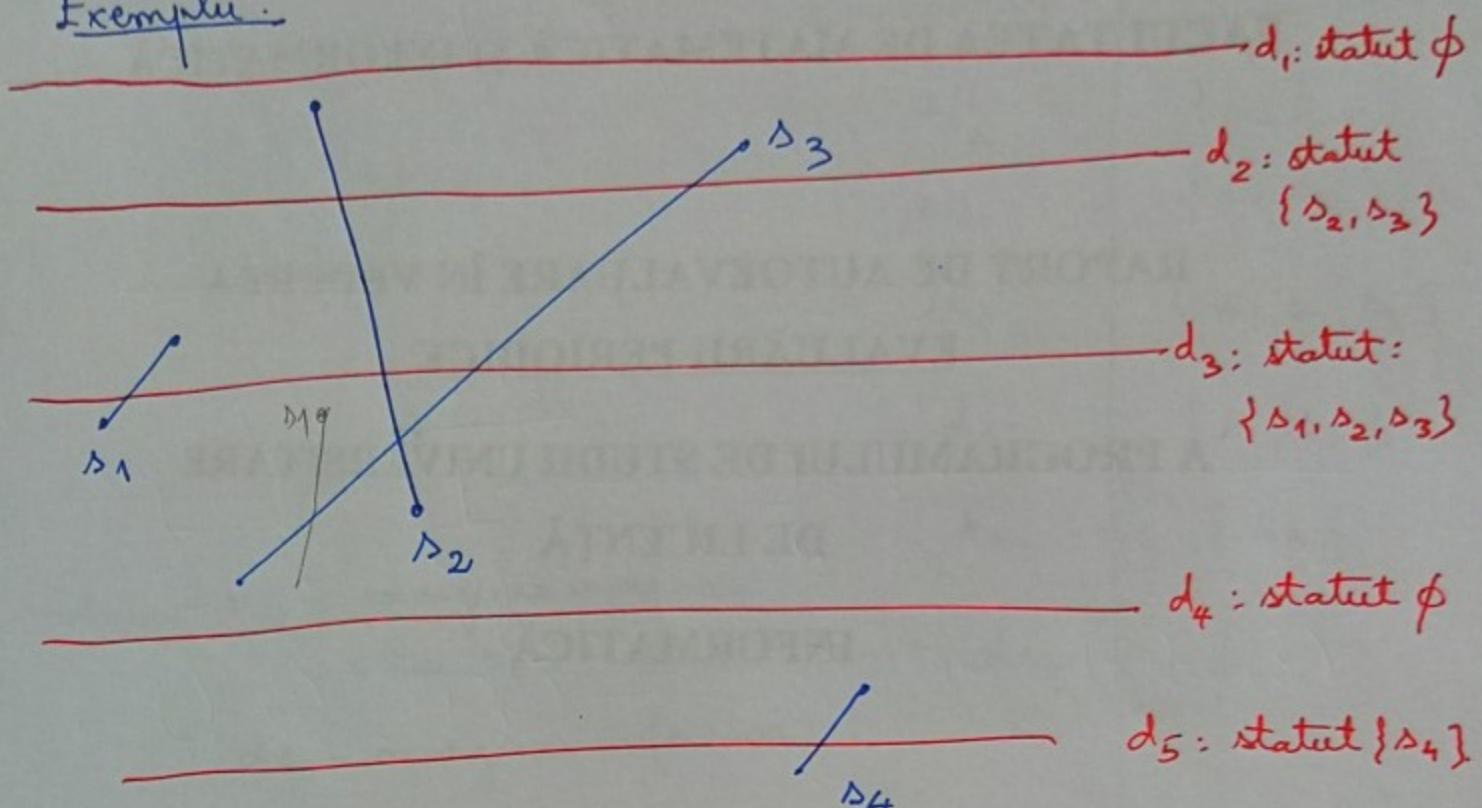
Mecanism de tip "baliere".

Statut : multimea segmentelor "active".

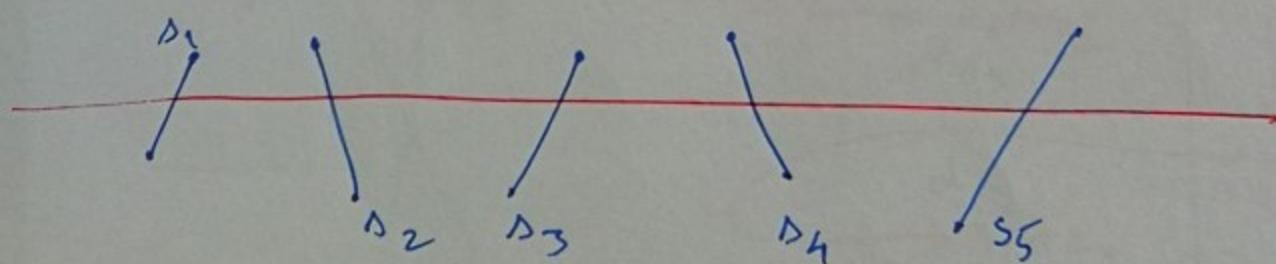
Evenimente : extremitățile segmentelor.

Intersecție de segmente (context 2D)

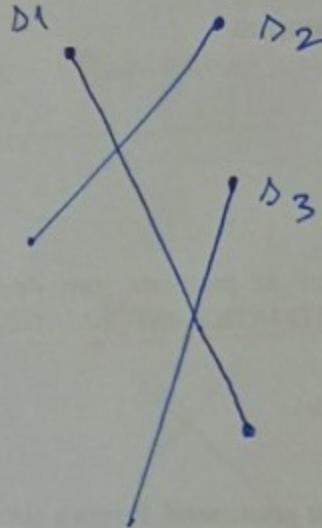
Exemplu:



Încă insuficientă:



Exemplu . Var. 1 Statutul = multime

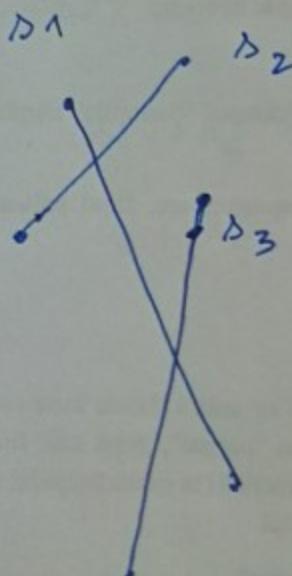


Eveniment	Statut
μ_{s_2}	\emptyset
μ_{s_1}	$\{s_2\}$
μ_{s_3}	$\{s_1, s_2\}$
d_{s_2}	$\{s_1, s_2\}$
d_{s_1}	$\{s_2\}$
d_{s_3}	\emptyset .

Notă: μ_{s_i} = marginea superioară
a lui s_i

d_{s_i} = marginea inferioară
a lui s_i

Exemplu Var. 2. Statutul = listă (multime ordonată)



Eveniment	statut
μ_{s_2}	\emptyset
μ_{s_1}	(s_2)
μ_{s_3}	(s_1, s_2) $\xrightarrow{s_1 \cap s_2 ?}$
d_{s_2}	(s_2)
d_{s_1}	(s_1, s_2)
d_{s_3}	(s_2, s_1, s_3) $\xrightarrow{s_1 \cap s_3 ?}$
	(s_1, s_3)
	(s_3, s_1)
	(s_3)
	\emptyset .

inserat

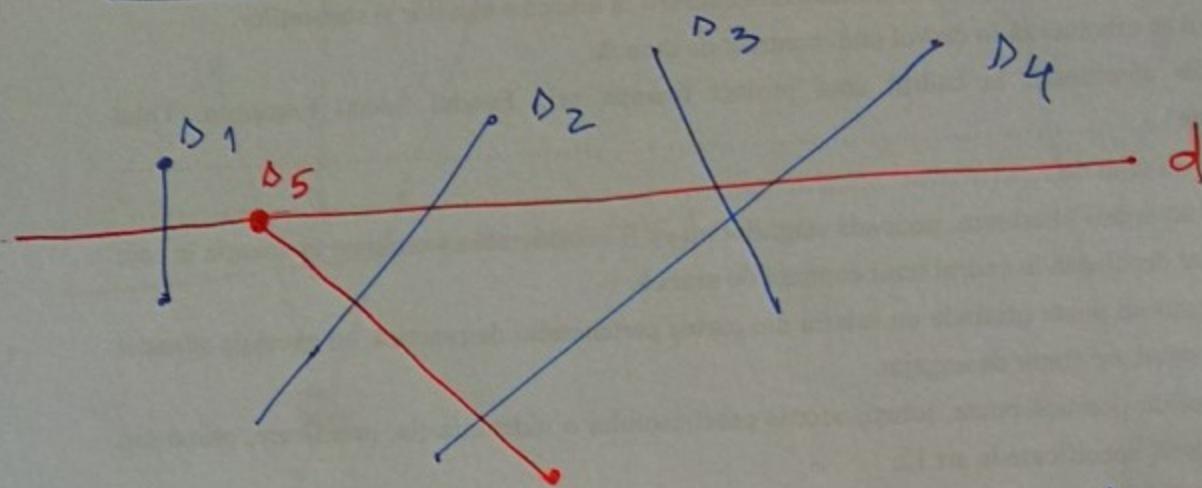
$\xrightarrow{s_1 \cap s_2 ?}$

$\xrightarrow{s_1 \cap s_3 ?}$

$\xrightarrow{s_1 \cap s_3 ?}$

Tipuri de evenimente

a) Margine superioară:



$(\Delta_1, \Delta_2, \Delta_3, \Delta_4)$.

Evenimentul μ_{Δ_5} :

$\downarrow \mu_{\Delta_5}$

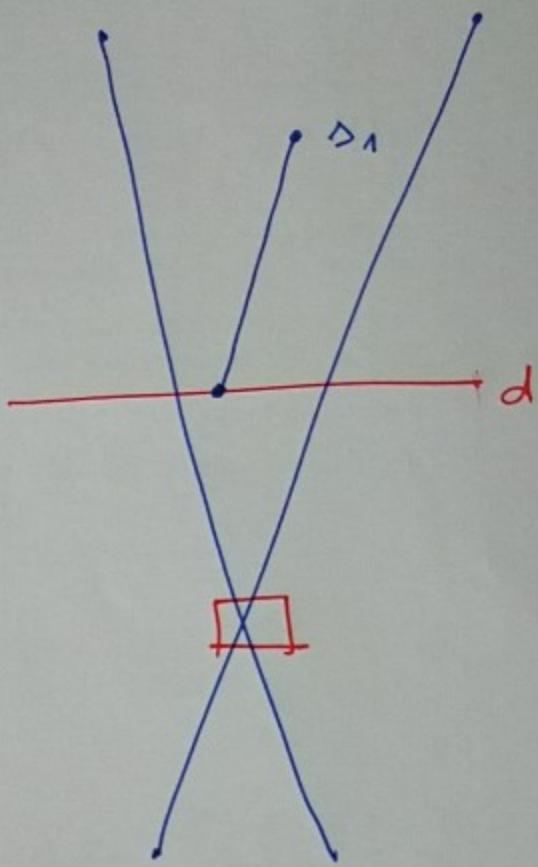
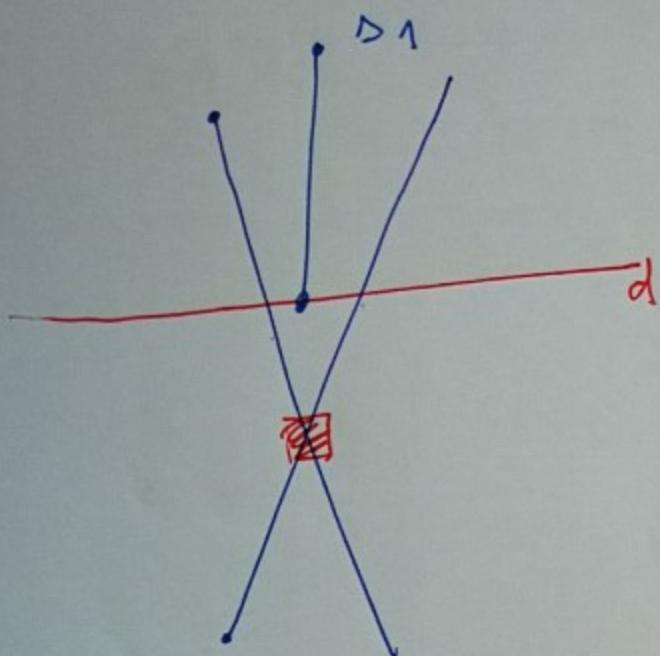
$(\Delta_1, \underline{\Delta_5}, \Delta_2, \Delta_3, \Delta_4)$

- teste de intersectie cu vecinii nou aparati.

$\Delta_1 \cap \Delta_5 \rightarrow$ nu se intersecteaza

$\Delta_2 \cap \Delta_5 \rightarrow$ se intersecteaza, inserat un nou eveniment $\Delta_2 \cap \Delta_5$.

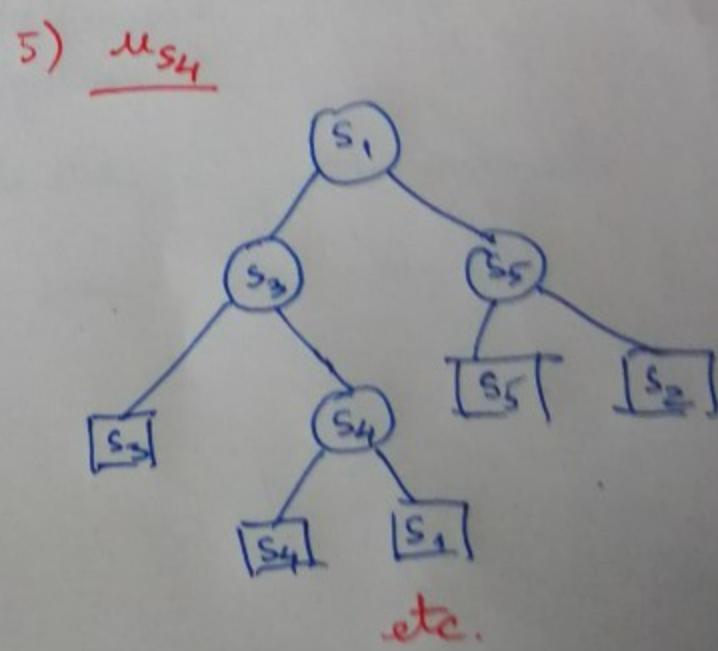
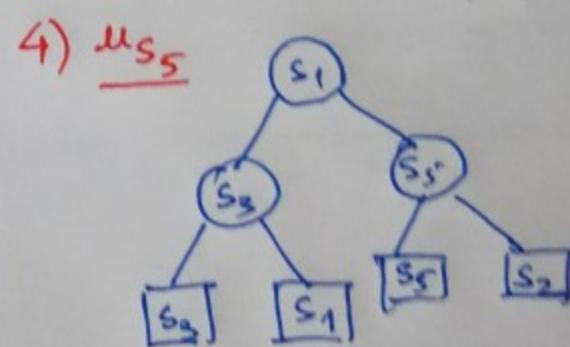
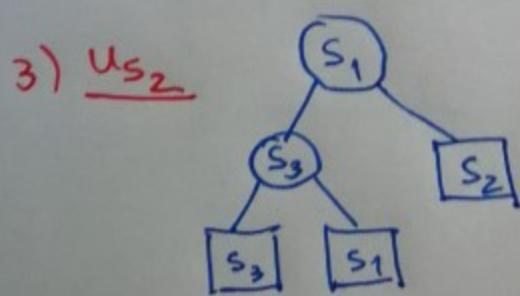
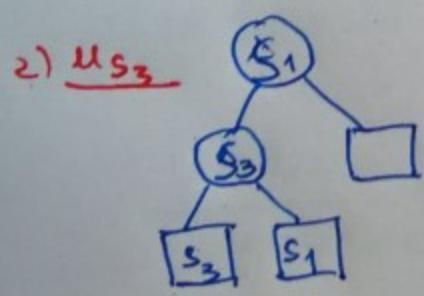
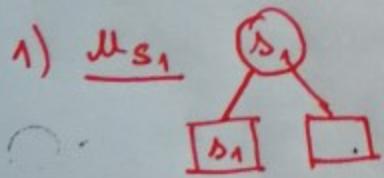
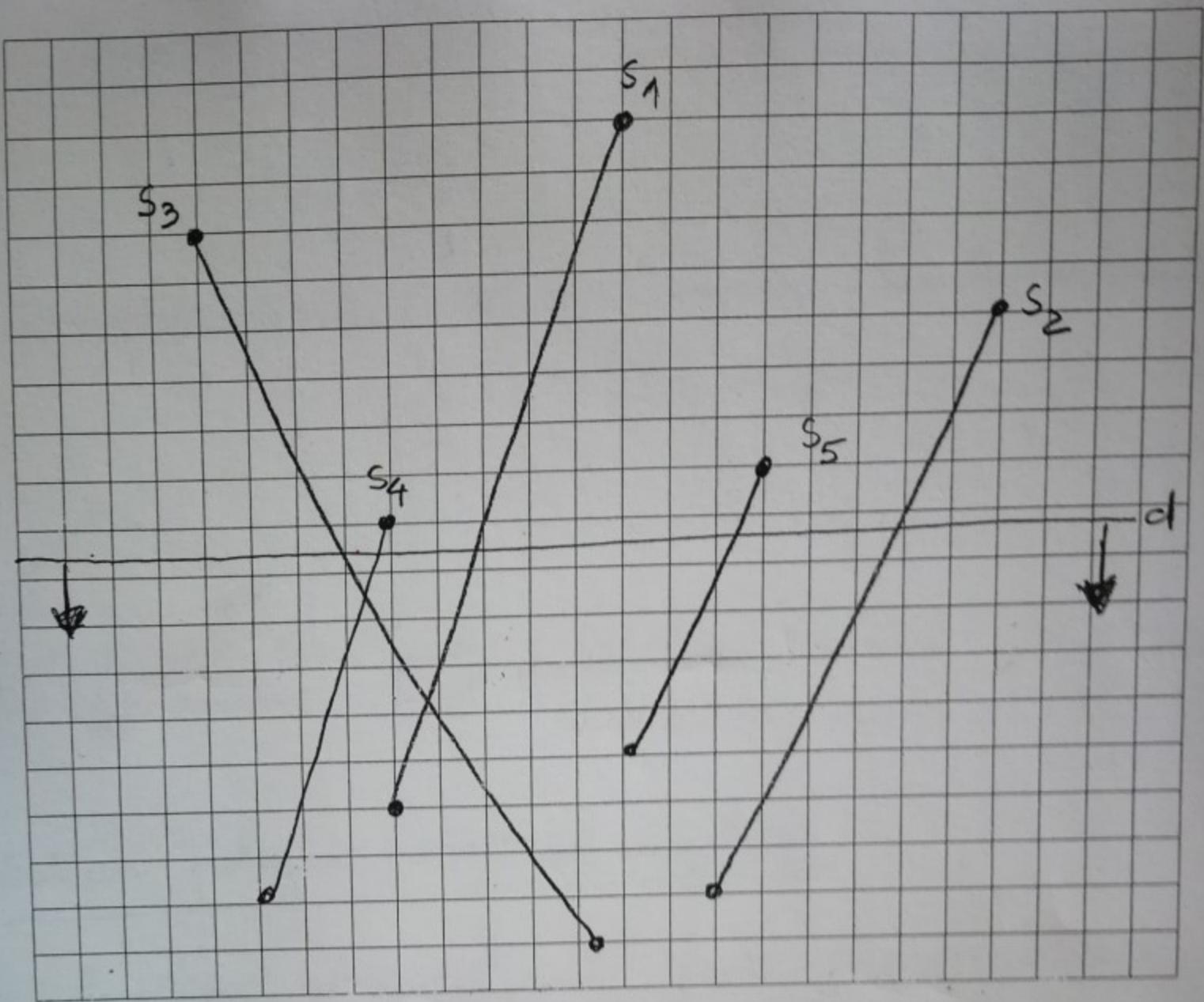
2) Margine în ferirea



Eveniment Δ_{S_1} .

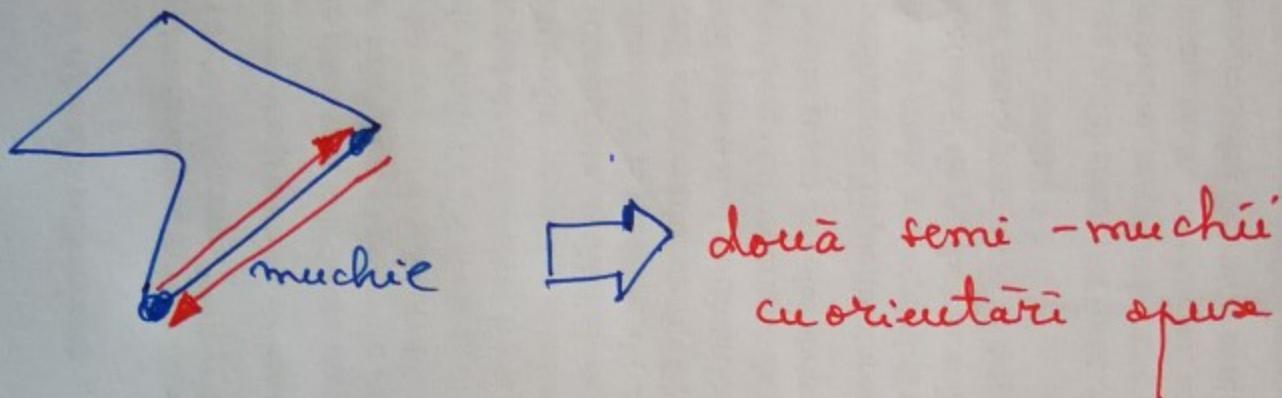
Punct de intersectie
nu detectat
anterior
(adăugat)

Punct de intersectie
detectat
anterior



Suprapunerea straturilor tematice (Overlay)

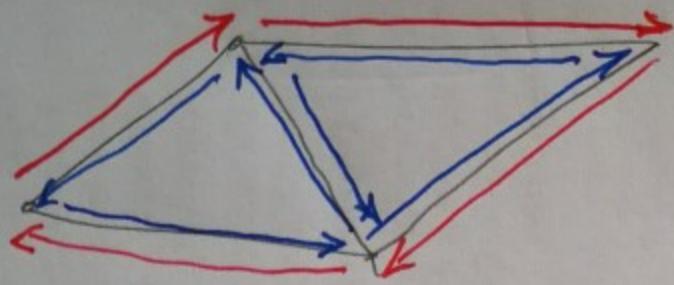
Conceptul cheie: de semi-muchie (muchie orientată)
"half-edge"



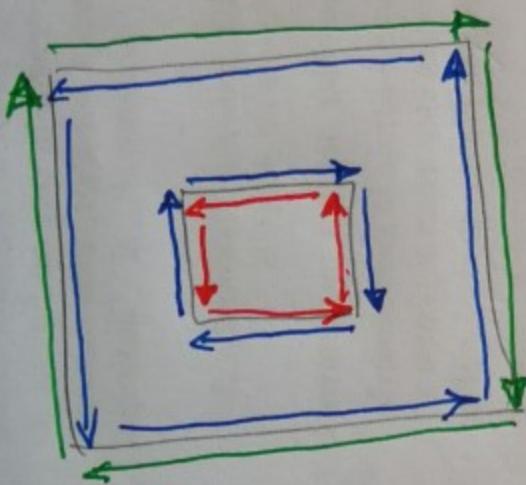
Dat un poligon (eventual cu goluri)

- frontiera exterioră, care poate fi parcursă cu ajutorul semi-muchiilor a.i. poligonul să fie la stânga frontierei, iar virajele convexe să fie la stânga;
- frontiera interioră (de ex. există goluri):
eventual cu mai multe goluri
poligonul este tot la stânga, dar virajele în vîrfurile convexe sunt la dreapta.

Exemplul 1



Exemplul 2



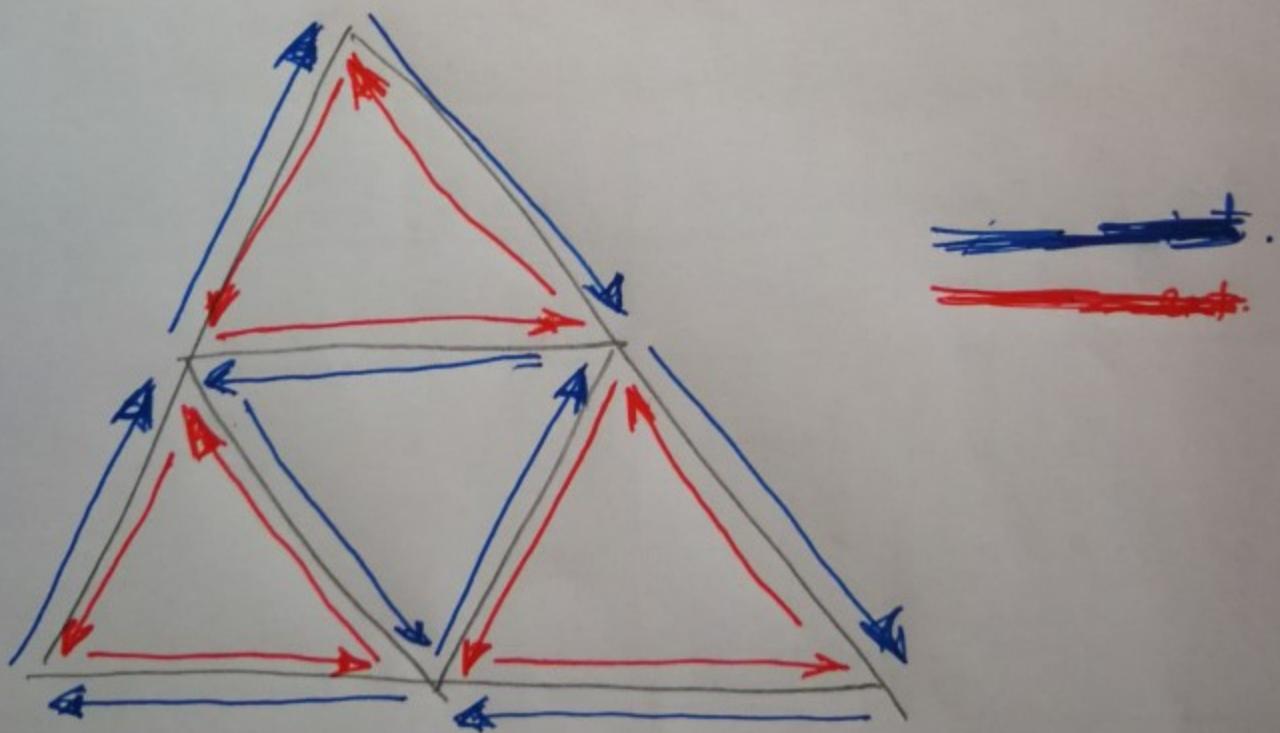
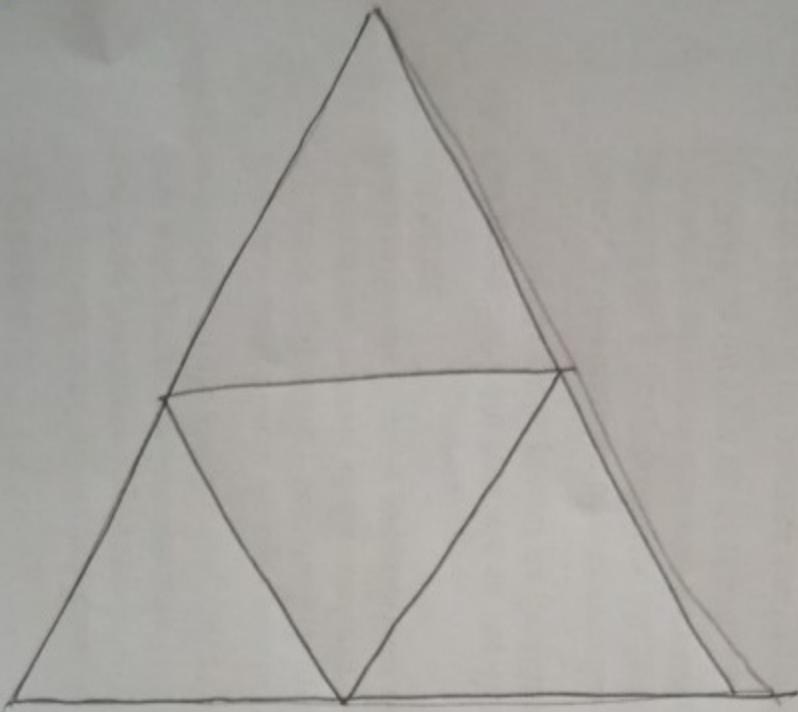
8 vârfuri

16 semii - muchii

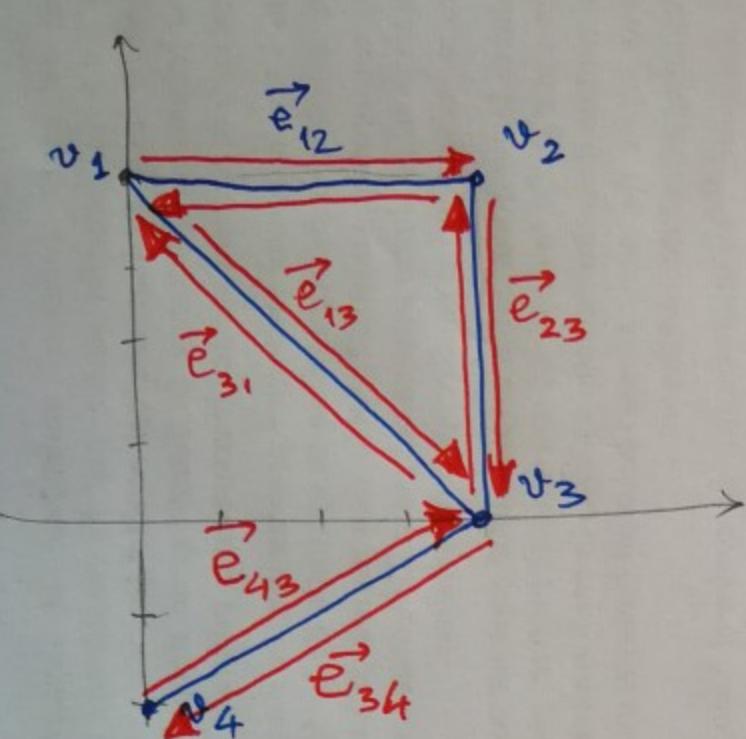
3 fete

- fr. ale "pătratului mare"
(ext + int -)
- fr. "pătratului mic"
- fr. "zonei exterioare"

Exemplul 3.



Exemplu de subdiviziune planară și lista dublu înținută (DCEL = Doubly Connected Edge List) asociată.



Vârf	Coordonate
v_1	(0, 4)
v_2	(4, 4)
v_3	(4, 0)
v_4	(0, -2)

Notăm \vec{e}_{ij} muchia de la v_i la v_j

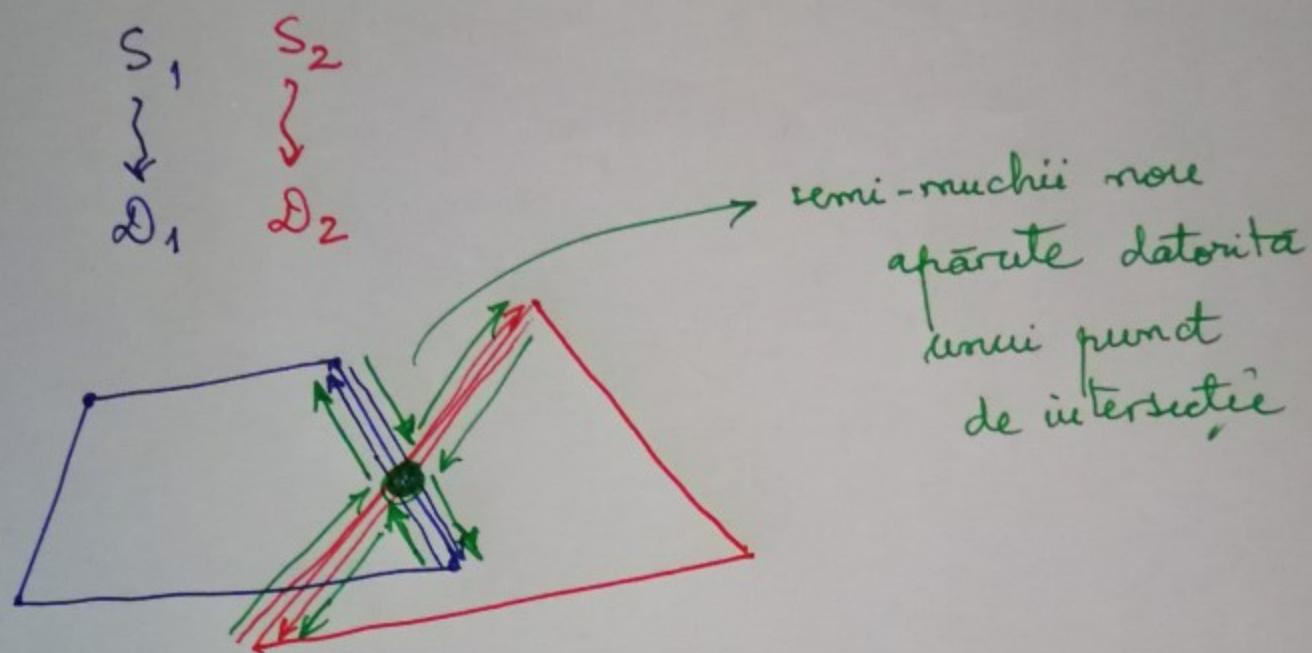
Fata	Outer Component (fr. ext.)	Inner Component (fr. int.)
f_1 (exterior)	nil	\vec{e}_{31}
f_2 (Δ)	\vec{e}_{13}	nil.

Semimuchie	Origine	Termin	Next	Prev	Incident Face
\vec{e}_{12}	v_1	\vec{e}_{21}	\vec{e}_{23}	\vec{e}_{31}	f_1
\vec{e}_{23}	v_2	\vec{e}_{32}	\vec{e}_{34}	\vec{e}_{12}	f_1
etc. . . .					

Explicați cum arunne, folosind pointerii de mai sus.

- poate fi parcursă frontiera unei fețe (poligon) exterioară / interioră,
 - pot fi găsite toate semi-muchiile incidente cu un vârf.
-

Overlay (suprapunerea straturilor tematice)



$S_1 : \begin{cases} 4 \text{ vârfuri} \\ 8 \text{ semi-muchii} \\ 2 \text{ fețe} \end{cases}$

$S_2 : \begin{cases} 3 \text{ vârfuri} \\ 6 \text{ semi-muchii} \\ 2 \text{ fețe} \end{cases}$

Overlay : $\begin{cases} 9 \text{ vârfuri} & (2 \text{ vârfuri detectate} \\ & \text{ca intersecții de segmente}) \\ 22 \text{ semi-muchii} \\ 4 \text{ fețe} \end{cases}$

Actualizarea listei de semi-muchii

Principiu: "fata delimitată de o muchie este la stânga acesteia".

Exemplu s_1 | prin vârful v al lui s_1 , trece o
 s_2 | muchie a lui s_2 (imaginăți alte
 situații!).

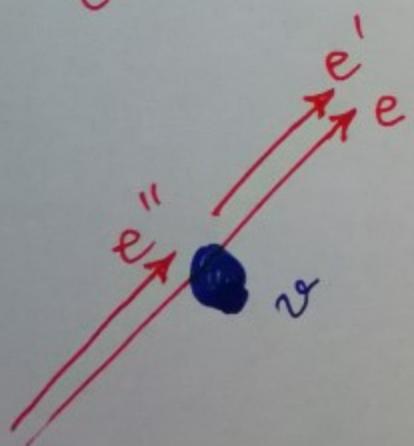
În acest caz:

- semi-muchiile din s_1 pot fi reutilizate (Δ actualizări).
- cele două semi-muchiile e și \tilde{e} din s_2 sunt înlocuite cu 4 semi-muchiile.

De exemplu: e este învelită cu e' și e'' .

e' | Origin = v
 Twin \rightsquigarrow $v \sim \tilde{e}$
 Next = Next(e)
 Prev \triangleleft

e'' | Origin = Origin(e')
 Twin \rightsquigarrow $v \sim e$
 Next \triangleright
 Prev = Prev(e').



(23)

Prev, respectiv Next pentru e' , respectiv e'' se găsesc $\frac{\delta}{2}$ parcurgând muchiile lui s_1 care trec prin v ; de exemplu: $\text{Prev}(e') =$ "cea mai apropiată în sens trigonometric"

Programare liniară

Pregătiri:

Obs. (i) Fie $\vec{v} = (v_1, v_2, v_3)$; $\vec{w} = (w_1, w_2, w_3)$ doi vectori din \mathbb{R}^3 .

$$\hat{\gamma}(\vec{v}, \vec{w}) = \arccos \left(\frac{\langle v, w \rangle}{\|v\| \cdot \|w\|} \right) \in [0, \pi].$$

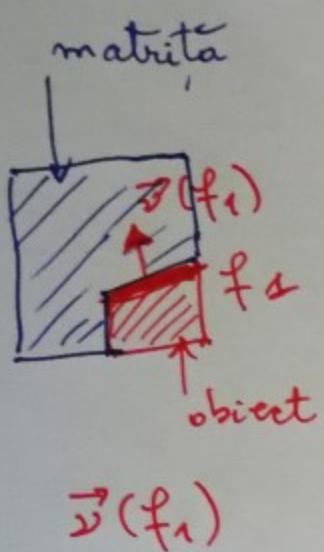
(am folosit: $\frac{\langle v, w \rangle}{\|v\| \cdot \|w\|} \in [-1, 1]$)
 \hookrightarrow ineq Cauchy - Buniakowski

$$\langle v, w \rangle = v_1 \cdot w_1 + v_2 w_2 + v_3 w_3; \|v\| = \sqrt{\langle v, v \rangle}.$$

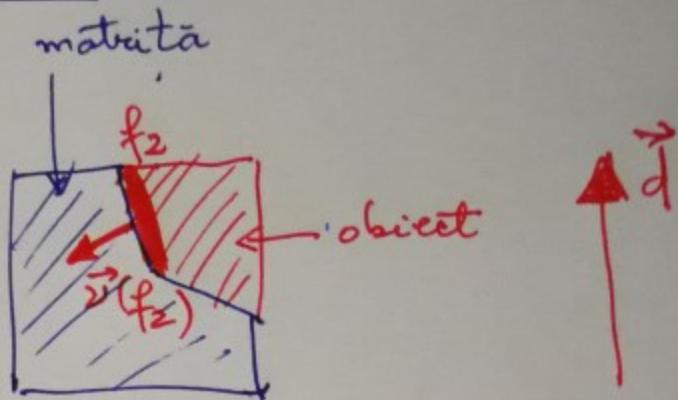
(ii). Pentru a măsura unghiul dintre directie data și fete este suficient să calculăm / să manevrăm unghiul dintre directie data și vectorii normali la fețele respective (un vector normal la un plan este un vector perpendicular pe plan, de normă 1).

Conditia ca o matră să blocheze /să nu blocheze

extragerea într-o direcție dată



$$\vec{v}(f_1)$$



$$\vec{v}(f_2)$$

$$\vec{f}_2 \perp \vec{n}$$

$$\vec{f}_2 \perp \vec{n}$$

\rightarrow nu blochează
extragerea

\Leftrightarrow unghiul dintre $\vec{v}(f_2)$
și \vec{d} este $\geq 90^\circ$

\Leftrightarrow

$$\cos(\angle(\vec{d}, \vec{v}(f_2))) \leq 0$$

Fata \hat{f}_1 a matrării, care
coresponde fetei f_1 a
piesei blochează extragerea
în direcție \vec{d}

\Leftrightarrow unghiul dintre

normala $\vec{v}(f_1)$ și \vec{d}

este $< 90^\circ$ \Leftrightarrow

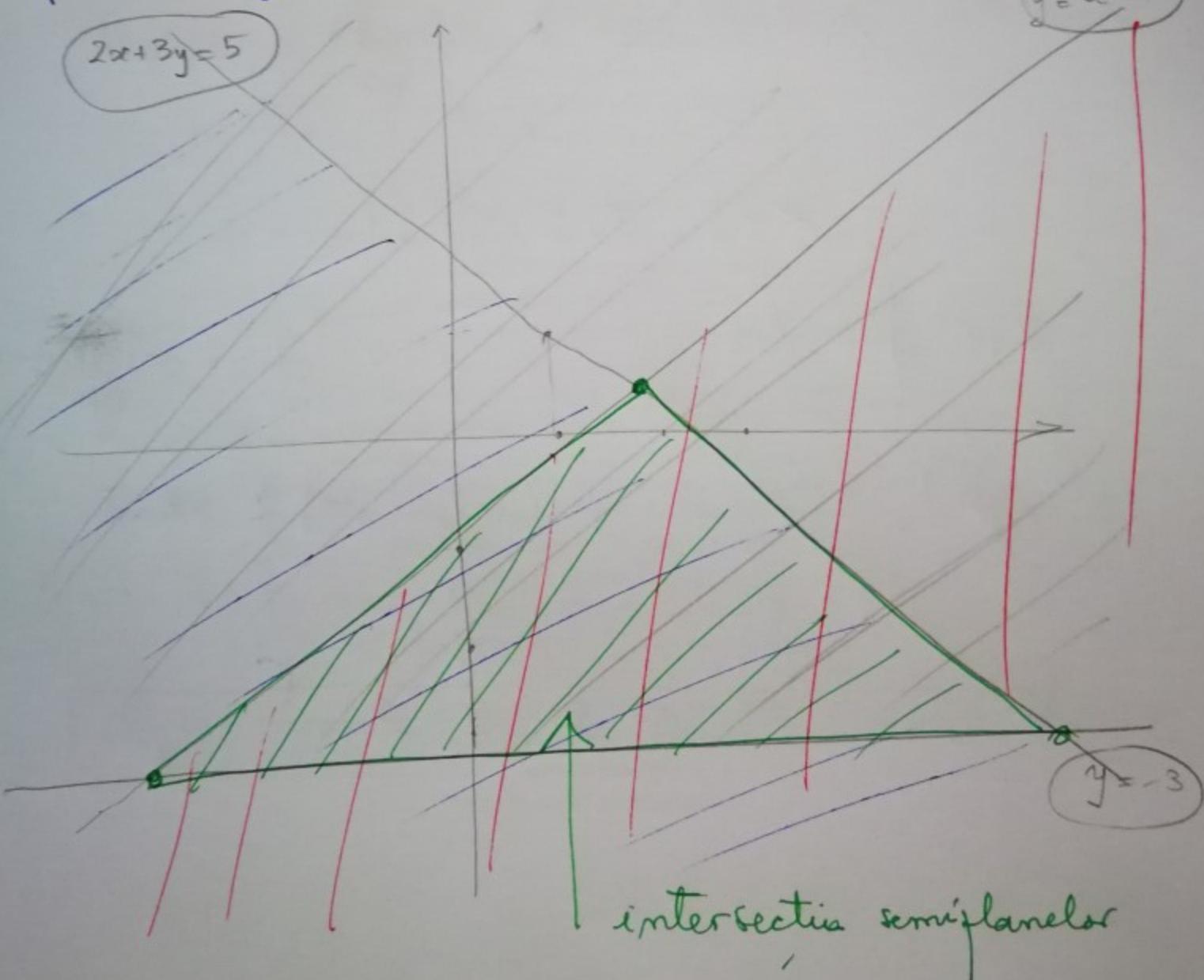
$$\Leftrightarrow \cos(\angle(\vec{d}, \vec{v}(f_1))) > 0$$

Această condiție trebuie verificată pentru toate
fetele!

Exemplu:

① Semiplane și intersecții

$$\begin{cases} -x + y + 1 \leq 0 \quad \text{[shaded]} \rightsquigarrow y = x - 1 \\ -y - 3 \leq 0 \quad \text{[shaded]} \rightsquigarrow y = -3 \\ 2x + 3y - 5 \leq 0 \cdot \text{[shaded]} \rightsquigarrow 2x + 3y = 5 \end{cases}$$



Detaliere (condiție scrisă în coordonate)

F.r.g. putem presupune că $\vec{d} = (d_x, d_y, 1)$ (de ce?)
(de fapt): "a da o direcție "în sus" este echivalent
cu a da un punct din planul $z=1$ ".

Fie f o față fixată a obiectului, $\vec{v}(f) = (v_x, v_y, v_z)$

Faptul că \vec{f} față \hat{f} nu blochează extragerea
în direcția $\vec{d} \iff$

$$\langle \vec{v}(f), \vec{d} \rangle \leq 0 \iff$$

$$v_x \cdot d_x + v_y \cdot d_y + v_z \leq 0$$

(\ast_f)

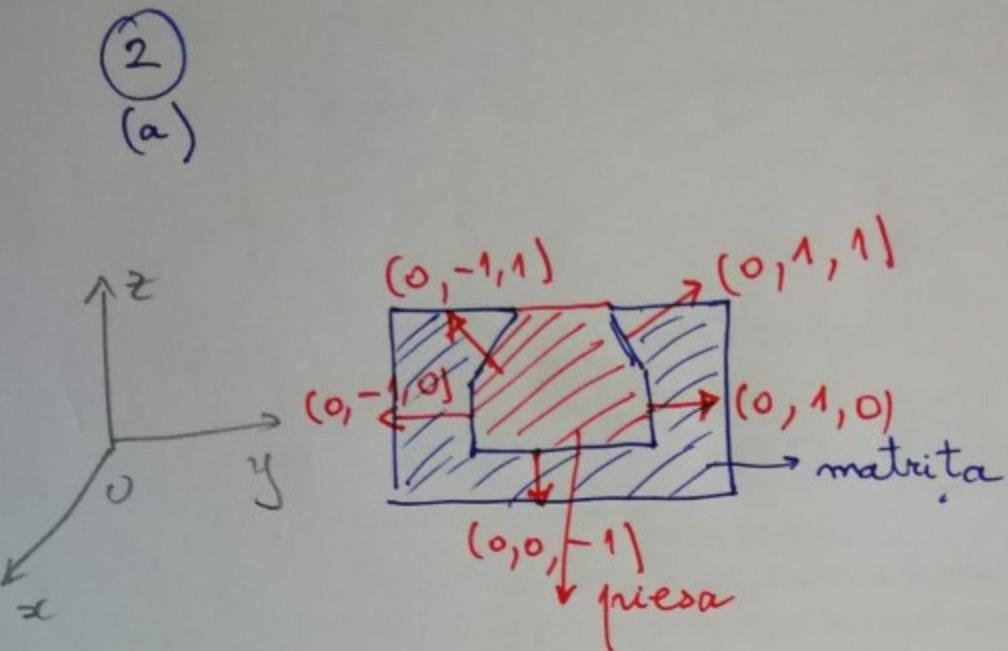
Fixată $f (v_x, v_y, v_z)$ și este căutat $\vec{d} (d_x, d_y)$
a.t. să fie verificată inegalitatea (\ast_f) .

(\ast_f) : inecuație care descrie un semiplan.

(23)

8/6

Exemplu - legătura dintre normale, extragerea unei
obiect și sisteme de inecuație
(intersectii de semiplane)



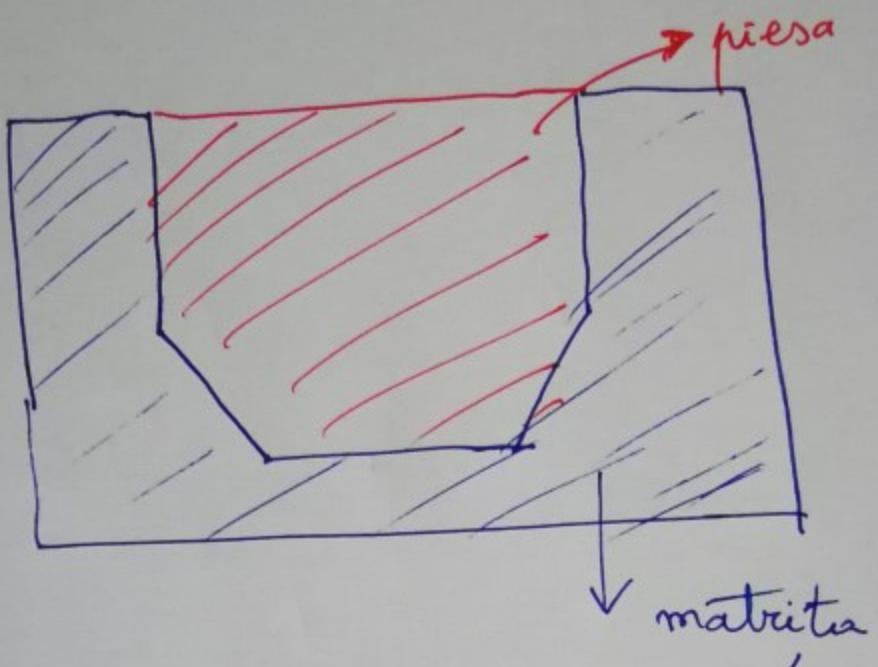
obiectul
nu poate
fi extras
în sus

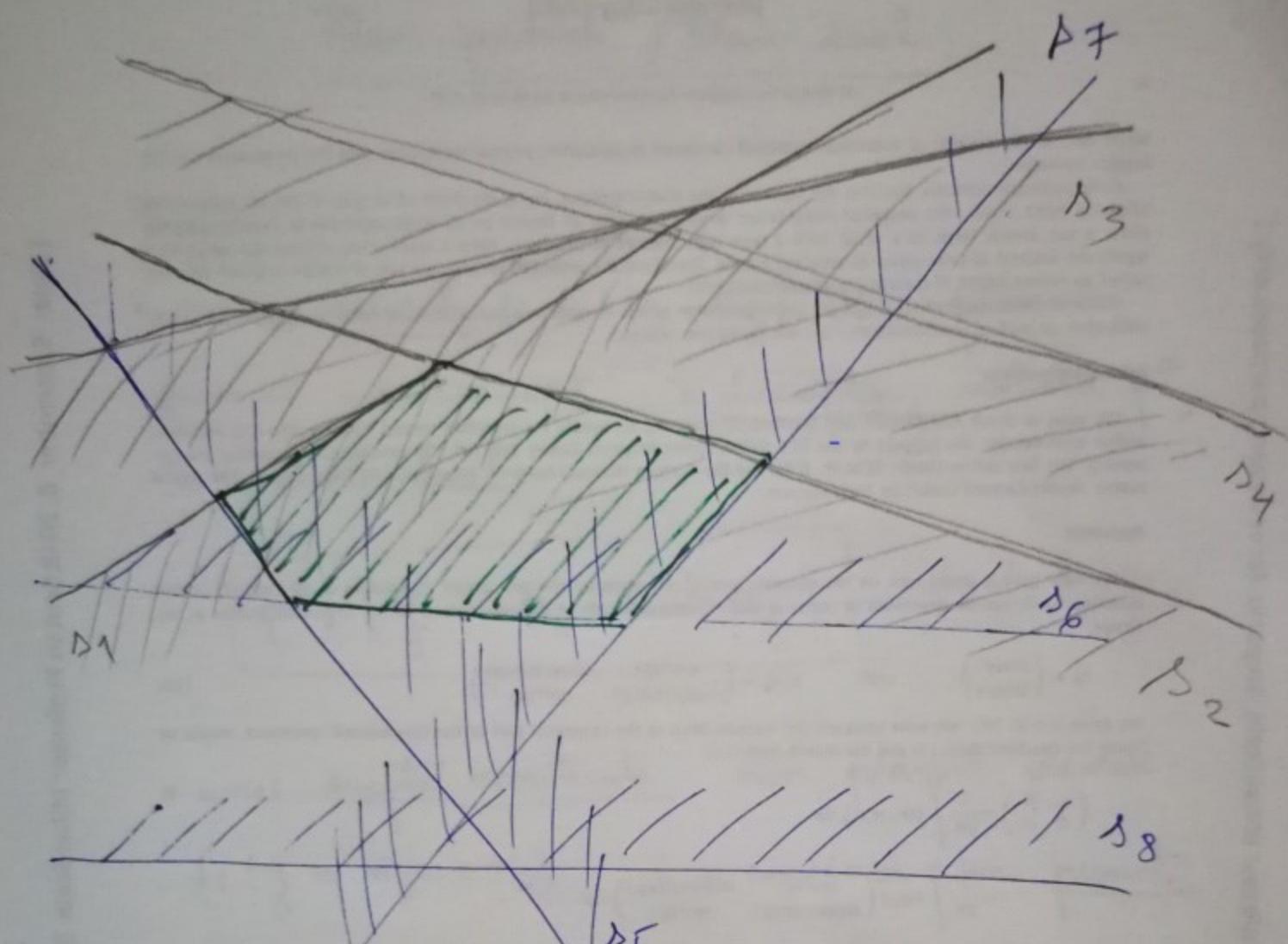
Normalele: $(0, -1, 1); (0, 1, 1); (0, 1, 0); (0, 0, -1);$
 $(0, -1, 0)$

Cf. teorie: scriem inecuații de tipul (x_f)

$$\begin{array}{l} (0, -1, 1) \rightarrow 0 \cdot x + (-1) \cdot y + 1 \leq 0 \\ (0, 1, 1) \rightarrow 0 \cdot x + 1 \cdot y + 1 \leq 0 \\ (0, 1, 0) \rightarrow 0 \cdot x + 1 \cdot y + 0 \leq 0 \\ (0, 0, -1) \rightarrow 0 \cdot x + 0 \cdot y - 1 \leq 0 \\ (0, -1, 0) \rightarrow 0 \cdot x + (-1) \cdot y + 0 \leq 0 \end{array} \quad \left. \begin{array}{l} y \geq 1 \\ y \leq -1 \\ y \leq 0 \\ -1 \leq 0 \\ y \geq 0 \end{array} \right\}$$

②
(b)





semiplane inferiore: s_1, s_2, s_3, s_4

relevante tr. intersec.

semiplane superiore: s_5, s_6, s_7, s_8

LE (lower envelope)

UE

Transformarea de dualitate.

Plan primal / Plan dual.

Reguli:

- unei punct $p = (p_x, p_y)$ din planul primal i se asociază o dreaptă, notată p^* , în planul dual:

$$p^*: (y = p_x \cdot x - p_y) \quad \text{duala lui } p$$

- unei drepte neverticale din planul primal $d: (y = m_d x + n_d)$ i se asociază un punct din planul dual, notat d^* :

$$d^* = (m_d, -n_d). \quad \text{dualul lui } d$$

Obs. Această transformare este, de fapt, polaritatea față de parabola $y = \frac{x^2}{2}$.

Proprietăți fundamentale:

Transformarea de dualitate:

1) Păstrează incidenta:

$$p \in d \Leftrightarrow d^* \in p^*$$

Exemplu.

$$d: y = 2x + 1$$

$$p = (1, 3)$$

$$p \in d$$

$$d^* = (2, -1)$$

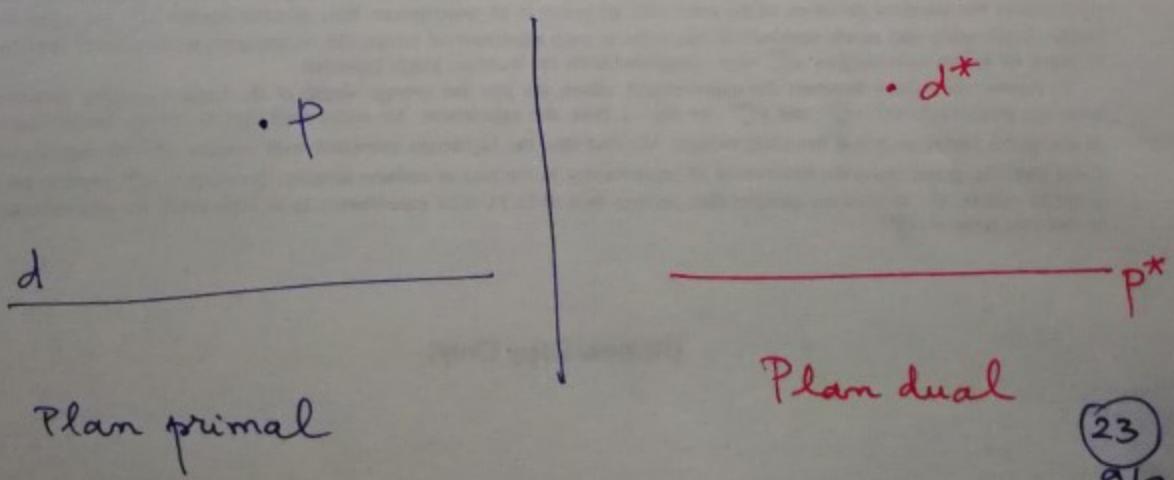
$$p^*: (y = x - 3)$$

$$d^* \in p^*$$

2) Păstrează "ordinea"

p este situat deasupra dreptei d (reverticale) \Leftrightarrow

$$d^* \rightarrowtail \rightarrowtail \rightarrowtail \rightarrowtail p^*.$$

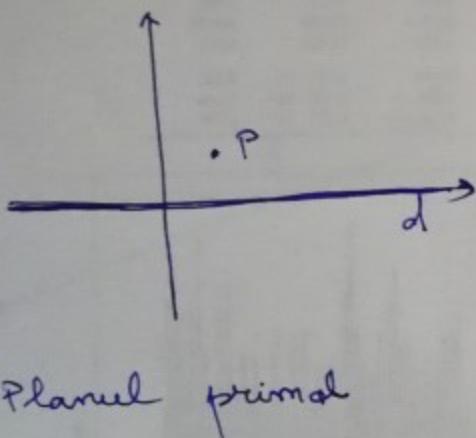


(23)
9/3

Exemple

$$p = (1, 1)$$

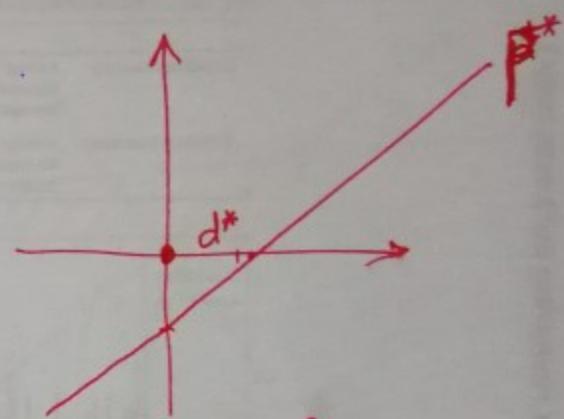
$$d: (y = 0)$$



Planel primal

$$p^*: (y = x - 1)$$

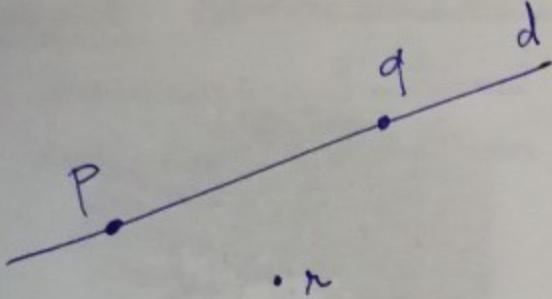
$$d^* = (0, 0)$$



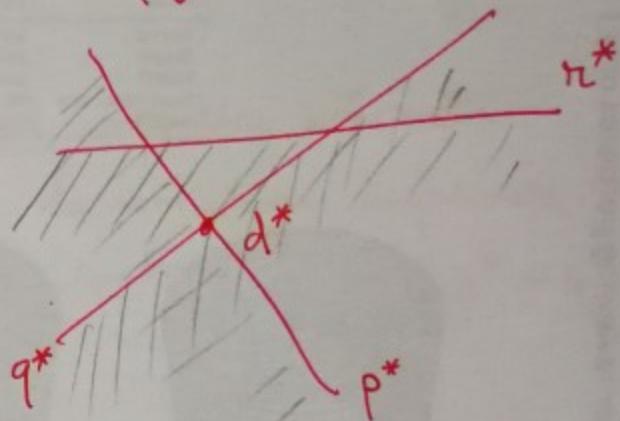
(23)
9/4

Observație + rezultate

- Fie $p, q, p \neq q$. Fie r un punct situat de desul dreptei $d = pq$. Configurația duală?



planul primal
 r este de desul dreptei $d = pq$

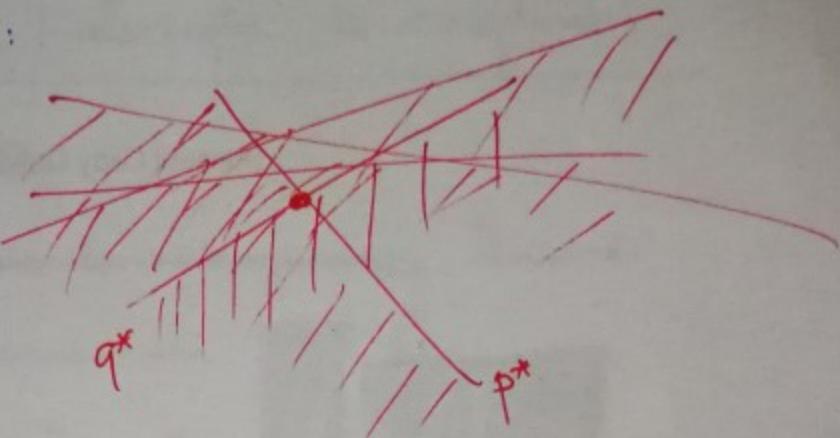
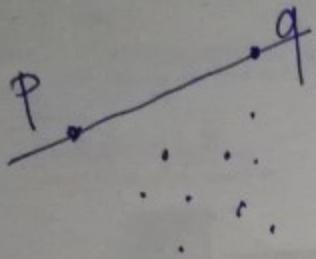


planul dual
 d^* este de desul dreptei r^*

- Fie β o mulțime de puncte. Ce înseamnă că un segment $[pq]$ participă la frontieră $p, q \in \beta$ superioară a lui $\text{Conv}(\beta)$? \rightarrow toate celelalte puncte sunt situate de desul dreptei pq .

Dual: Se consideră dreptele p^* și q^* și punctul de intersecție $d^* \Rightarrow$ punctul d^* este situat de desul dreptelor corespunzătoare celorlalte puncte. Prin trecere la semiplane inferioare (v. figura): dintre $p_{inf}^*, q_{inf}^*, r_{inf}^*$ (23) "conțină" p_{inf}, q_{inf} . 9/5.

Cumai multe puncte:



primal: pentru partea
superioară a frontierei
acp. convexe "contează"
doar p și q

dual: pentru intersecții
de semiplane inferioare
"contează" doar p^* și q^*

Concluzie: a determina frontiera superioară a
acoperirii convexe pentru mulțimea de puncte P
este echivalent cu a determina $\mathcal{L}E$ pentru
semiplanele inferioare determinate de dreptele duale.

Analog: frontiera inferioară / UE .

Aspecte calitative (intersectie de semiplane)

Programare liniara

Exemplu Problema de programare liniara
1-dimensională ($d=1$).

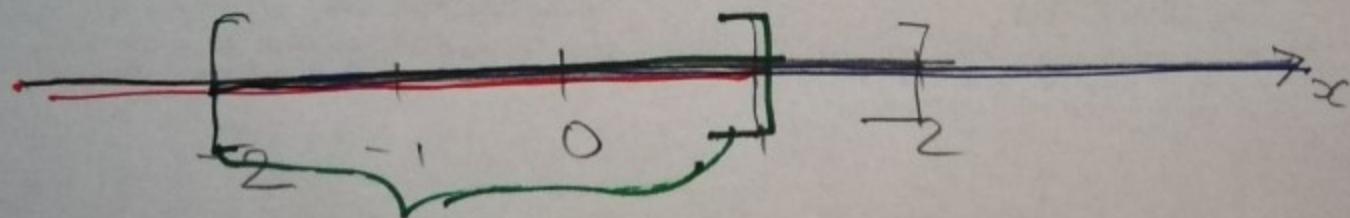
Cordonata $x_1 \stackrel{\text{NOT}}{=} x$.

funcție obiectiv

$$\begin{aligned} & \text{maximizează } (cx) \\ & \left\{ \begin{array}{l} a_1x \leq b_1 \rightarrow \text{interval} \\ a_2x \leq b_2 \rightarrow \text{---} \\ \vdots \\ a_nx \leq b_n \rightarrow \text{---} \end{array} \right. \\ & \text{contrângeri} \end{aligned}$$

Exemplu "concret"

$$\begin{cases} \text{maximizează } (2x) \\ \begin{array}{ll} 3x \leq 6 & x \leq 2; x \in [-\infty, 2] \\ -2x \leq 4 & x \geq -2; x \in [-2, \infty) \\ 6x \leq 6 & x \leq 1; x \in (-\infty, 1] \end{array} \end{cases}$$



↓ intervalul $[-2, 1]$:
regiune fezabilă

Maximul funcției obiectiv este egal cu 2 și se obține pentru $x = 1$.

Prop. Pentru $d=1$: un program liniar 1-dimensional
→ timp liniar ($O(n)$).

(23)

9/7

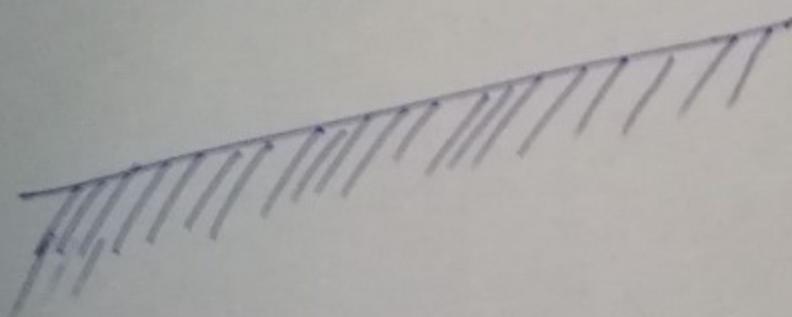
Intersecții de semiplane

- caracterizare "cantitativă"

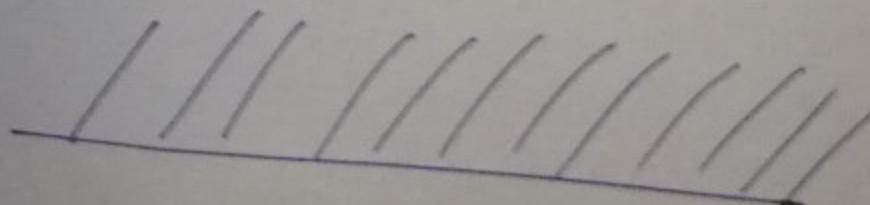
Var. 1 alg. "Divide et Impera" / și "overlay".
(v. suprăt de curs).

Var. 2

Terminologie:



semiplan
inferior
(lower half-plane)



semiplan
superior
(upper half-plane)

? Dat un semiplan ^{dat de o dreaptă} neverticală

$$ax + by + c \leq 0$$

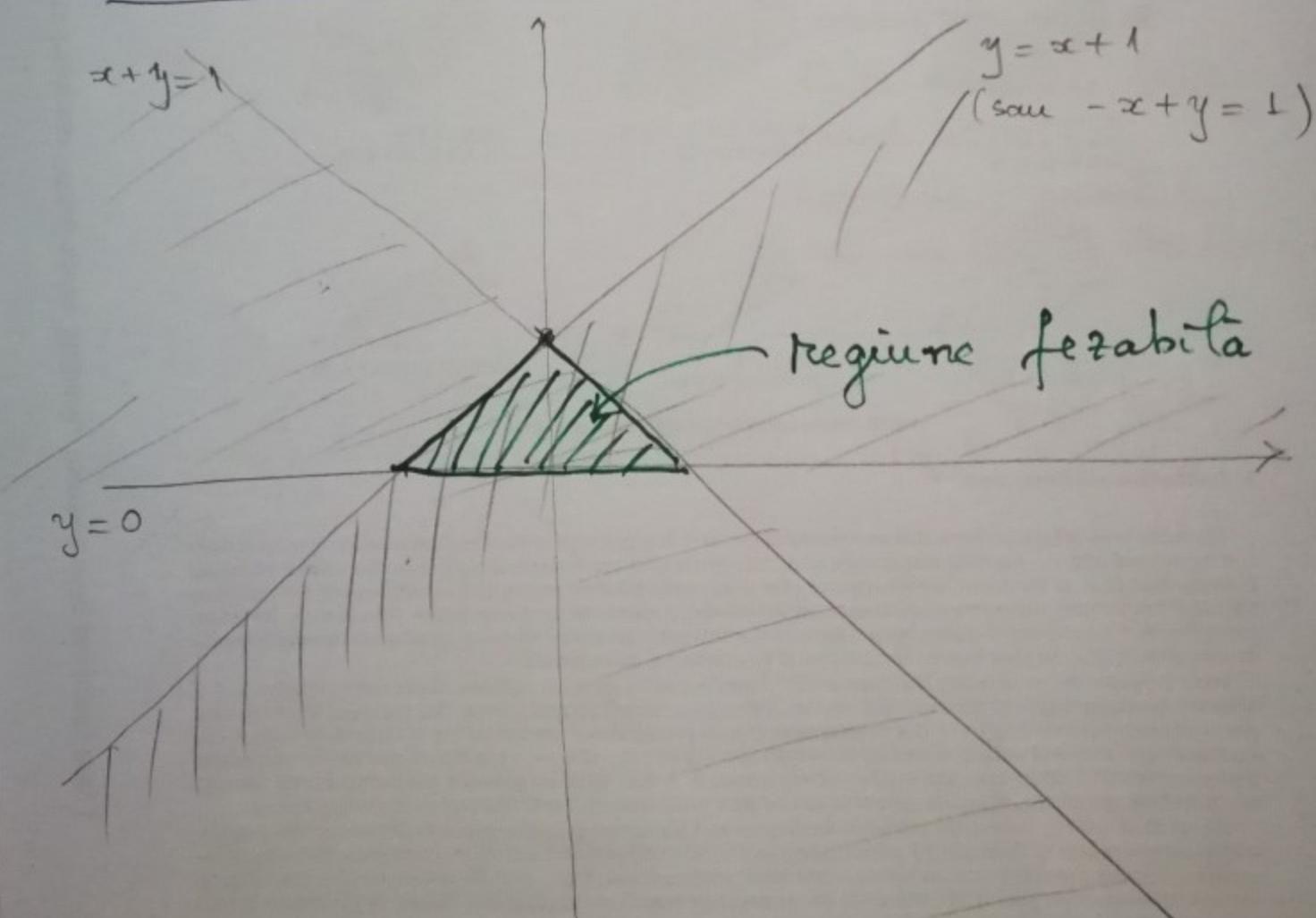
cum se decide dacă este inferior / superior?

Exemplu. Problema de programare liniară
2-dimensionala ($d=2$).

Notăm coordonatele cu x și y .

maximizează y ; $\vec{c} = (0, 1)$
 $0 \cdot x + 1 \cdot y$

$$\begin{cases} x+y \leq 1 \\ -y \leq 0 \\ -x+y \leq 1 \end{cases}$$



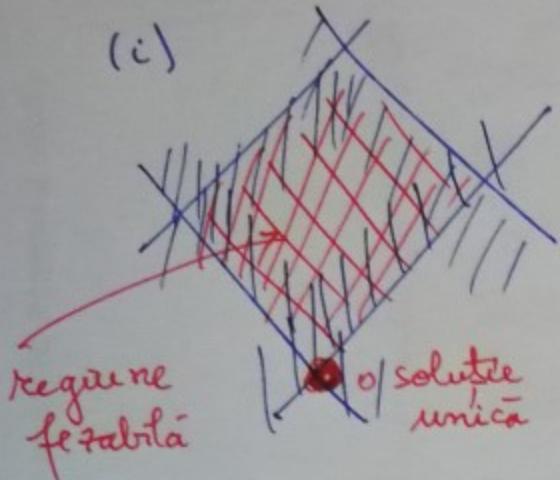
În punctul $(0, 1)$ este maximizată funcția obiectiv.

Situatie

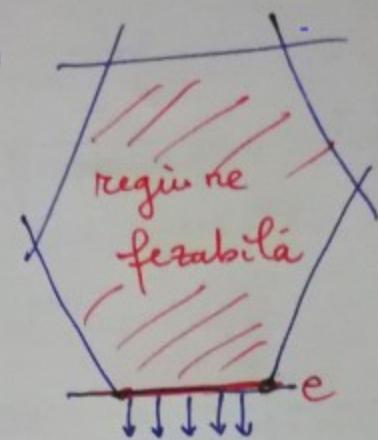
(referitor la regiunea fezabilită).

În desene: $\vec{c} = (0, -1)$

(i)



(ii)



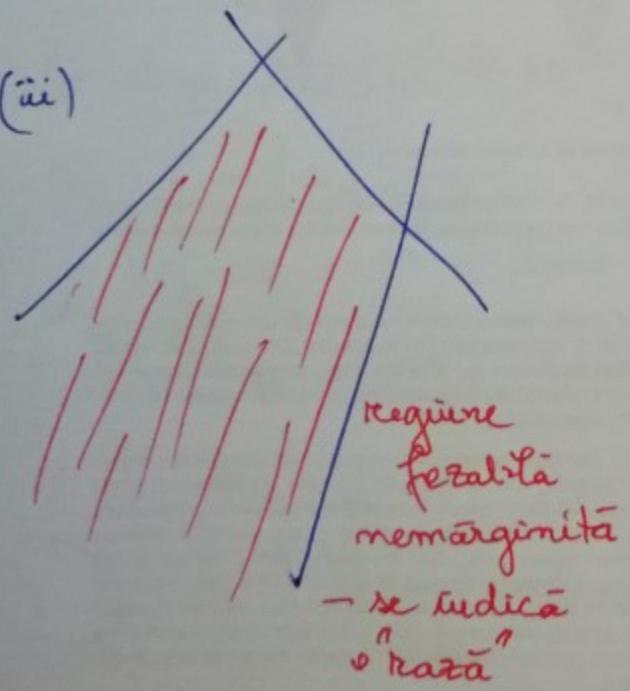
$$\begin{cases} Z = \\ (0, -1) \end{cases}$$

mormăla exterioară are același sens ca \vec{c}

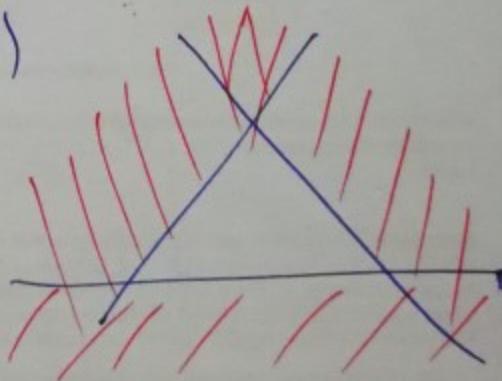
toate punctele lui e sunt soluții

Convenție: punctul "cel mai mic în sens lexicografic".

(iii)



(iv)



regiunea fezabilită \emptyset

Algoritm incremental pentru rezolvarea unei probleme de programare liniară 2D

Principiu:

- constrângările sunt adăugate una câte una;
- după ce la fiecare pas soluția (pt. de maxim) există, apoi "actualizată";
- sunt adăugate la început constrângeri care garantează mărginirea programului liniar.

Noile constrângeri:

Se alege $M > 0$ și se definesc noi constrângeri astfel:

Fie $\vec{c} = (c_x, c_y)$.

$$m_1: \begin{cases} x \leq M, & \text{dacă } c_x > 0 \\ x \geq -M, & \text{dacă } c_x \leq 0. \end{cases}$$

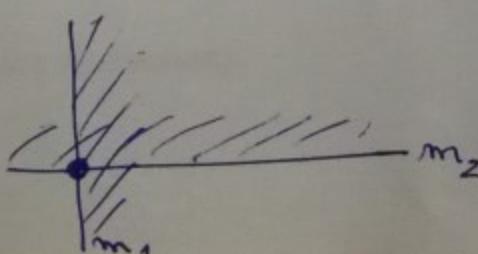
$$c_0 \stackrel{\text{NOT}}{=} m_1 \cap m_2.$$

$$m_2: \begin{cases} y \leq M, & \text{dacă } c_y > 0 \\ y \geq -M, & \text{dacă } c_y \leq 0 \end{cases}$$

În continuare: $\vec{c} = (0, -1)$, azadar:

$$m_1: x \geq -M$$

$$m_2: y \geq -M$$



23

10/3

Obs. Se lucrează cu convenția de ordonare lexicografică,
a.i. există o unică soluție optimă.

Notatii Fie (H, \vec{c}) un program liniar cu
constrângările h_1, h_2, \dots, h_n .

Se notează: \downarrow multime de semiplane

$$H_i = \{m_1, m_2, h_1, h_2, \dots, h_i\}$$

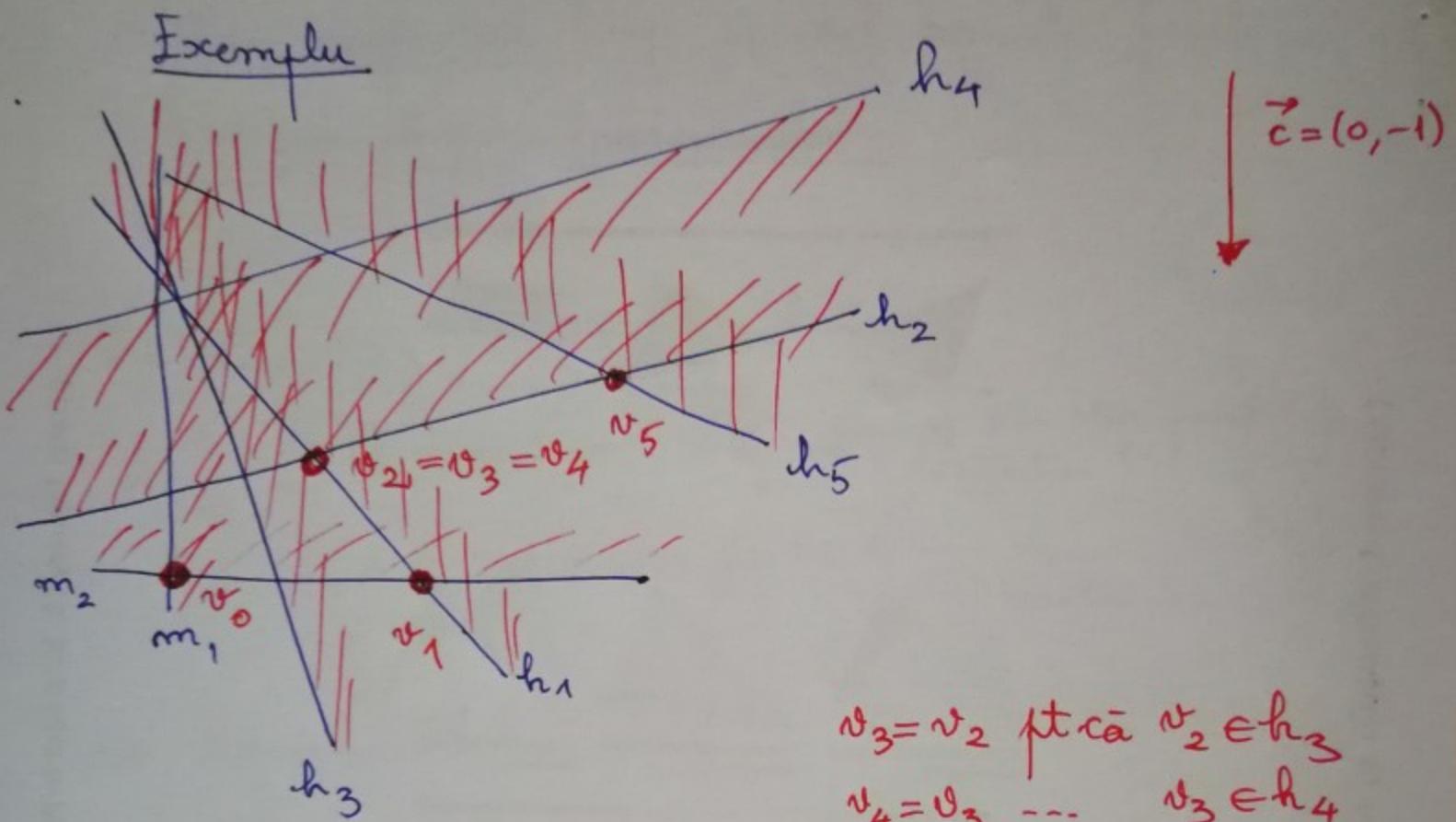
$$C_i = m_1 \cap m_2 \cap h_1 \cap h_2 \cap \dots \cap h_i$$

\uparrow regiune feasible

(notatia este pentru $i = \overline{0, n}$, în particular
 $H_0 = \{m_1, m_2\}$; $C_0 = m_1 \cap m_2$).

Obs. (i) $C_0 \supseteq C_1 \supseteq C_2 \supseteq \dots \supseteq C_n = C$

(ii) Pentru fiecare i , regiunea feasible C_i , dacă
este nevoidă, are un vîrf care reprezintă o
soluție (optimală) problemei (H_i, \vec{c}) , notată cu
 v_i (! depinde de alegera lui m_1, m_2).



$v_3 = v_2$ pt că $v_2 \in h_3$

$v_4 = v_3 \dots v_3 \in h_4$

$v_5 \neq v_4 \dots v_4 \notin h_5$

Obs. Tie $1 \leq i \leq m$; pp. $c_{i-1} \geq v_{i-1}$ determinante.

Considerăm h_i . Sunt două situații:

(i) dacă $v_{i-1} \in h_i$, atunci $v_i = v_{i-1}$;

(ii) dacă $v_{i-1} \notin h_i$, atunci

fie $c_i = \phi$

fie $v_i \in d_i$, unde d_i este dreapta care mărginește h_i .

În cazul în care $v_i \in d_i$, găsirea lui v_i revine la găsirea lui $p \in d_i$ care maximizează $f_0(p)$, date constrângeri deja existente ($p \in h_i \wedge h_i \in H$) \rightarrow de fapt: acesta este o problemă de programare liniară 1-dimensională care are complexitatea - timp liniar, adică $O(i)$ 23

Analiza complexității timp în cazul variantei bazată pe alegeră aleatorie a semiplanelor

Fie $(x_i)_{i=1,n}$ variabila aleatoare.

$$x_i = \begin{cases} 0, & \text{dacă } v_{i-1} \in h_i \text{ (adică este ales pixel 5)} \\ 1, & \text{dacă } v_{i-1} \notin h_i \text{ (iterație } i \text{)} \end{cases}$$

$$\Rightarrow \text{timpul total } \sum_{i=1}^n x_i O(i)$$

Valoarea așteptată (timpul mediu):

$$\begin{aligned} E \left[\sum_{i=1}^n x_i \cdot O(i) \right] &\stackrel{\substack{\text{altă} \\ \text{notare}}}{=} \mu \left(\sum_{i=1}^n x_i O(i) \right) \\ &= \sum_{i=1}^n O(i) \cdot \underbrace{\mu(x_i)}_{\text{afirmatie}} \leq \sum_{i=1}^n O(i) \cdot \frac{2}{i} = O(n). \end{aligned}$$

Afirmatie: $\mu(x_i) \leq \frac{2}{i}$.

Altfel spus, probabilitatea ca $v_{i-1} \notin h_i$ este $\leq \frac{2}{i}$.

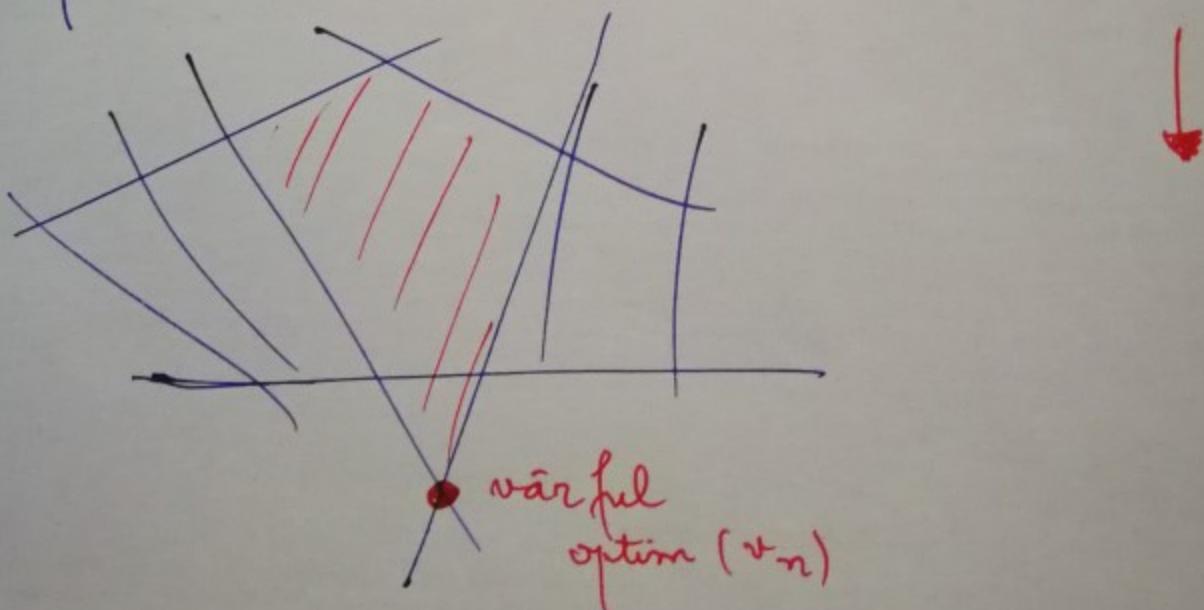
"Backward analysis" Pp algoritmul terminat,

v_n vârful optim.

- care este probabilitatea ca $v_{n-1} \neq h_n$, adică la adăugarea lui h_n , vârful v_{n-1} să fie modificat în v_n ?



care este probabilitatea ca eliminând unul dintre semiplane să fie modificat vârful optim?



- nr. de cazuri posibile: m
- nr. de cazuri în care survine modificarea statului de vârf optim : $\underline{\underline{2}} (\leq 2)$

=> probabilitatea de a actualiza la pasul n vârful optim este $\leq \frac{2}{m}$; analog pîr. cazul i

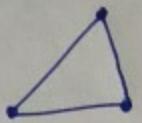
$$\Rightarrow \mu(X_i) \leq \frac{2}{i} \quad \underline{\text{ctd}}$$

23
10%

Triangularea mulțimilor de puncte

Exemple

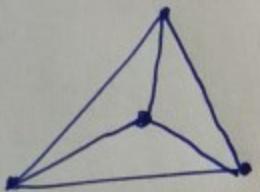
(i) 3 puncte necoliniare



3 vârfuri
3 muchii
1 față / 1 Δ

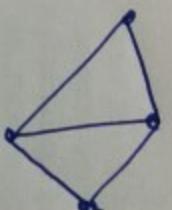
(ii) 4 puncte necoliniare (resituate pe o aceeași dreaptă |)

a)



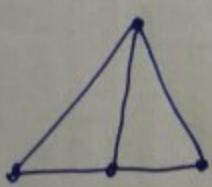
4 vârfuri
6 muchii
3 Δ

b)



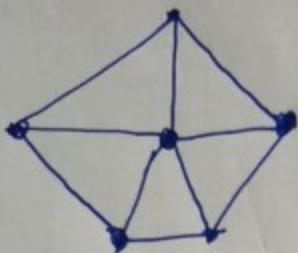
4 vârfuri
5 muchii
2 Δ

c)



4 vârfuri
5 muchii
2 Δ

Dem. (Propoziția referitoare la numărul de muchii/de fețe ale unei triunghiuri).



graf:

- vârfurile (nodurile) grafului: punctele initiale (n)
- muchii: laturile Δ ($\cancel{m_m} = ?$)
- fetele: triunghiurile + fața exterioară

$\cancel{m_t}$
" ?

1

- Relația lui Euler: $m - m_m + (m_t + 1) = 2$

- Incidente (adiacente) tuturor muchiilor și fețelor

$$2 \cdot m_m = 3m_t + k$$

↑
incidente din
"perspectiva
muchiielor"

↑
incidente din
"perspectiva fețelor"

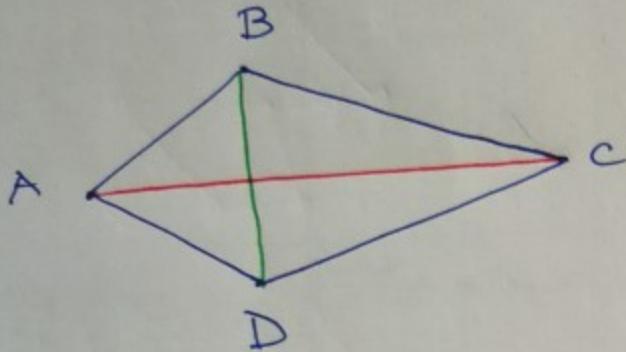
$$\Rightarrow \dots \Rightarrow \begin{cases} m_m = \dots \\ m_t = \dots \end{cases} \text{ (v. suportul de curs).}$$

(23)

11/2

Compararea triunghiurilor

Exemplu. Fie $ABCD$ un patrulater convex.



T_{AC} cu $\triangle ACB$
 $\triangle ACD$

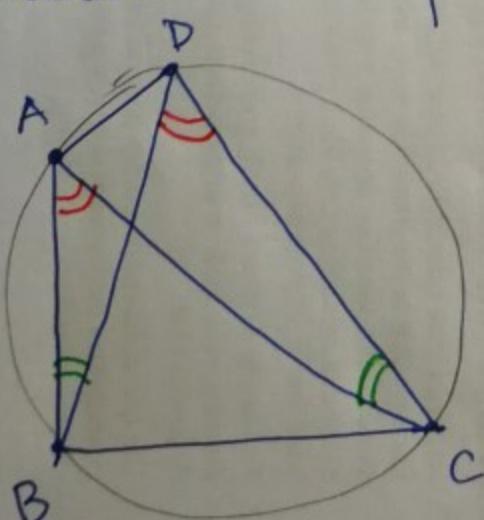
T_{BD} cu $\triangle BDA$
 $\triangle BDC$

- $A(T_{AC}) < A(T_{BD})$
- $\min(A(T_{AC})) < \min(A(T_{BD}))$, deci muchia

AC este ilegală.

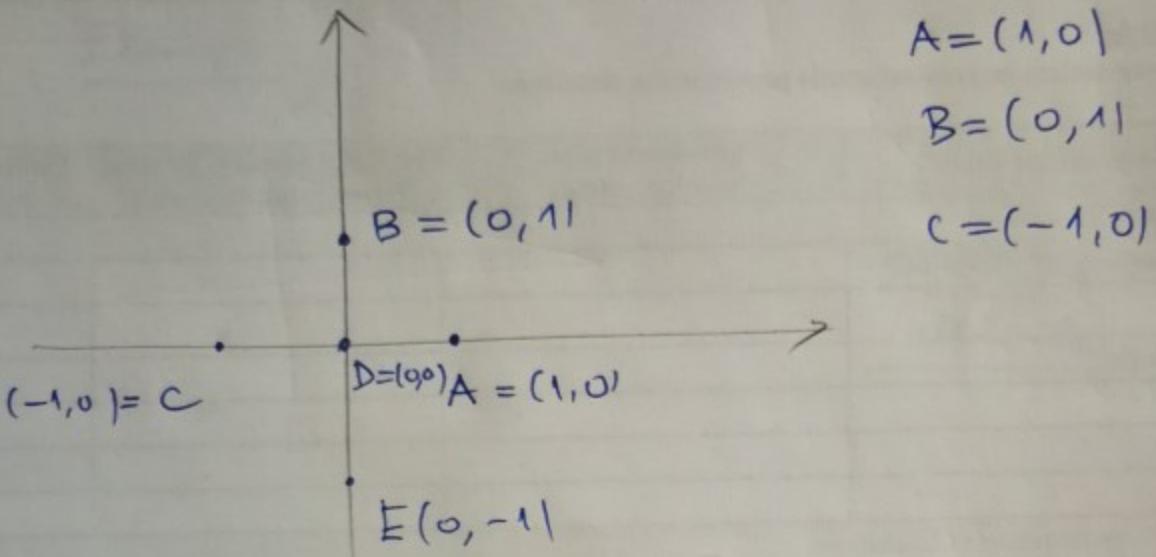
Excepție de la definitia muchiei ilegale (caz degenerat)

Fie $ABCD$ un patrulater inscris în cerc (adică A, B, C, D coincid) → situate pe același cerc



Vectoarele $A(T_{AC})$ și $A(T_{BD})$
nu sunt măsurări egale, dar
 $\min(A(T_{AC})) = \min(A(T_{BD}))$
În acest caz: ambele muchii
sunt legale
(niciuna nu este ilegală!).

Exemplu pentru $\Theta(A, B, C, D)$



ABC este un viraj la stanga.

$$\Theta(A, B, C, D) = \begin{vmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \stackrel{\text{cancel}}{=} 1$$
$$= \begin{vmatrix} 1 & 1 \\ -1 & 1 \end{vmatrix} > 0.$$

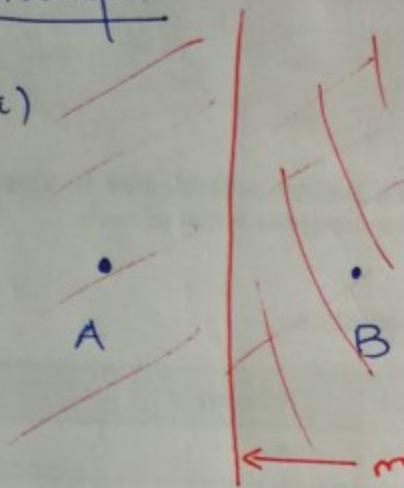
Exercitiu. Fie $E = (0, -1)$. Calculati $\Theta(A, B, C, E)$.

(23)
11/4

Diagramme Voronoi ($\in \mathbb{R}^2$)

Exemple

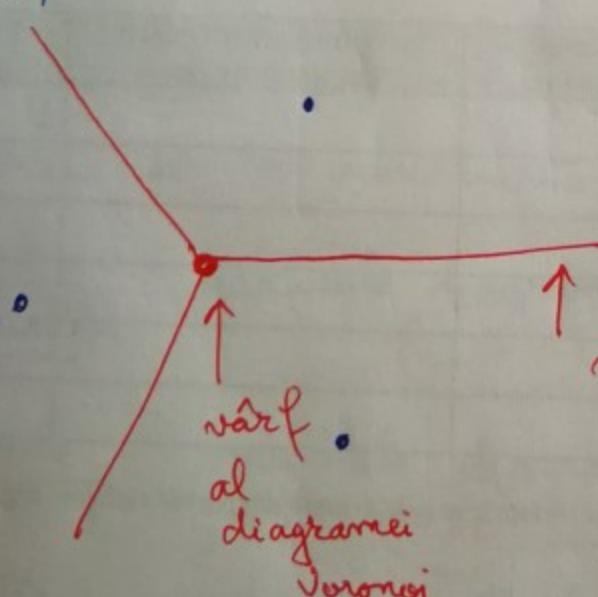
(i)



două puncte (sunt)
~~împreună~~
distințe

mediatoarea segmentului $[AB]$
(locul geometric al
punctelor egal departate
de A și de B).

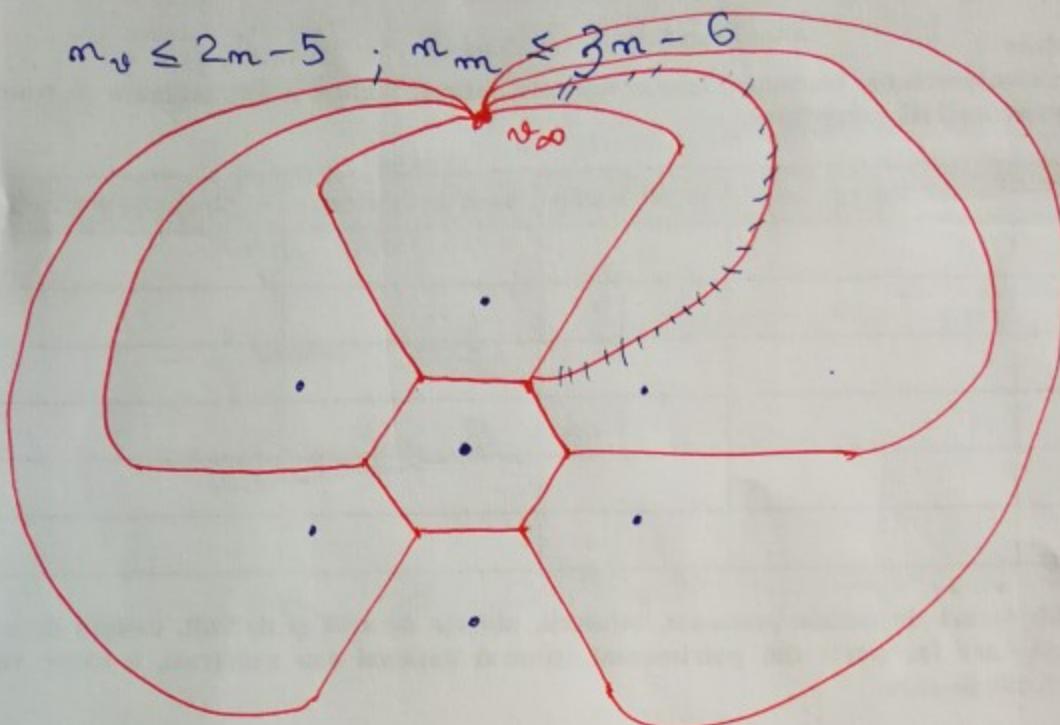
(ii)



trei puncte
necoliniare

muchie a diagrammei
Voronoi

Ideu. Rezultatul referitor la numărul de muchii /
de vârfuri ale unei diagrame Voronoi



Îp.că punctele nu sunt coliniare (altminteri evident...)

Adăugăm un punct (v_{oo}) prin care trece toate
muchiiile de tip semidreapta \rightarrow graf planar conex.

vârfurile | : vârfurile diagramei Voronoi și v_{oo}
(nodurile) | $n_v + 1 = ?$

muchiiile : muchiile diagramei Voronoi
 $n_m = ?$

fetele : siturile interale
 m

(23)

11/6

• Relația lui Euler:

$$(n_v + 1) - n_m + n_f = 2$$

• Incidente: muchii / vârfuri

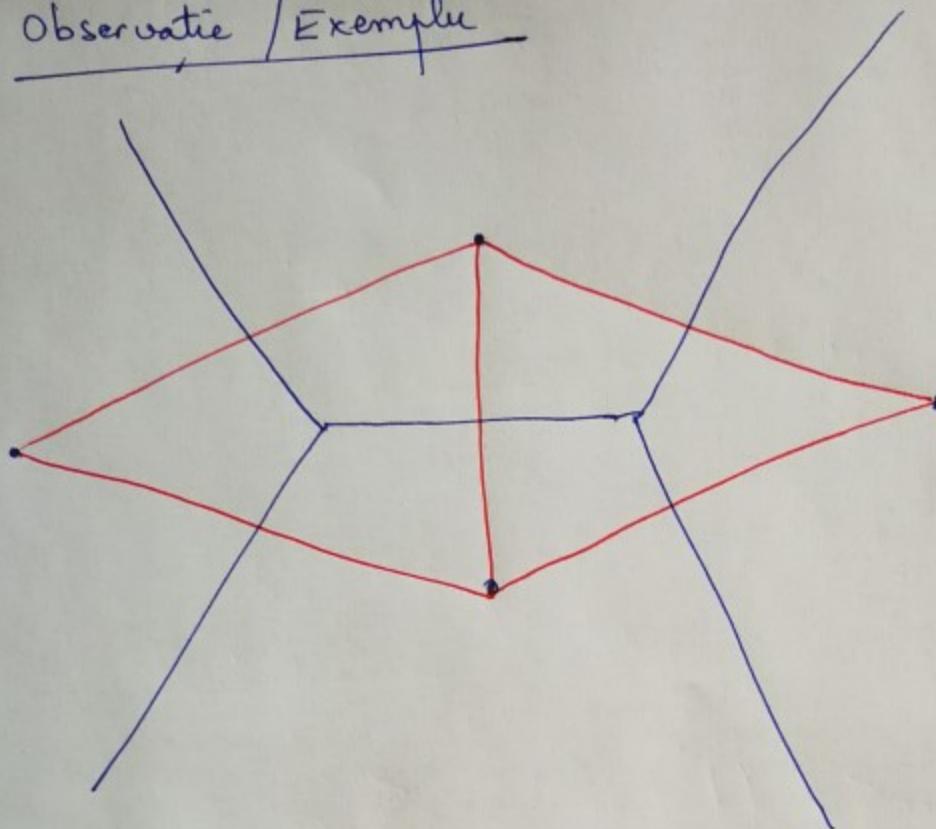
- fiecare muchie este incidentă cu exact
două vârfuri

- fiecare vârf (inclusiv v_0) este incident
cu cel puțin 3 muchii

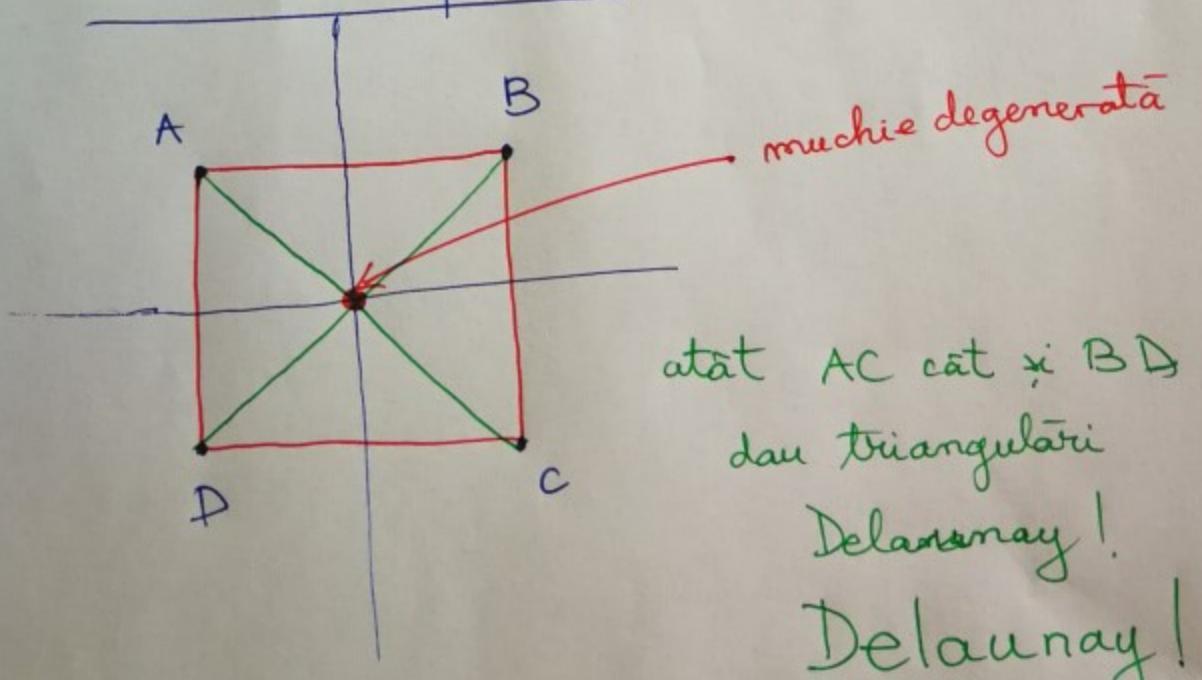
$$2 \cdot n_m \geq 3(n_v + 1) \Rightarrow \dots \Rightarrow \text{ctd.}$$

Întrebare: când avem egalitate, adică
 $n_v = 2n_m - 5$, $n_m = 3n_v - 6$?

Observatie / Exemplu



Intrebare / Exemplu

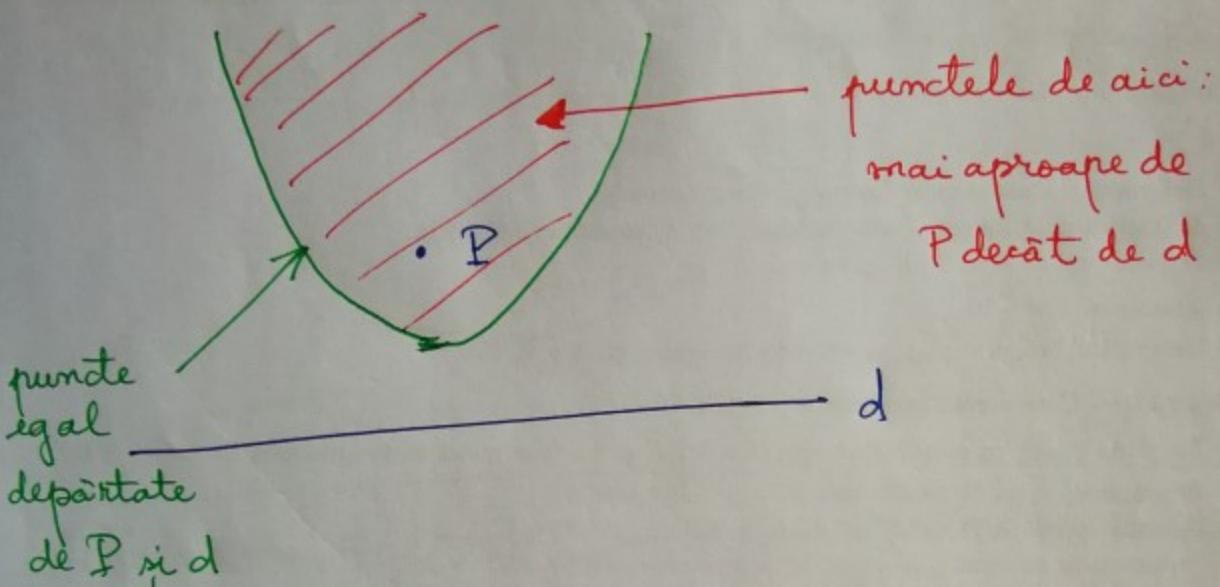


muchie degenerată
atât AC cât și BD
dau triunghiuri
Delaunay!
Delaunay!

Despre curba parabolică ("beach line")

- reuniune de arce de parabolă

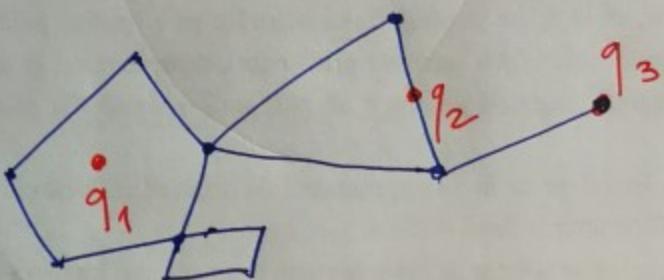
- de ce parabolă?



Probleme de localizare

(Point location)

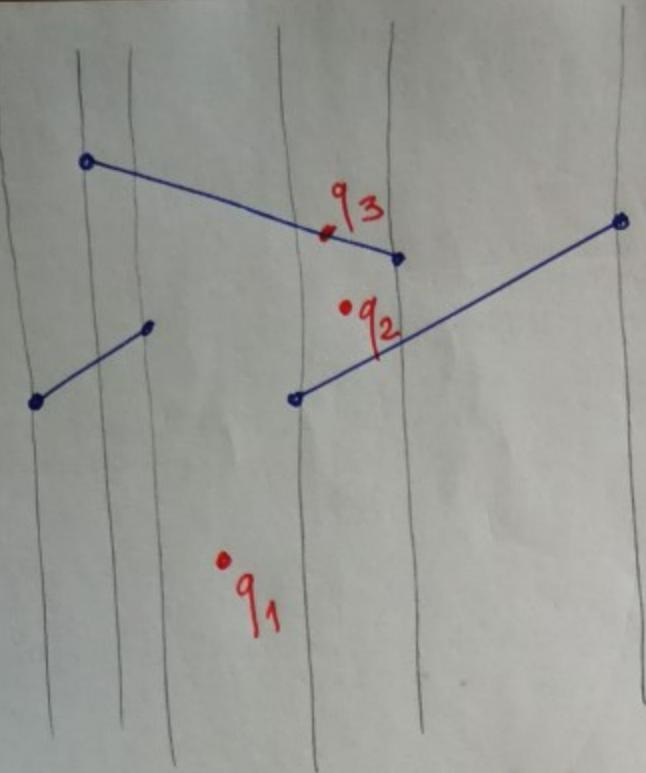
Motivatie: date o subdiviziune planară și un punct q (query), să se determine poziția lui q în raport cu subdiviziunea



Subdiviziune a planului în fascii verticale (benzi)

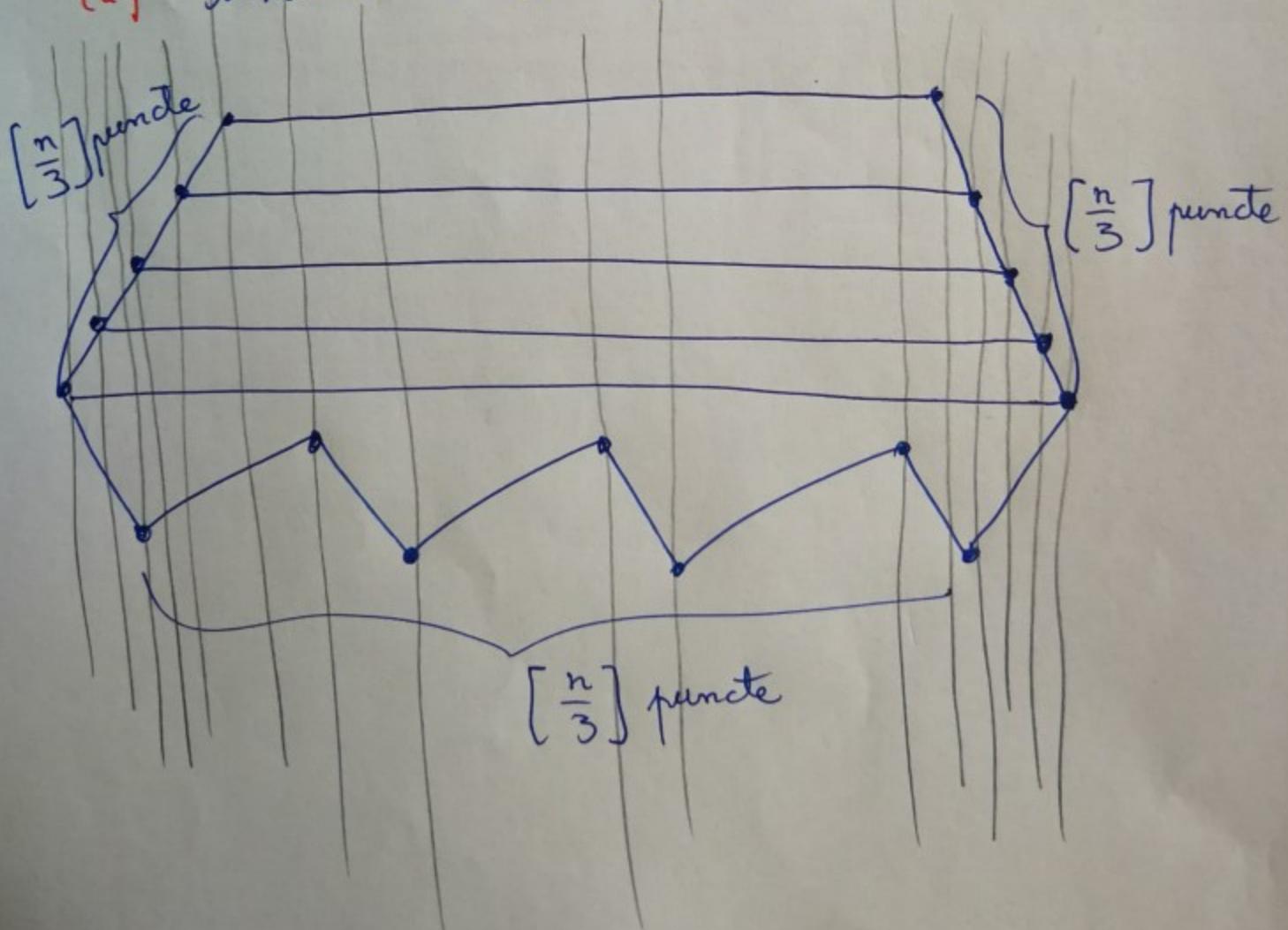
"slab"

- căutarea după abscisă → pentru identificarea fasciei verticale;
- căutarea în cadrul unei fascii, fiind indicat elementul subdiviziunii care conține punctul q
 - ⚠ se realizează în raport cu segmente.



obs. (i) Identificarea fâșiei verticale: timp de căutare $O(\log n)$.

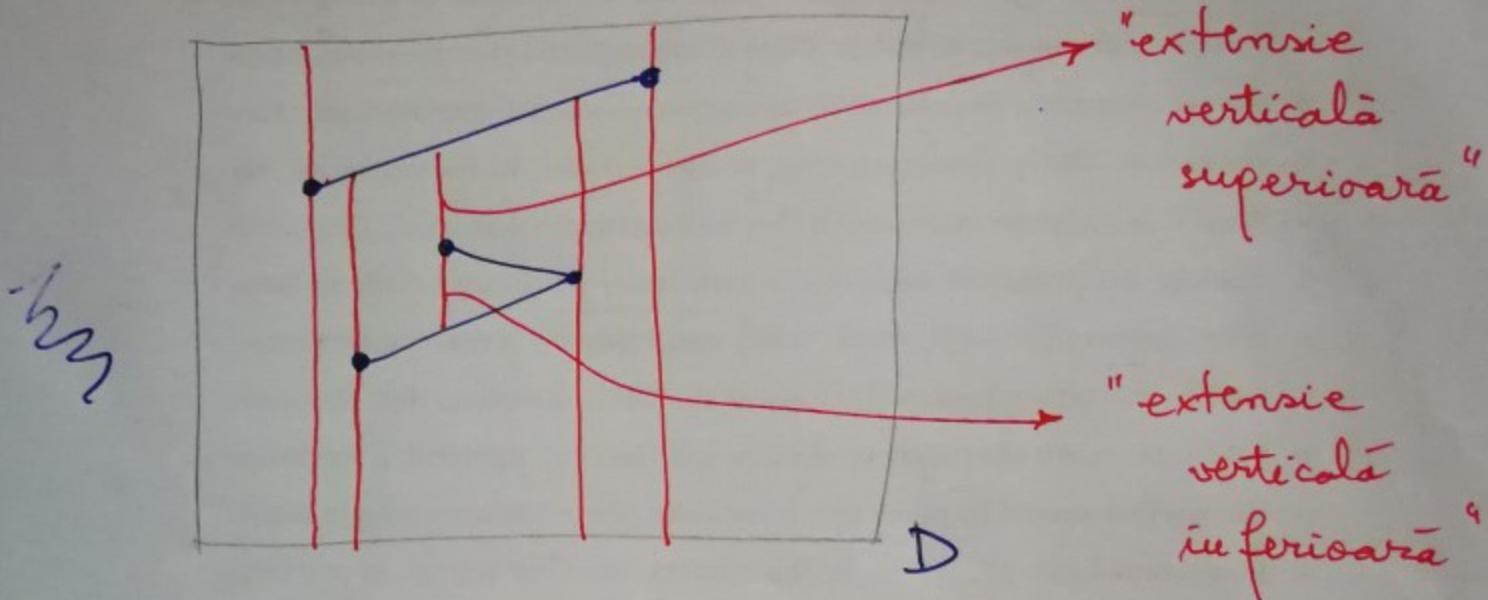
(ii) Memorie necesară: uneori $O(n^2)$.



Hartă trapezoidală a unei subdiviziuni

planare

(continuare ; cf. curs anterior).

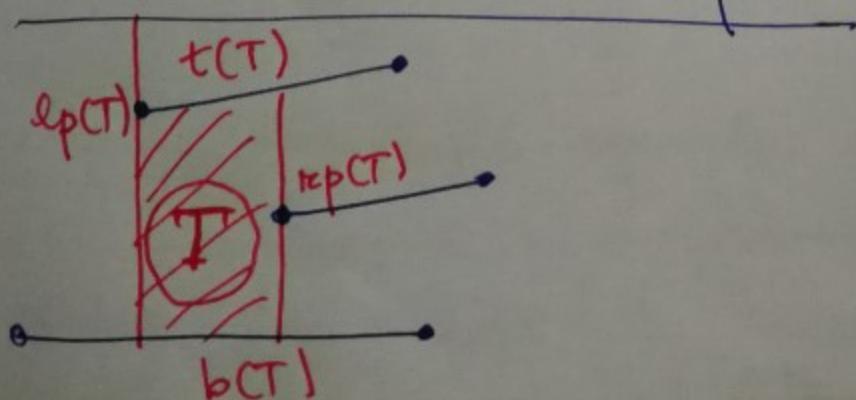


Pentru fiecare vârf al subdiviziunii sunt considerate două "extensi" verticale (superioară / inferioră)

care se "opresc" atunci când este atâlnit un alt segment sau $D \Rightarrow$

harta trapezoidală : { - S
- D
- extensiile .

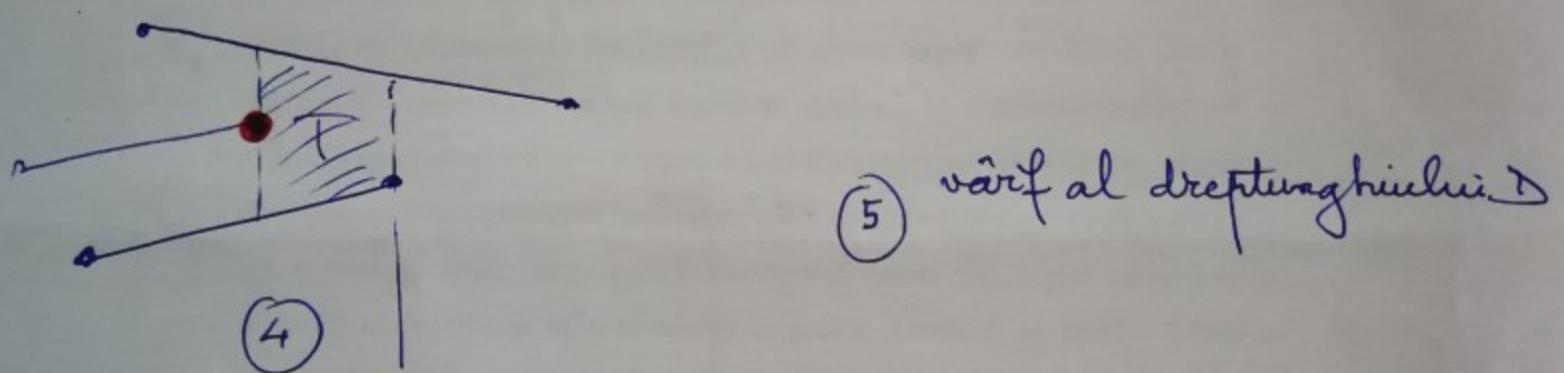
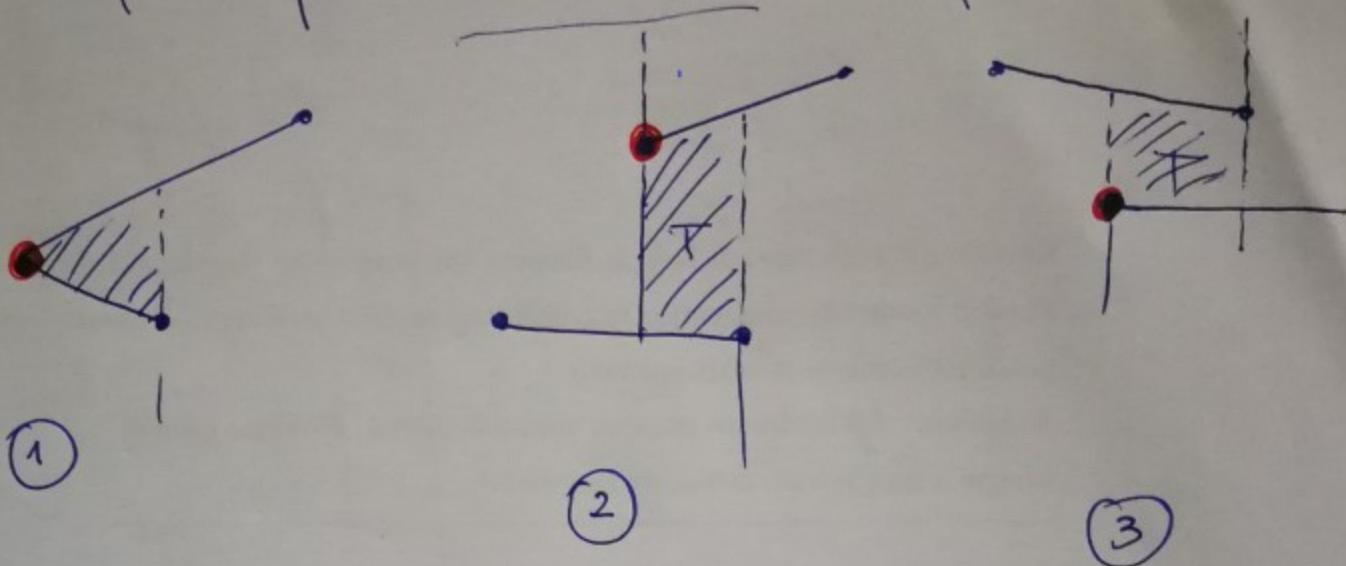
Cum este "memorat" un trapez ?



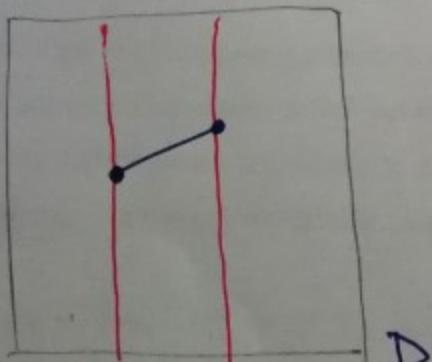
(23)

13/1

Obs. Sunt posibile configurații pentru un vîrful $\text{lp}(T)$ asociat unei trapeze T :



Exemplu (referitor la numărul de trapeze/vârfuri asociate unei hărți trapezoïdale).



$$\underline{n = 1}$$

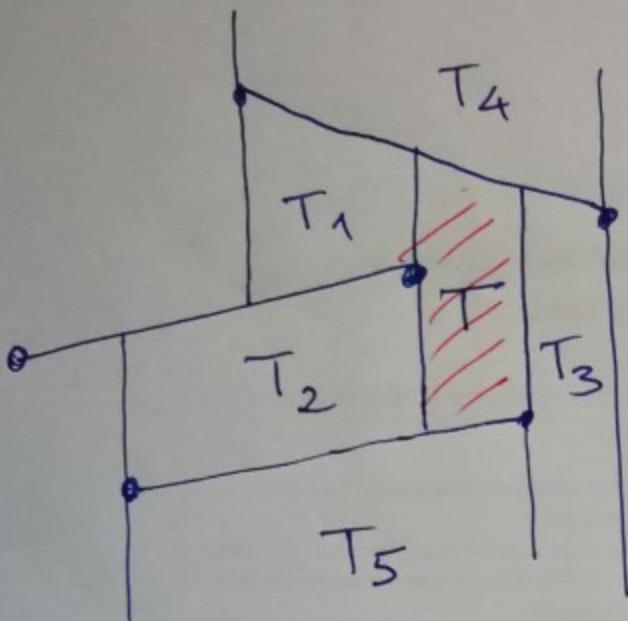
$$4 \text{ trapeze } (= 3n + 1)$$

$$10 \text{ vârfuri } (= 6n + 4)$$

Exercițiu Dati și alte exemple de hărți trapezoïdale, în care se realizează / nu se realizează nr. maxim de vârfuri / de trapeze.

(23)

Exemplu - adiacente trapezelor



T este adiacent

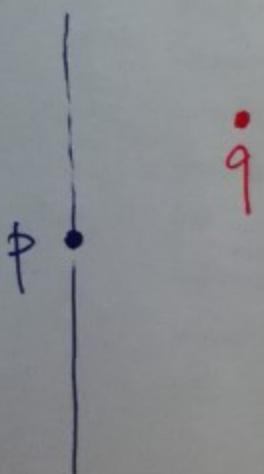
cu T_1, T_2, T_3

mu este adiacent

cu T_4, T_5

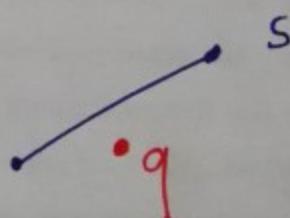
Căutarea în grafel asociat unei hărți trapezoidele

x-nod (p)



q este la stânga sau la
dreapta dreptei verticale
care trece prin p
(comparatie de abscise)

y-nod (s)

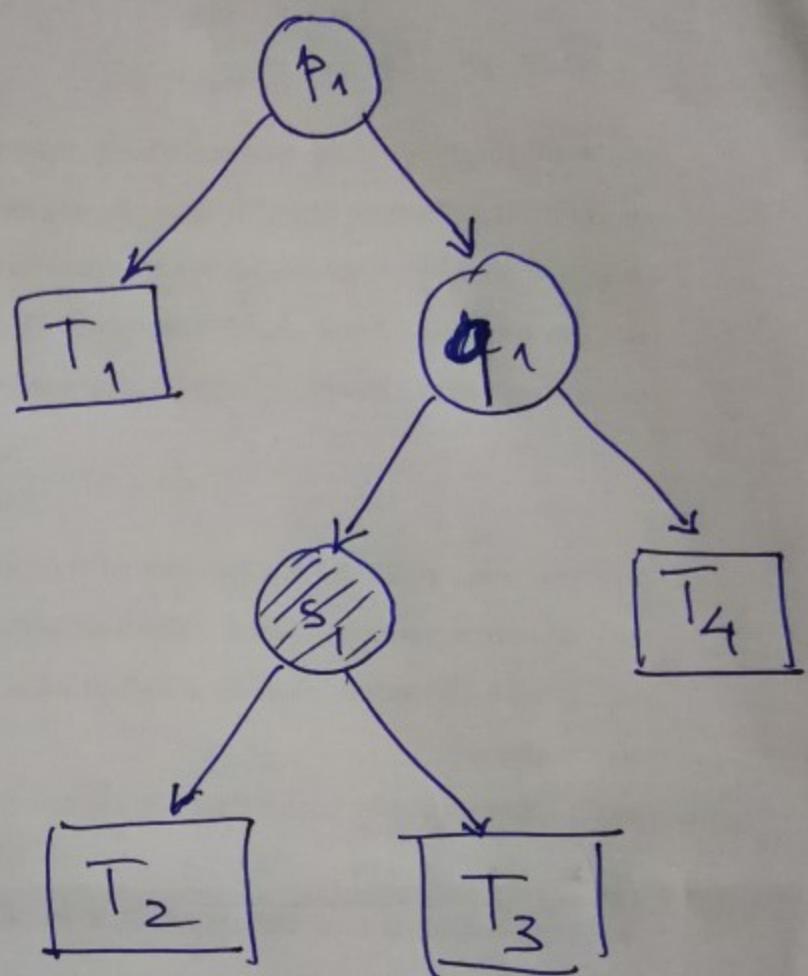
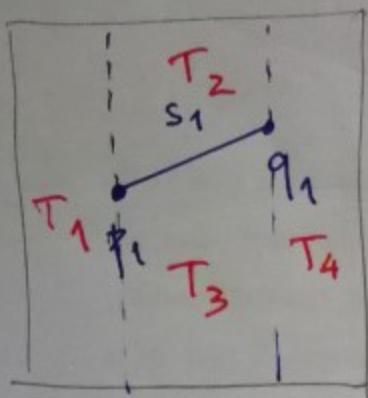


q este deasupra / de dedesubtul
lui s
(testul de orientare)

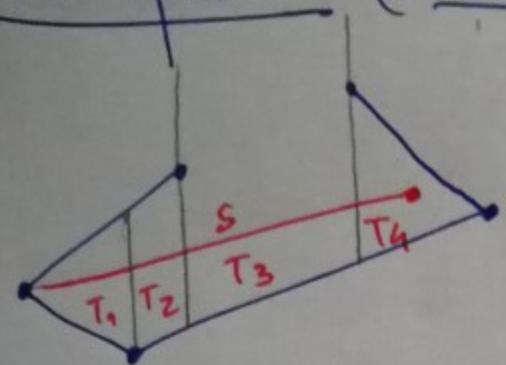
(23)

13/3

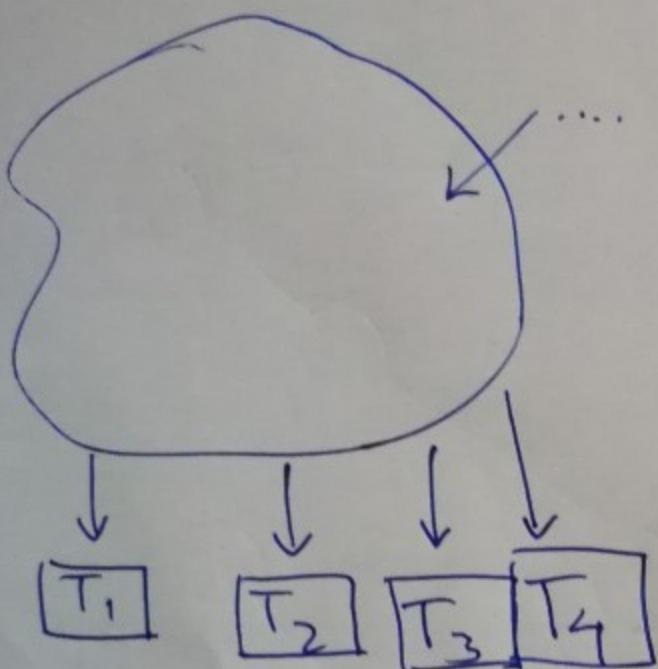
Exemplul 1 (structură de căutare asociată)



Exemplul 3 (Exercitiu)

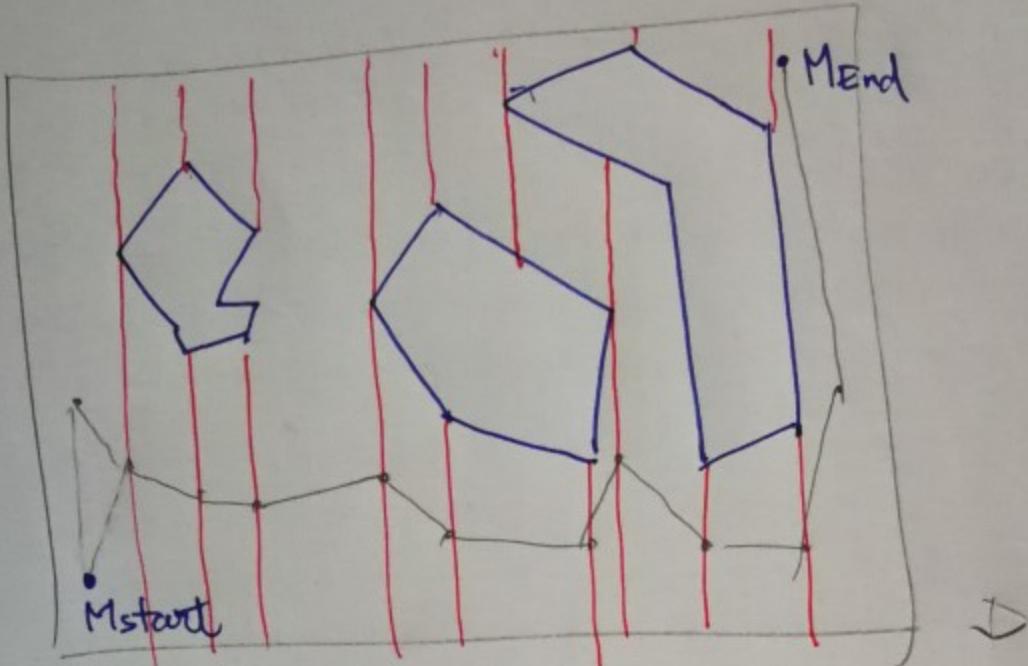


?) cum se modifică struct.
de căutare dacă este
inserat s?



?) cum se actualizează

Aplicatie a hartilor trapezoidale



dorim să deplasăm un robot (punct) din M_{start} în M_{end} ; robotul se deplasează prin translată

Pas 1. "determinarea spațiu lui liber" \Rightarrow

harta trapezoidală a spațiului liber C_e , rotată cu $T(C_e)$.

Pas 2. date M_{start} , M_{end} se căută un drum în interiorul lui C_e

- dacă sunt în același trapez: ✓
- dacă sunt în trapeze diferite: se folosesc centrele de greutate ale trapezelor în care sunt situate punctele și mijloacele laturilor (parallele) ale trapezelor adiacente \rightarrow muchii verticale?

Se utilizează un graf ~~—~~ asociat C_e , care poate fi construit din $T(C_e)$ în timp liniar (de ce?)

(23)

13/7

Examen:

- pe 31.01, cf. programării ; repartizare pe sali → v.moodle
- pe 30.01 (?) consultări → v.moodle

-
- lucrare scrisă (60p/100p) ; timp de lucru $1\frac{1}{2}$ h - 2 h.
 - cu "cartile pe masă" **FĂRĂ** resurse electronice
 - fiecare lucrare: originală

NU COPIAȚI !

NU îți AJUTAȚI PE L

—

COLEGi SĂ COPIEZE !

↳ Regulamentul de Etică !

Subiecte: 7 subiecte

① (5p) Aplicatie elementara din capitolul de
preliminarii (produs vectorial, raport, test de
orientare ...).

② (10p)

③ (10p).

④ (10p) Complexitatea algebraica

Exemplu: In \mathbb{R}^2 sunt date trei drepte, prin
ecuatiiile lor generale. Care este complexitatea
algebraica a calculelor pentru a stabili:

- daca cele trei drepte au exact un
punct comun;
- coordonatele unicului punct de
intersectie (se sp. ca el exista).

a) Fie $d: ax + by + c = 0$

$$d': a'x + b'y + c' = 0$$

$$d'': a''x + b''y + c'' = 0.$$

Date de intrare: $a, b, c, a', b', c', a'', b'', c''$

Condiția de concurență (într-un singur punct):

$$\begin{vmatrix} a & b & c \\ a' & b' & c' \\ a'' & b'' & c'' \end{vmatrix} = 0. + \text{minor de ord. II} \neq 0,$$

polinom de gradul III.

b) Pp. că a) este verificat. Coordonatele punctului de intersecție sunt date de soluțiile sistemului
(de ex.)

$$\begin{cases} ax + by + c = 0 \\ a'x + b'y + c' = 0 \end{cases}$$

$$\Delta = \begin{vmatrix} a & b \\ a' & b' \end{vmatrix} \rightarrow \text{Pol II}$$

$$\Delta_x = \begin{vmatrix} -c & b \\ -c' & b' \end{vmatrix} \rightarrow \text{Pol II}$$

$$x = \frac{\Delta_x}{\Delta} = \frac{\text{Pol II}}{\text{Pol II}} ; \text{ analog ptr. } y.$$

(24)

⑤ (10p) Algoritmii discutati

- background logic / matematic;
- complexitatea timp / spatiu.

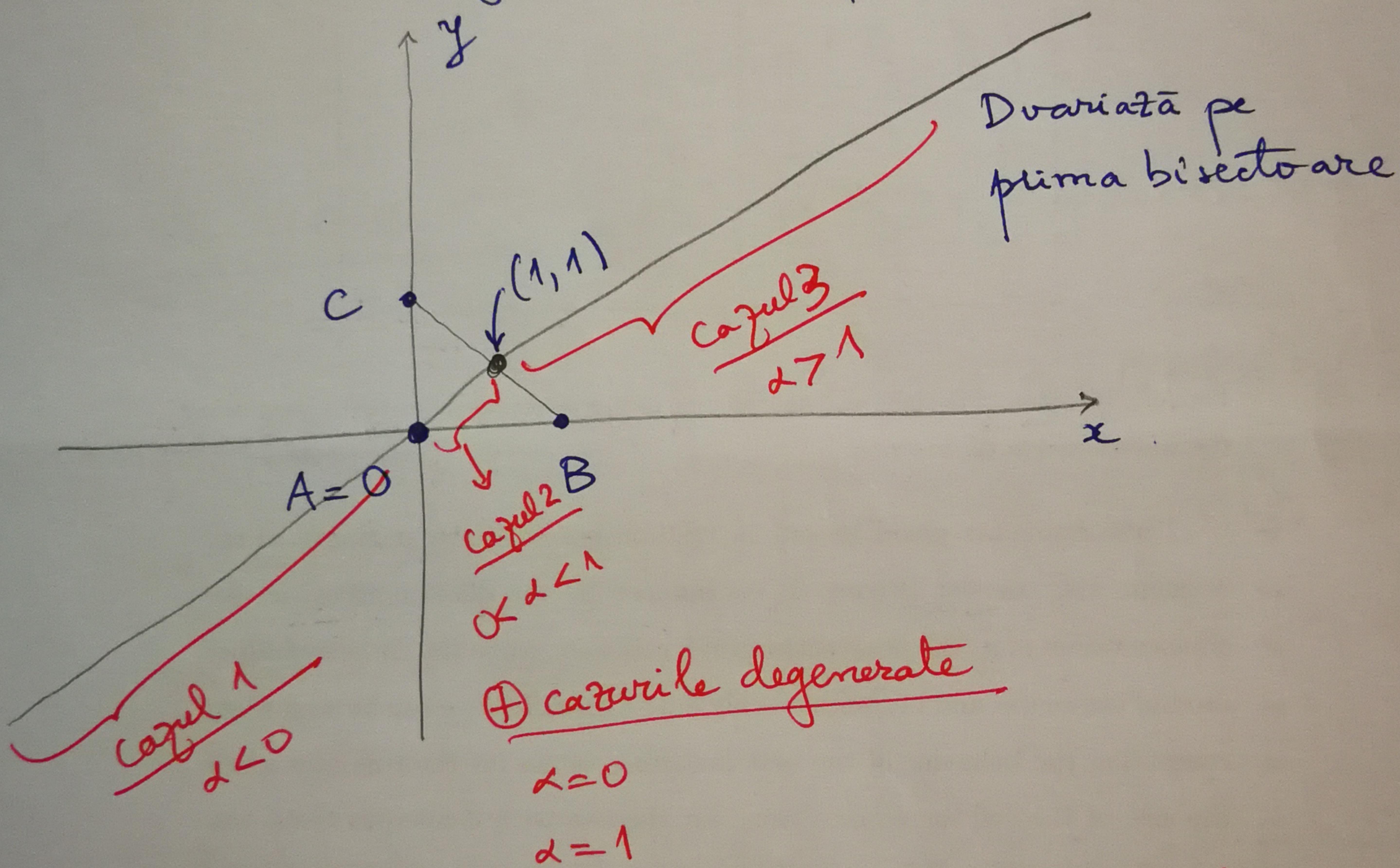
Exemplu:

- la Graham's scan, varianta Andrew: unde intervine numod esential ordonarea punctelor?
- de ce la hartiile trapezoidale sunt preferate "extensii verticale" (superioare si inferioare) dreptelor verticale duse pînă la extremitătile segmentelor / vîrfuri?
- de justificat (scurt, concis, la obiect)
 - un algoritm discutat la curs (o parte a sa);
 - un pas/ anumiti pasi ai unui algoritm din supertul de curs;
 - un algoritm indicat explicit in subiect.

⑥ (10p) "Algoritm / Problema cu parametri"

Transferul unei probleme geometrice într-un algoritm.

Exemplu. Fie $A = (0,0)$; $B = (2,0)$; $C = (0,2)$; $D = (\alpha, \alpha)$, cu $\alpha \in \mathbb{R}$. Scrieți un algoritm care să indice vârfurile acoperirii convexe a mulțimii $\{A, B, C, D\}$ și centrul de greutate al poligonului convex asociat.



Se puntează:

- ⊗ figura
 - ⊗ înțelegerea contextului geometric
 - ⊗ calcul (geometrice)
 - ⊗ cazuri degenerate
 - + interpretare
 - ⊗ selectarea corectă a input-urilor
- (24) 14/5

Detalii:

- ptr. $\alpha < 0$: acoperirea convexă: D, B, C,
centrul de greutate ΔDBC : $(\frac{\alpha+2}{3}, \frac{\alpha+2}{3})$
- ptr. $\alpha = 0$
casă degenerat: A=D (interpretare: două puncte coincid)
a.c.: A, B, C;
c.g. ΔABC : $(\frac{2}{3}, \frac{2}{3})$
- ptr. $\alpha \in (0, 1)$ a.c.: A, B, C; c.g. ΔABC : $(\frac{2}{3}, \frac{2}{3})$
- ptr. $\alpha = 1$: a.c. A, B, C; c.g. ΔABC : $(\frac{2}{3}, \frac{2}{3})$
casă degenerat: D ∈ [BC] (interpretare: D este situat
pe frontieră acoperirii convexe, fără să fie vîrf
al acestora)
- ptr. $\alpha > 1$: a.c.: A, B, C, D c.g. al patrulaterului
ABDC este $(\frac{\alpha+2}{4}, \frac{\alpha+2}{4})$.

Algoritmul : | Input: α
Output: calculele de mai sus.

7 (5p) ...

(24)
14/6.