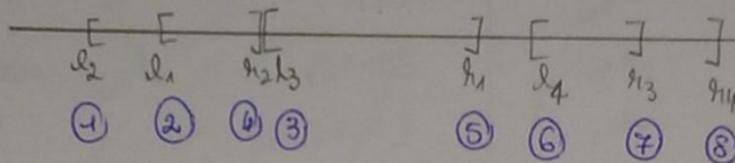


GEOMETRIE COMPUTACIONALĂINTERSECȚII DE SEGMENTE- Rezolvarea problemei în context 1D -

- Segmentele $[l_i, h_i]$, $i = \overline{1, m}$ date
- Ordonarea lexicografică a extremităților segmentelor

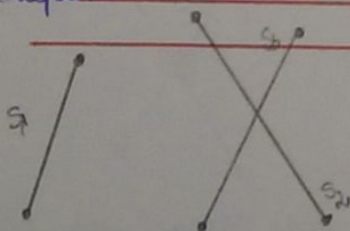
Exemplu

De fapt este ordonată mulțimea ($\dim \mathbb{R}^2$) în care coordonatele sunt de forma $\{(l_i, 0) \mid i = \overline{1, m}\} \cup \{(h_i, 1) \mid i = \overline{1, m}\}$

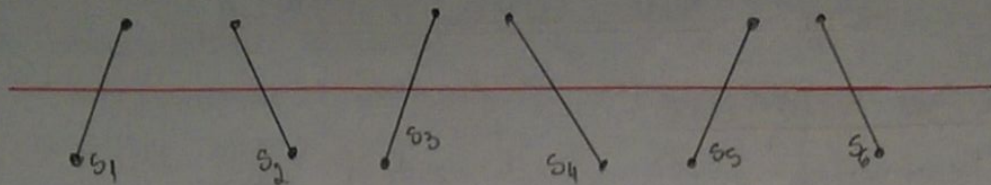
Algoritmul implică o listă L a segmentelor active (este statut) care este modificată la fiecare punct (extremitate = eveniment)

În exemplu: statut

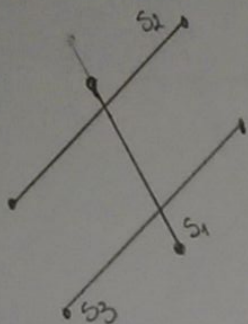
\emptyset		
① $\{s_2\}$	④ $\{s_1, s_3\}$	⑦ $\{s_4\}$
② $\{s_1, s_2\}$	⑤ $\{s_3\}$	⑧ \emptyset
③ $\{s_1, s_2, s_3\}$	⑥ $\{s_3, s_4\}$	

- Rezolvarea problemei în context 2D -Exemplu d_1 : statut \emptyset d_2 : statut $\{s_2, s_3\}$ d_3 : statut $\{s_4\}$

Metoda este încă insuficientă.



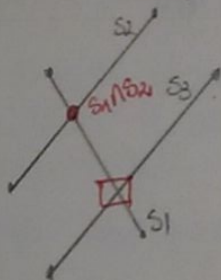
Ex 4) Var 1 În care statutul dreptei de balere este o mulțime neordonată



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

Am notat cu
 μ_{s_i} = marginea sup.
 a lui s_i
 d_{s_i} = marginea inf.
 a lui s_i

VAR 2 În care statutul dreptei de balere este o mulțime ordonată



Eveniment	Statut
μ_{s_2}	\emptyset
μ_{s_1}	(s_2)
μ_{s_3}	(s_1, s_2)
d_{s_2}	(s_2, s_1)
d_{s_1}	(s_2, s_1, s_3)
d_{s_3}	(s_1, s_3)
μ_{s_2}	(s_2, s_1)
μ_{s_1}	(s_3)
d_{s_3}	\emptyset

calculează $s_1 \cap s_2$
 DA!

calculează $s_1 \cap s_3$
 DA!

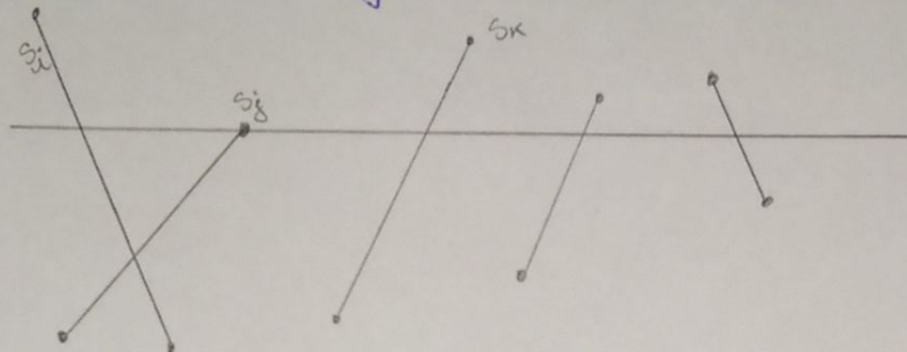
PRINCÍPIU (de urmărit la fiecare tip de eveniment)

→ Schimbarea statutului dreptei de balieră;

→ Verificarea intersecțiilor dintre vecinii nou apăruiți în statut

① la MARGINI SUPERIOARE ("UPPER POINTS")

- este inserat segmentul a cărei margine superioară a fost detectată
(dacă sunt mai multe segmente inserarea se face în ordinea adecvată!)



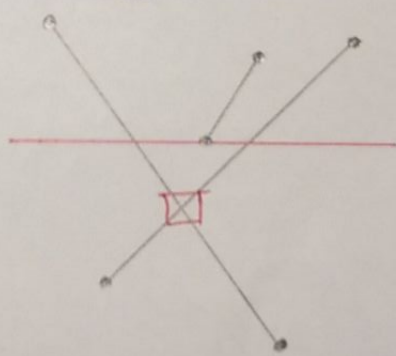
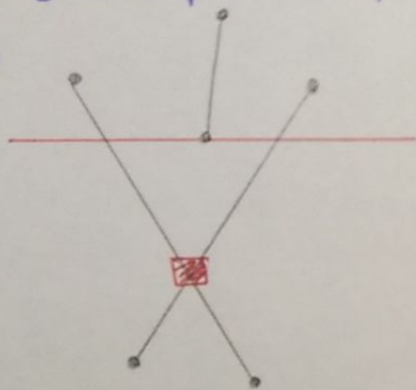
• Pt. evenimentul u_{s_j} (marginea superioară a lui s_j) efectuăm teste de intersecție ~~pentru~~, deoarece să

s_j și s_i se intersectează
 s_j și s_k ! nu se intersectează → **NOU
EVENIMENT**

② la MARGINI INFERIOARE ("DOWN POINTS")

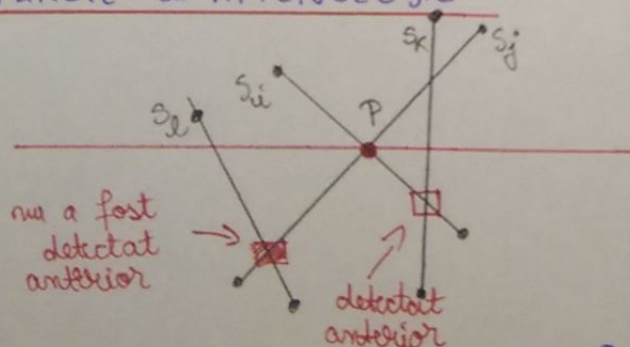
→ este șters din statut segmentul a cărei margine inferioară a fost detectată și sunt testate segmentele care devin vecine. Există două posibilități:

① dau peste un punct de intersecție nedetectat anterior



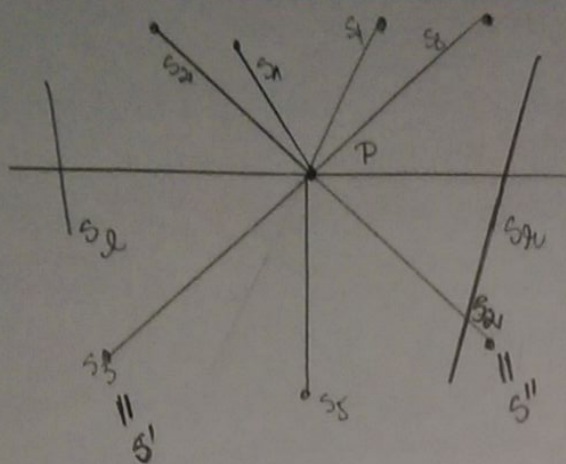
② dau peste un punct de intersecție detectat anterior

③ PUNCTE de INTERSECȚIE



La evenimentul $P(s_i \cap s_j)$ avem
→ evenimentul " $s_i \cap s_j$ " este detectat
→ evenimentul " $s_i \cap s_k$ " a fost detectat anterior deoarece s_i și s_k au fost vecini în statutul dreptei de balieră

ALGORITM de INTERSECȚII



$$U(p) = \{s_5\}$$

$$D(p) = \{s_1, s_4\}$$

$$Int(p) = \{s_2, s_3\}$$

Statutul : (s_2, s_1, s_4, s_3) Anunț de evenimentul P

Statutul (s_3, s_5, s_2) după evenimentul P