

Geometria Computacional

Cristina G. Fernandes

Departamento de Ciência da Computação do IME-USP

`http://www.ime.usp.br/~cris/`

segundo semestre de 2018

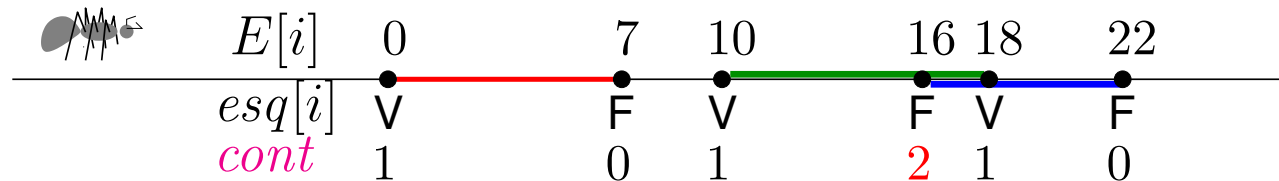
Método da linha de varredura

Ideia: reduzir um problema estático bidimensional a um problema dinâmico unidimensional

Método da linha de varredura

Ideia: reduzir um problema estático bidimensional a um problema dinâmico unidimensional

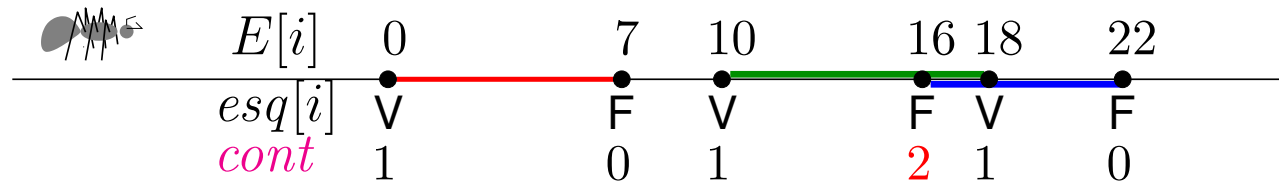
Uma **linha imaginária** move-se da esquerda para a direita.



Método da linha de varredura

Ideia: reduzir um problema estático bidimensional a um problema dinâmico unidimensional

Uma **linha imaginária** move-se da esquerda para a direita.

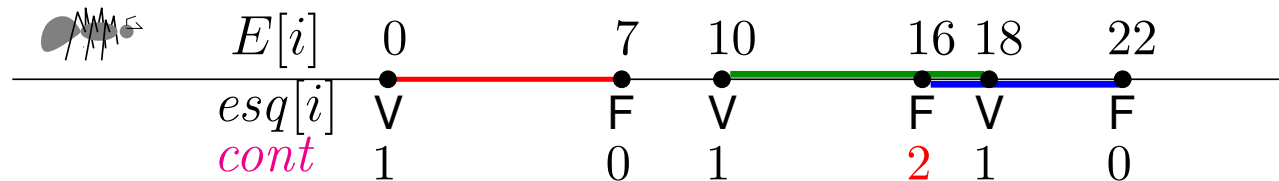


À medida que ela move,
o problema restrito à esquerda dela é resolvido.

Método da linha de varredura

Ideia: reduzir um problema estático bidimensional a um problema dinâmico unidimensional

Uma **linha imaginária** move-se da esquerda para a direita.



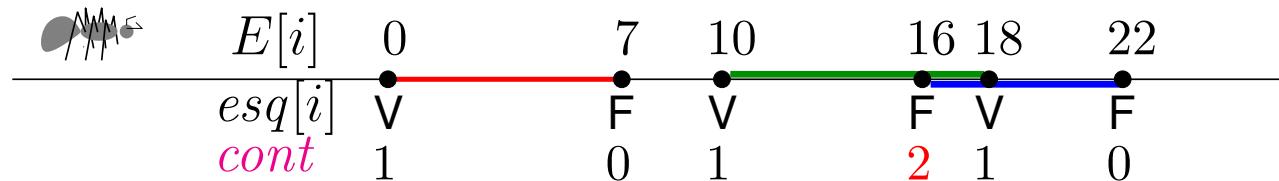
À medida que ela move,
o **problema restrito à esquerda dela é resolvido.**

Informação necessária para estender a solução parcial é mantida numa **descrição combinatória da linha.**

Método da linha de varredura

Ideia: reduzir um problema estático bidimensional a um problema dinâmico unidimensional

Uma **linha imaginária** move-se da esquerda para a direita.



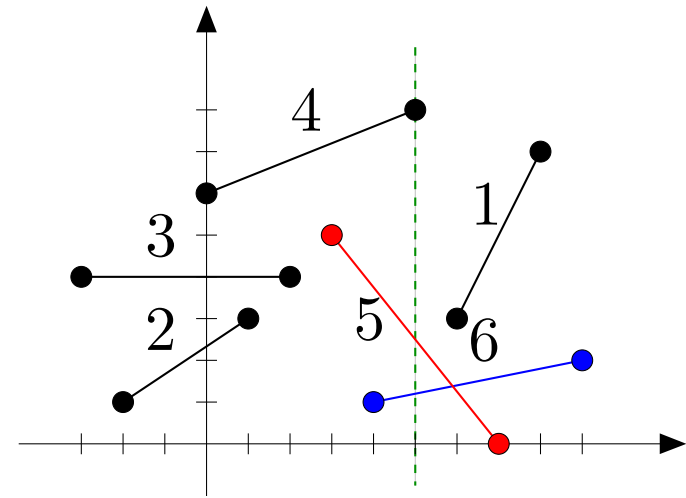
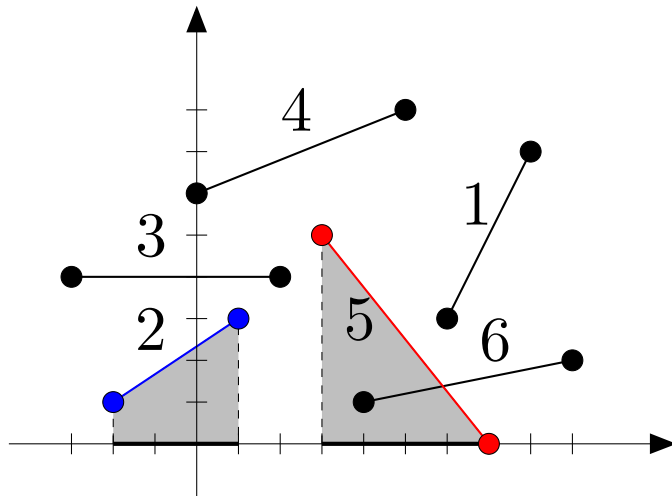
À medida que ela move,
o **problema restrito à esquerda dela é resolvido.**

Informação necessária para estender a solução parcial é mantida numa **descrição combinatória da linha.**

Muda apenas em posições chaves: os **pontos eventos.**

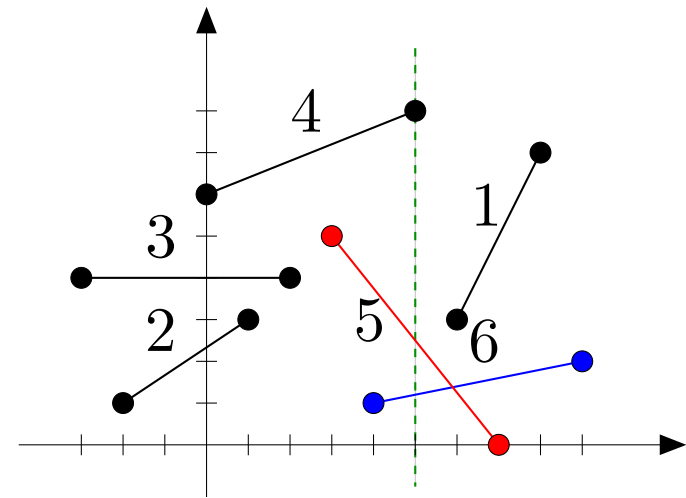
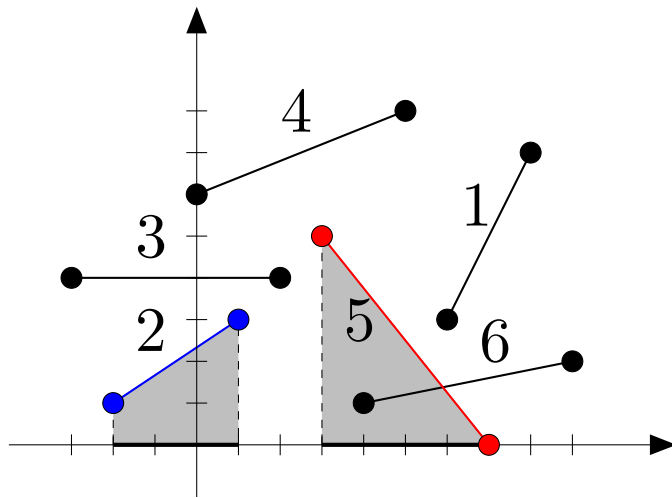
Algoritmo de Shamos e Hoey

Ideia: Dois segmentos cuja projeção no eixo X sejam disjuntas não se intersectam.



Algoritmo de Shamos e Hoey

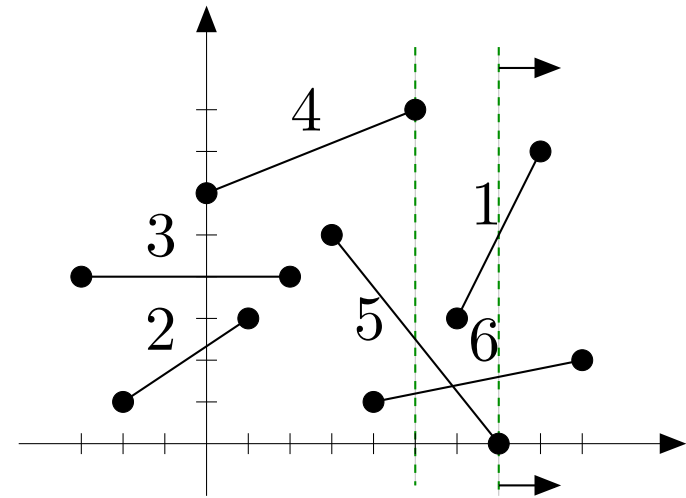
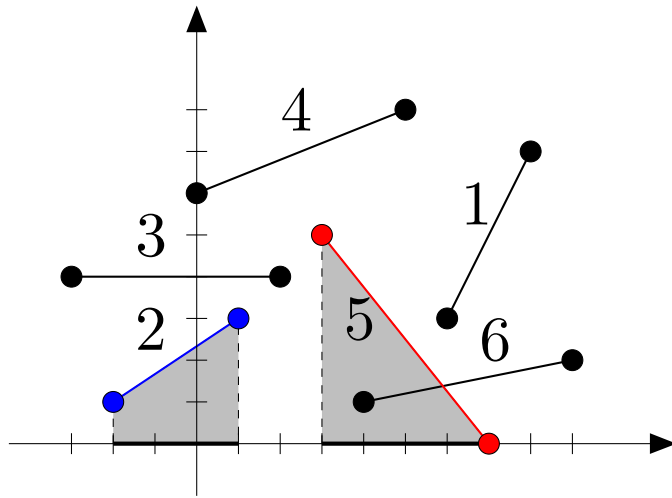
Ideia: Dois segmentos cuja projeção no eixo X sejam disjuntas não se intersectam.



Se a projeção no eixo X de dois segmentos tem interseção, então há uma **linha vertical** que intersecta ambos.

Algoritmo de Shamos e Hoey

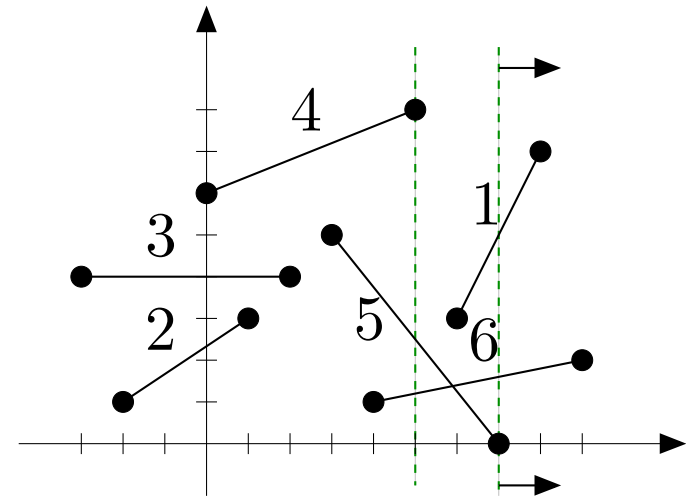
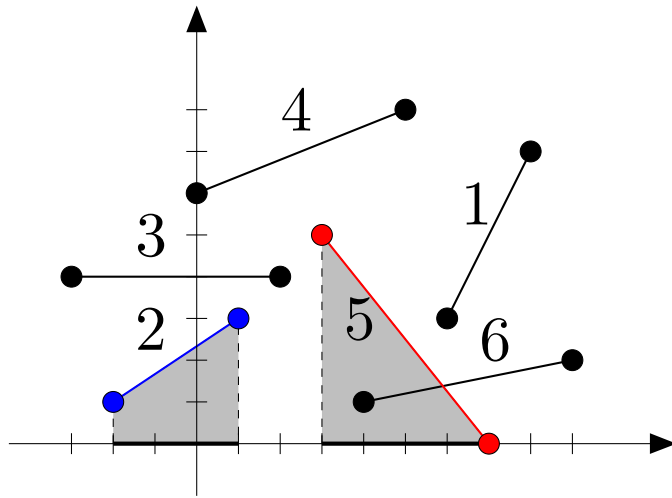
Ideia: Dois segmentos cuja projeção no eixo X sejam disjuntas não se intersectam.



Imagine esta **linha vertical** varrendo o plano da esquerda para a direita...

Algoritmo de Shamos e Hoey

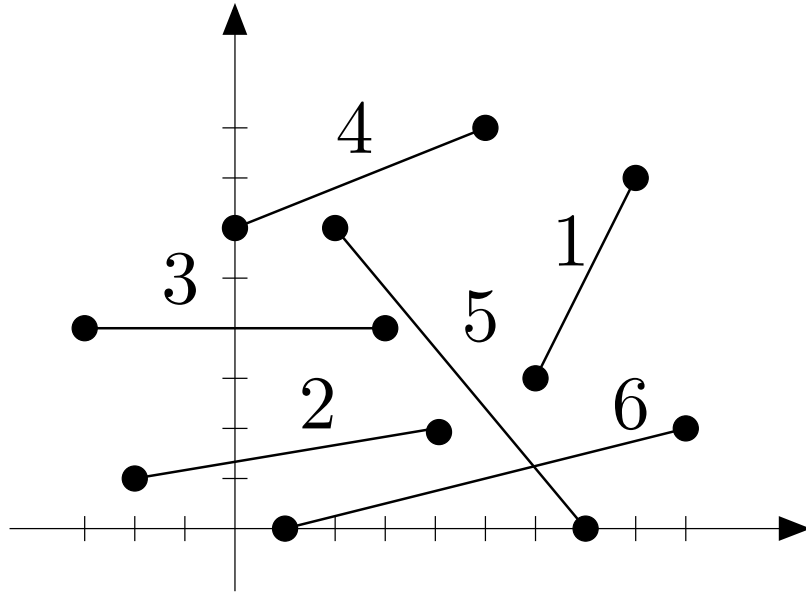
Ideia: Dois segmentos cuja projeção no eixo X sejam disjuntas não se intersectam.



Imagine esta **linha vertical** varrendo o plano da esquerda para a direita...

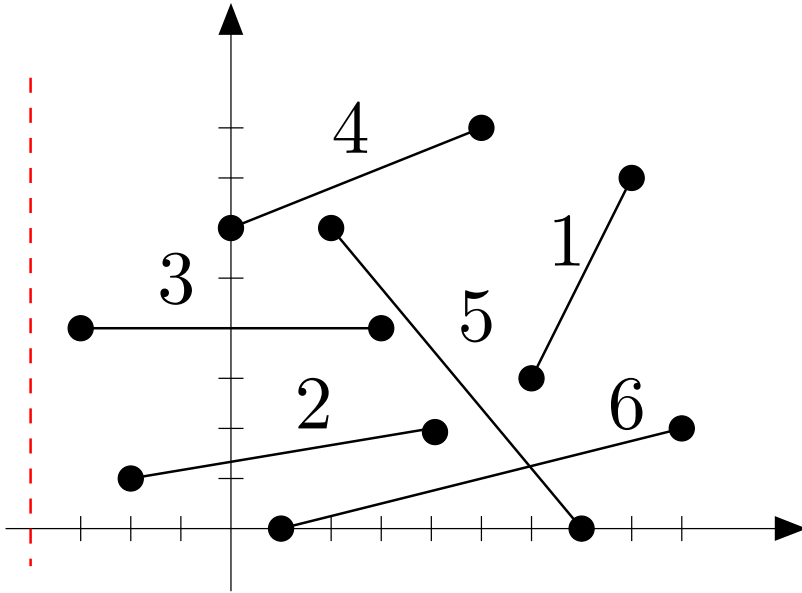
Enquanto a **linha** varre o plano, mantemos os segmentos intersectados por ela na **descrição combinatória da linha**.

Descrição combinatória da linha



$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

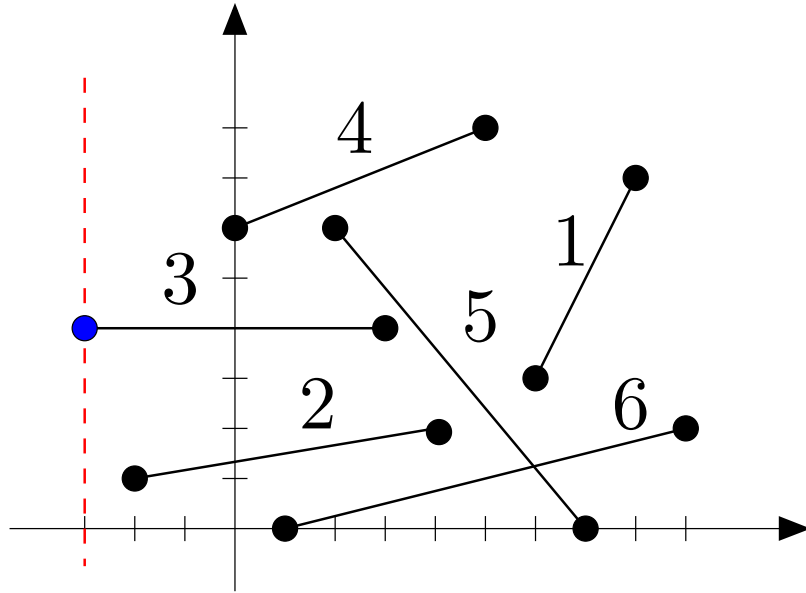
Descrição combinatória da linha



Alterações ocorrem
nos **extremos dos segmentos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

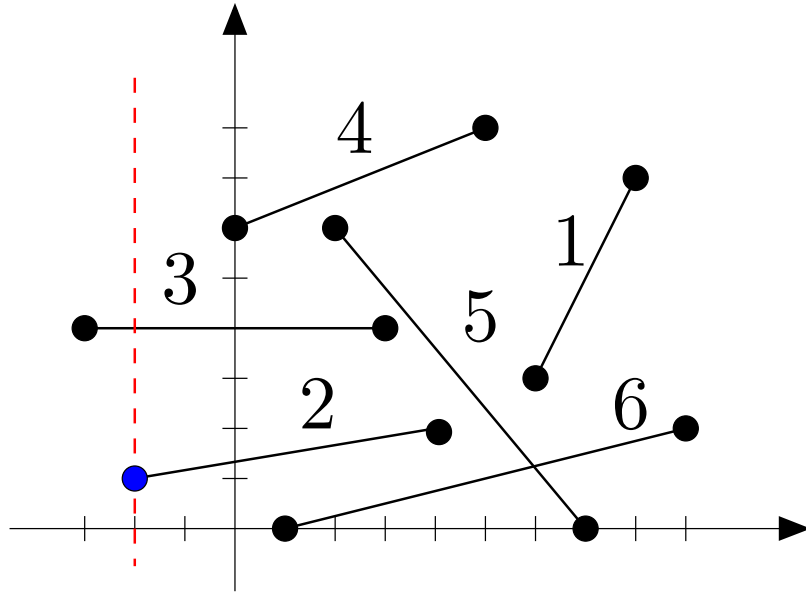


Alterações ocorrem nos **extremos dos segmentos**.

Estes são os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

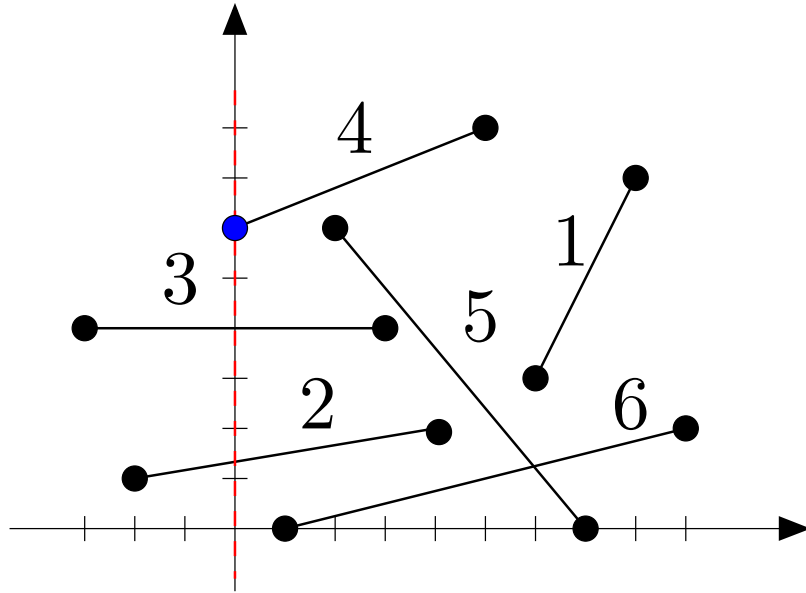


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

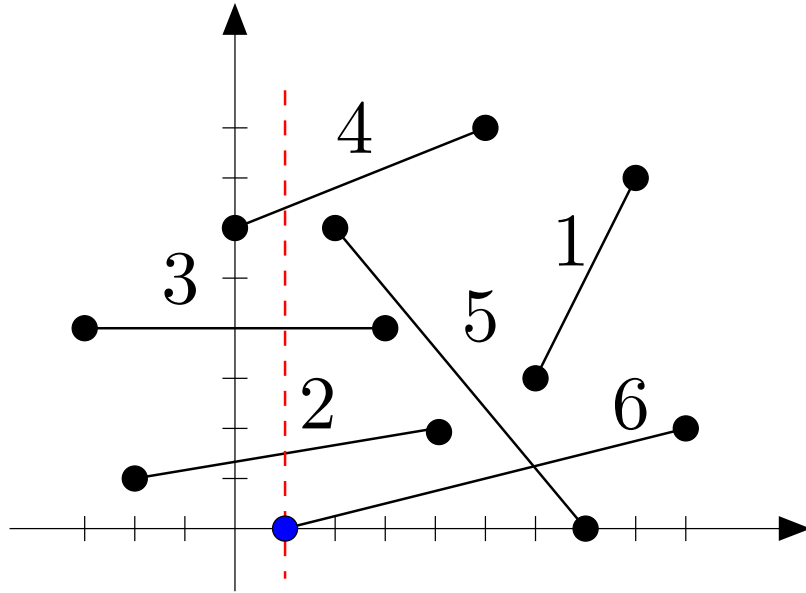


Alterações ocorrem nos **extremos dos segmentos**.

Estes são os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

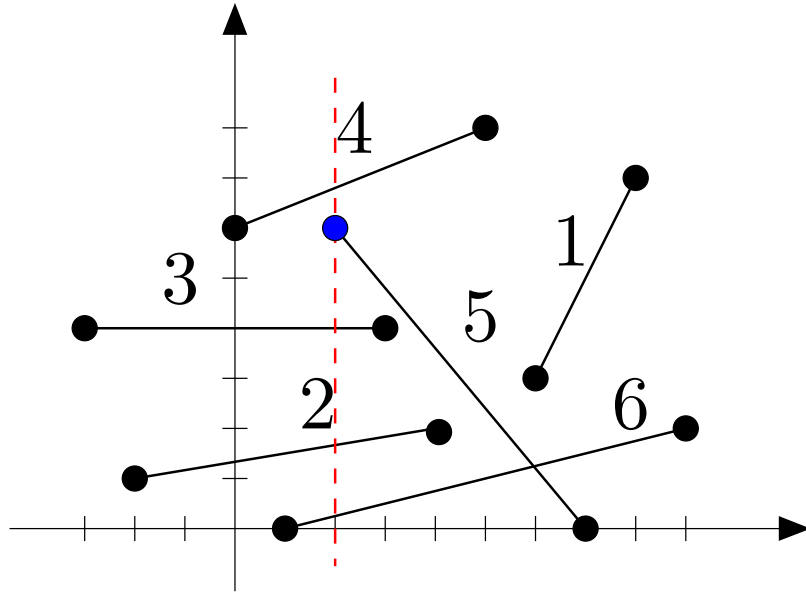


Alterações ocorrem nos **extremos dos segmentos**.

Estes são os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

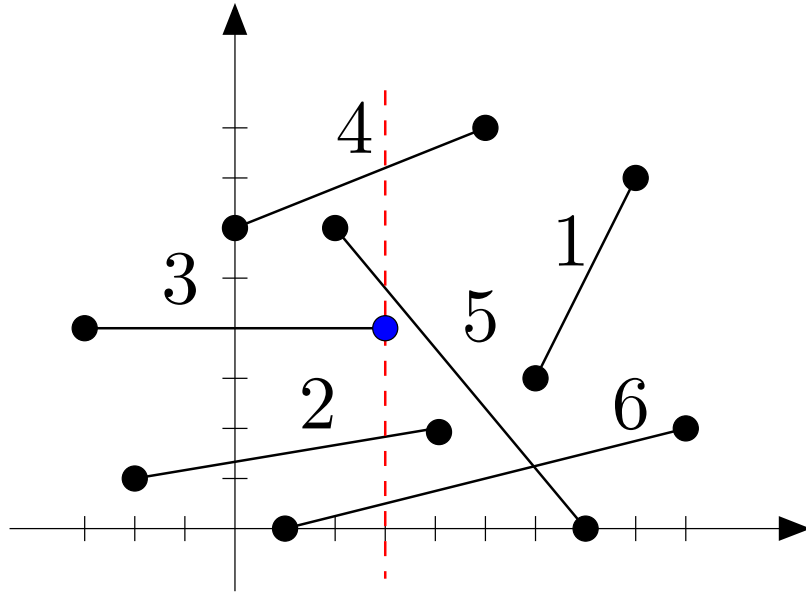


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

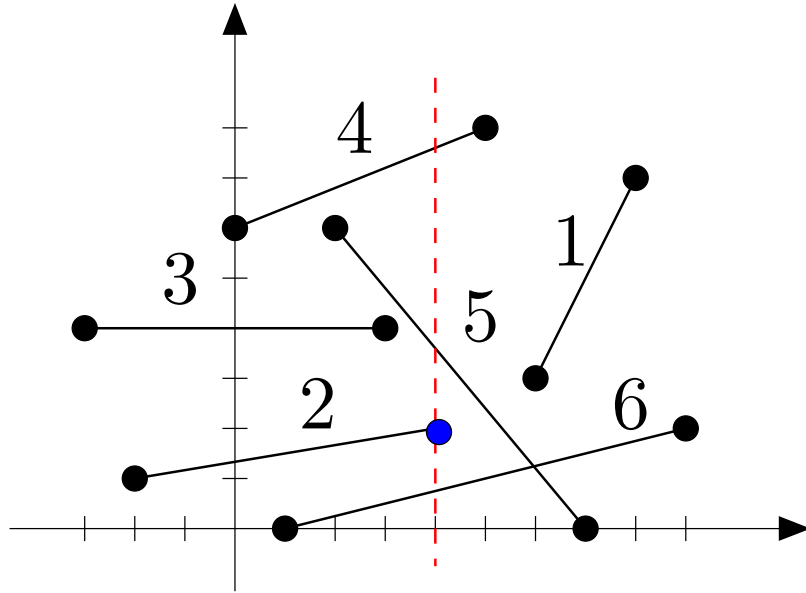


Alterações ocorrem nos **extremos dos segmentos**.

Estes são os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

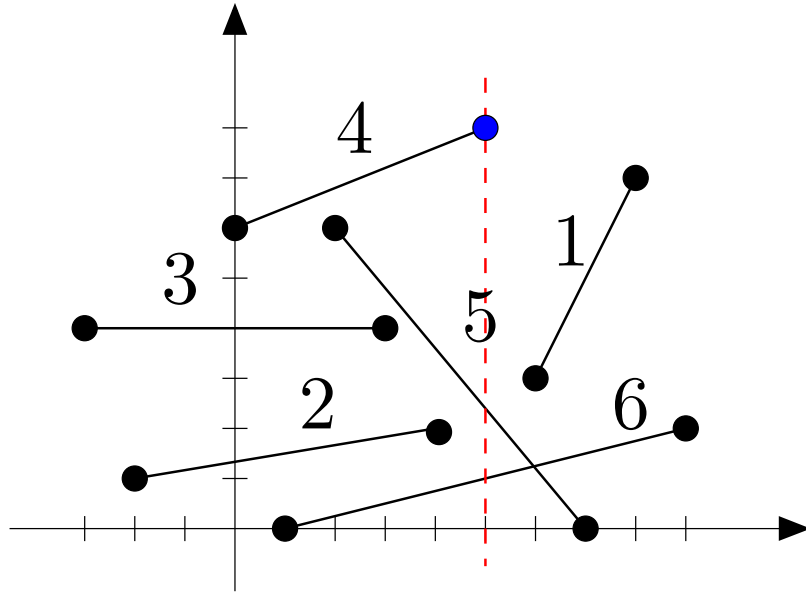


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

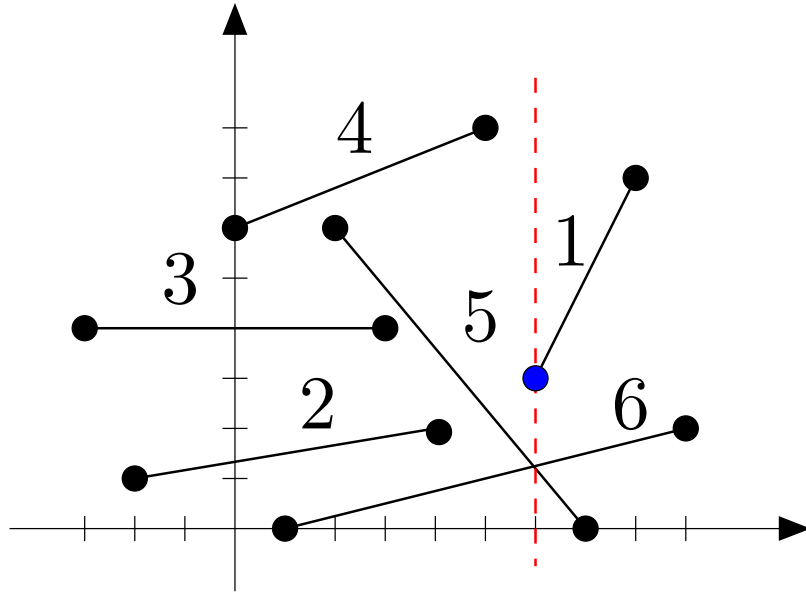


Alterações ocorrem nos **extremos dos segmentos**.

Estes são os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

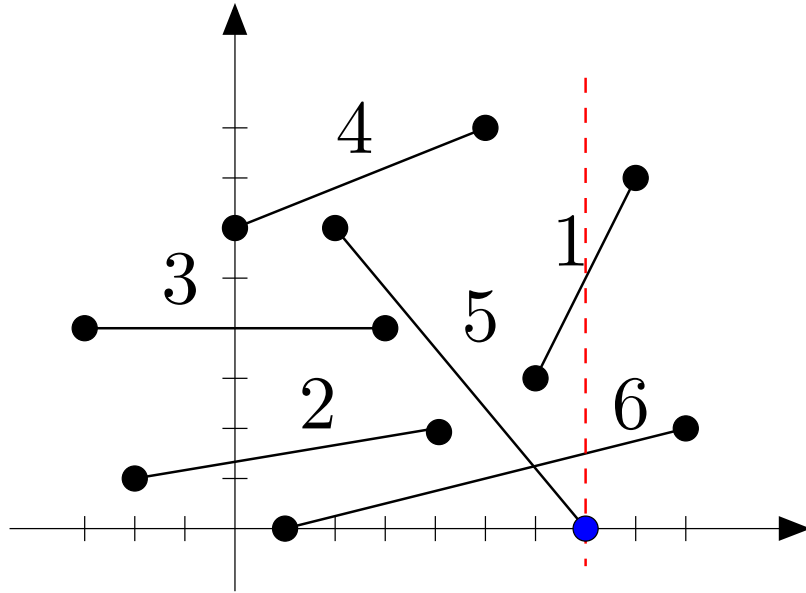


Alterações ocorrem nos **extremos dos segmentos**.

Estes são os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

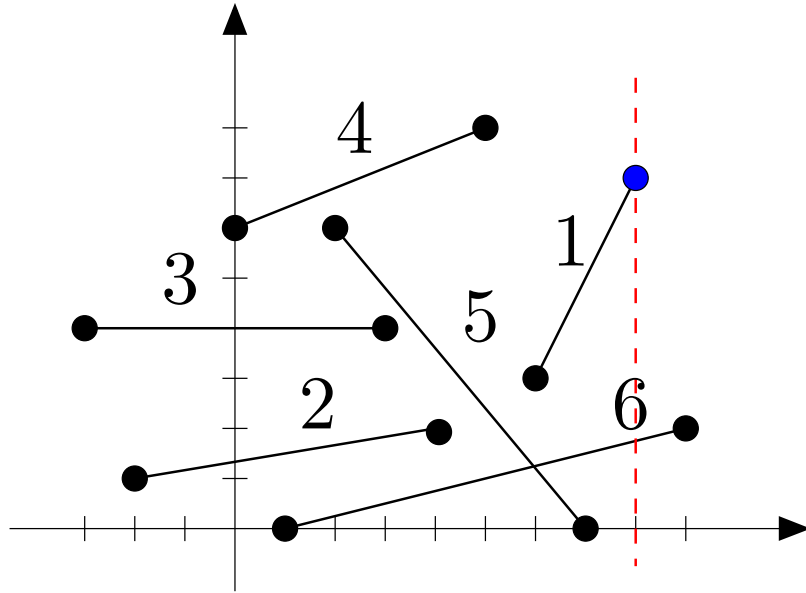


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

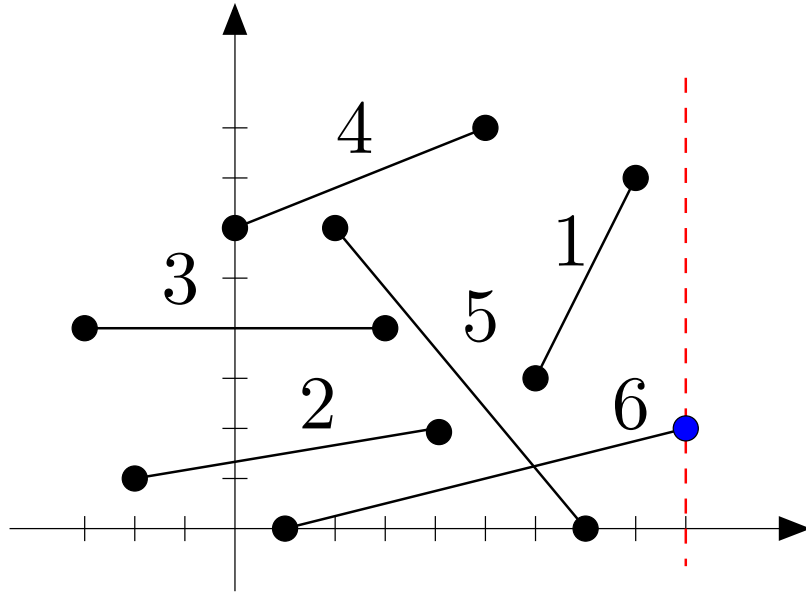


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

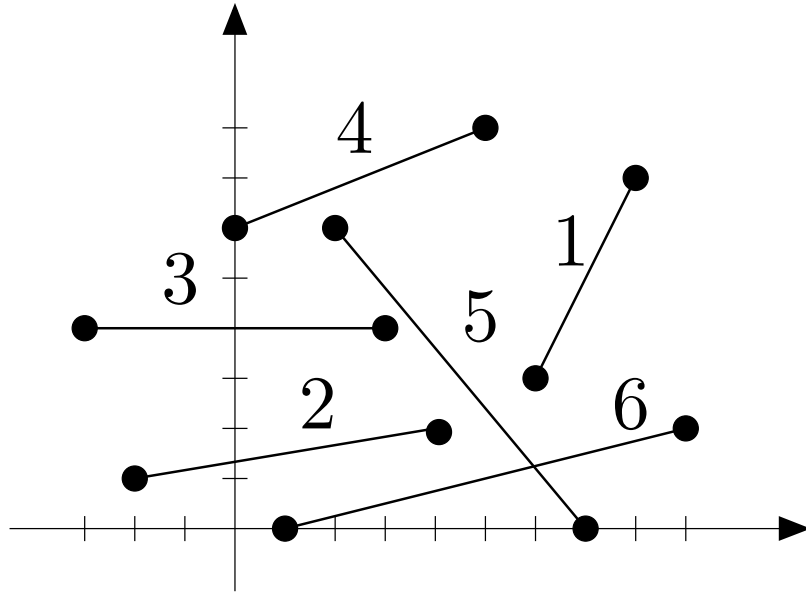


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

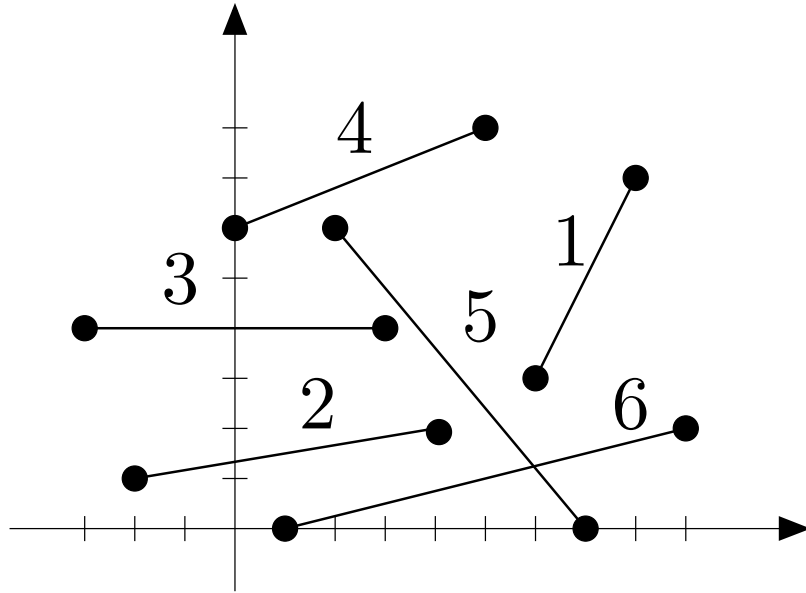
Descrição combinatória da linha



Como guardar
um destes conjuntos?

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha



Como guardar
um destes conjuntos?
Que operações ele sofre?

$x < -3$	\emptyset
$-3 \leq x < -2$	$\{3\}$
$-2 \leq x < 0$	$\{2, 3\}$
$0 \leq x < 1$	$\{2, 3, 4\}$
$1 \leq x \leq 2$	$\{2, 3, 4, 6\}$
$2 < x < 3$	$\{2, 3, 4, 5, 6\}$
$3 \leq x < 4$	$\{2, 4, 5, 6\}$
$4 \leq x \leq 5$	$\{4, 5, 6\}$
$5 < x < 6$	$\{5, 6\}$
$6 \leq x \leq 7$	$\{1, 5, 6\}$
$7 < x \leq 8$	$\{1, 6\}$
$8 < x \leq 9$	$\{6\}$
$9 < x$	\emptyset

Descrição combinatória da linha

O conjunto dos segmentos na linha sofre **inserções** e **remoções**.

Descrição combinatória da linha

O conjunto dos segmentos na linha sofre **inserções** e **remoções**.

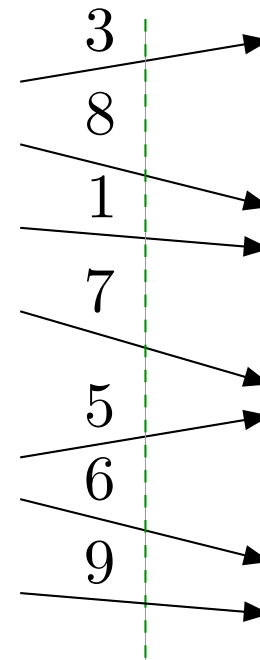
Como a linha vai nos ajudar a detectar interseção?

Descrição combinatória da linha

O conjunto dos segmentos na linha sofre **inserções** e **remoções**.

Como a linha vai nos ajudar a detectar interseção?

Ideia: testar interseção apenas entre segmentos “vizinhos na linha”.



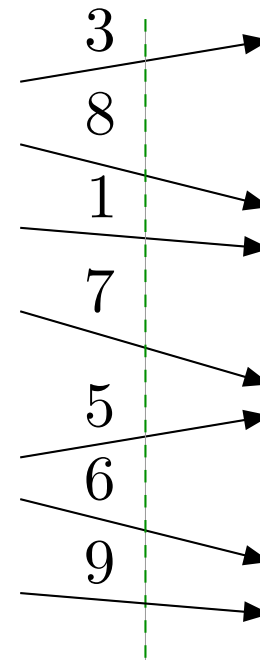
Descrição combinatória da linha

O conjunto dos segmentos na linha sofre **inserções** e **remoções**.

Como a linha vai nos ajudar a detectar interseção?

Ideia: testar interseção apenas entre segmentos “vizinhos na linha”.

Para isso, mantemos os segmentos na linha **ordenados**.



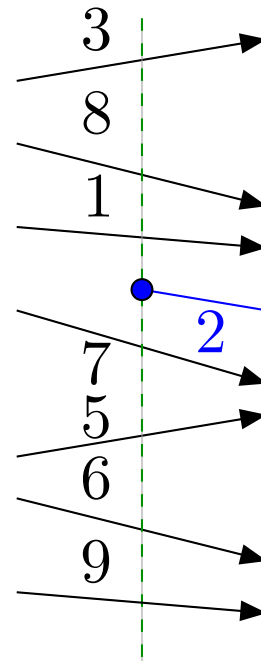
Descrição combinatória da linha

Os segmentos ficam na ordem em que intersectam a **linha**.

Descrição combinatória da linha

Os segmentos ficam na ordem em que intersectam a **linha**.

3 < 8 < 1 < 2 < 7 < 5 < 6 < 9

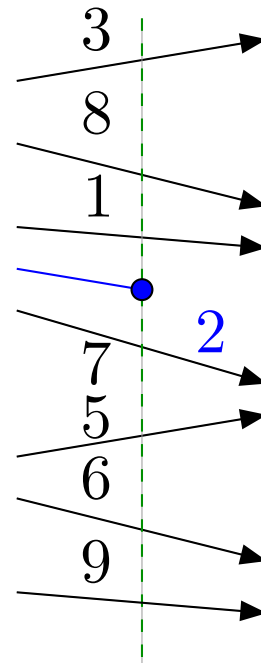


Ao **inserir** um segmento, testamos a interseção dele com seu **predecessor** e com seu **sucessor** na ordem.

Descrição combinatória da linha

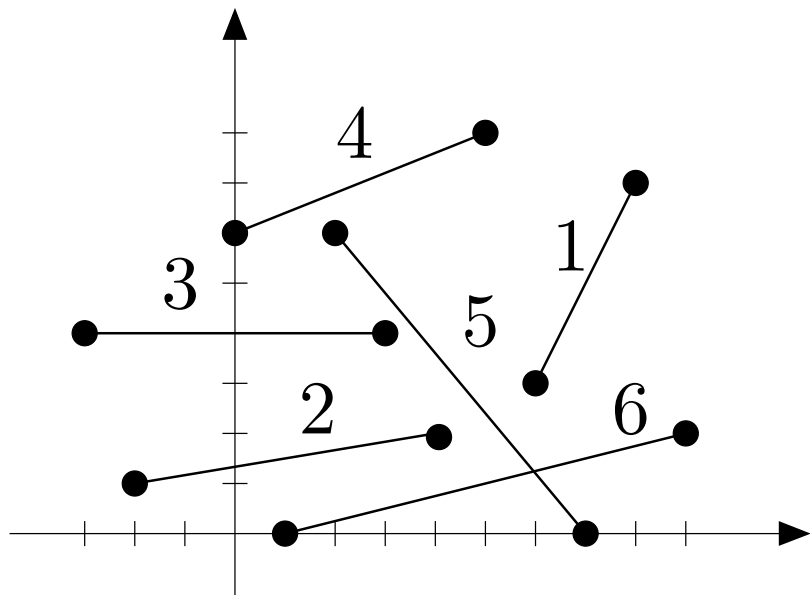
Os segmentos ficam na ordem em que intersectam a **linha**.

3 \prec 8 \prec 1 \prec 2 \prec 7 \prec 5 \prec 6 \prec 9



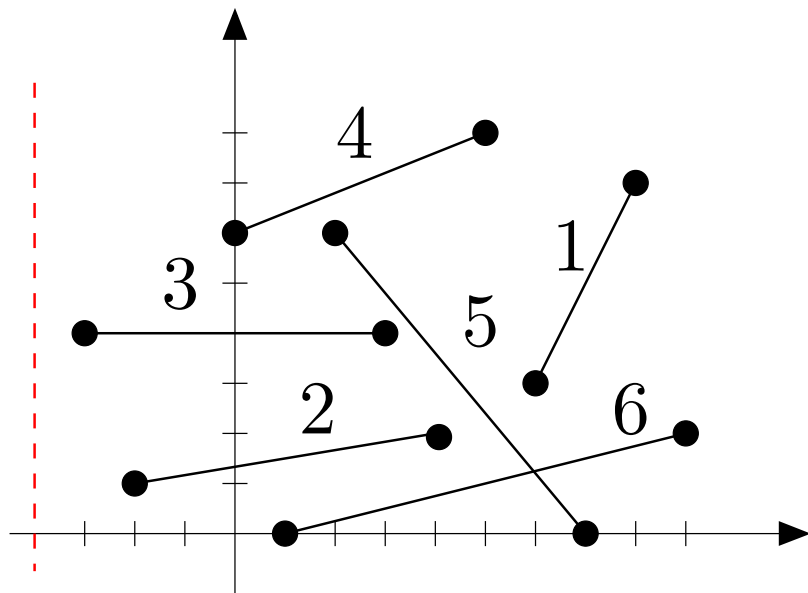
Ao **removermos** um segmento, testamos a interseção de seu **predecessor** e com seu **sucessor** na ordem.

Algoritmo de Shamos e Hoey



$-\infty$	
-3	3
-2	3 \prec 2
0	4 \prec 3 \prec 2
1	4 \prec 3 \prec 2 \prec 6
2	4 \prec 5 \prec 3 \prec 2 \prec 6
3	4 \prec 5 \prec 2 \prec 6
4	4 \prec 5 \prec 6
5	...
6	
7	
8	

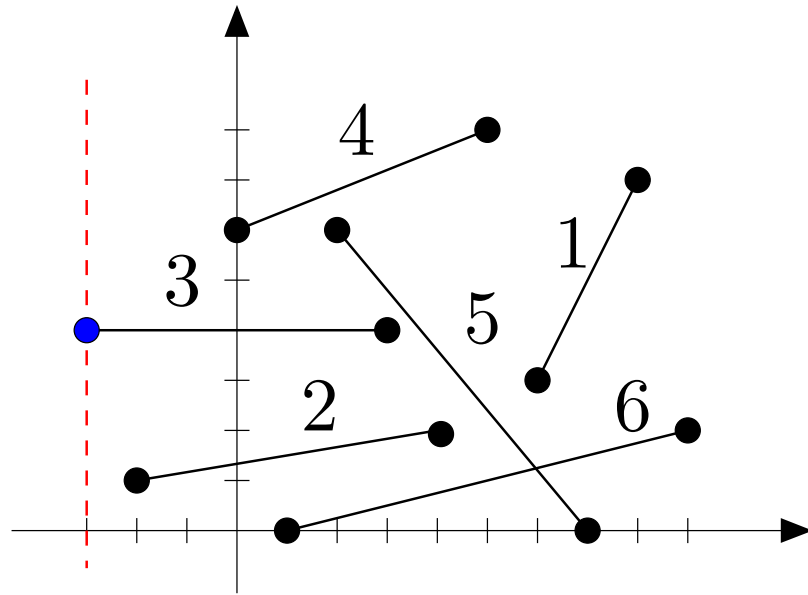
Algoritmo de Shamos e Hoey



Alterações ocorrem
nos **extremos dos segmentos**.

$-\infty$	
-3	3
-2	3 \prec 2
0	4 \prec 3 \prec 2
1	4 \prec 3 \prec 2 \prec 6
2	4 \prec 5 \prec 3 \prec 2 \prec 6
3	4 \prec 5 \prec 2 \prec 6
4	4 \prec 5 \prec 6
5	
6	
7	
8	

Algoritmo de Shamos e Hoey

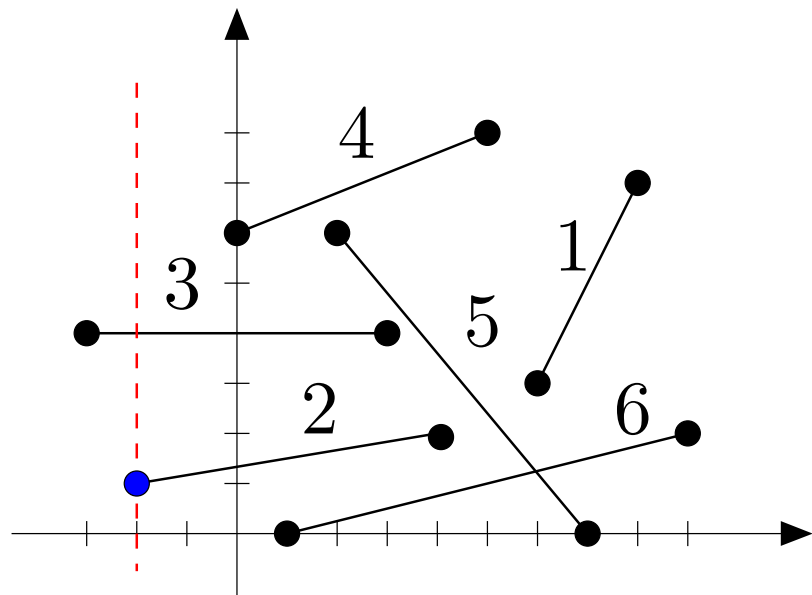


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$-\infty$	
-3	3
-2	3 < 2
0	4 < 3 < 2
1	4 < 3 < 2 < 6
2	4 < 5 < 3 < 2 < 6
3	4 < 5 < 2 < 6
4	4 < 5 < 6
5	
6	
7	
8	

Algoritmo de Shamos e Hoey

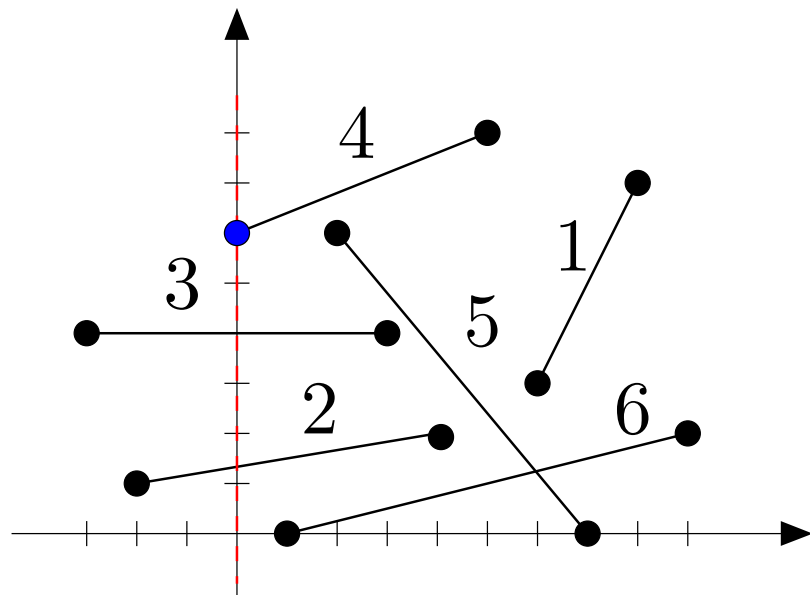


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$-\infty$	
-3	3
-2	3 < 2
0	4 < 3 < 2
1	4 < 3 < 2 < 6
2	4 < 5 < 3 < 2 < 6
3	4 < 5 < 2 < 6
4	4 < 5 < 6
5	5 < 6
6	...
7	
8	

Algoritmo de Shamos e Hoey

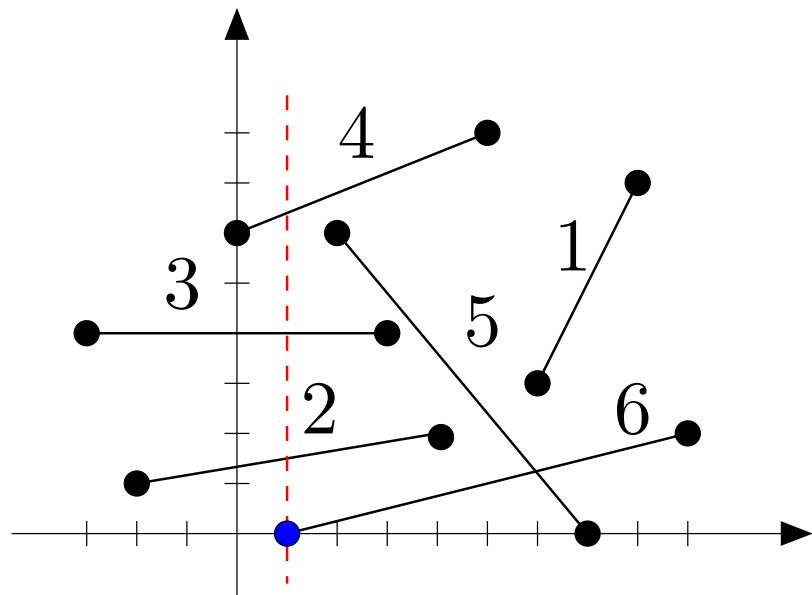


Alterações ocorrem nos **extremos dos segmentos**.

Estes são os **pontos eventos**.

$-\infty$	
-3	3
-2	3 \prec 2
0	4 \prec 3 \prec 2
1	4 \prec 3 \prec 2 \prec 6
2	4 \prec 5 \prec 3 \prec 2 \prec 6
3	4 \prec 5 \prec 2 \prec 6
4	4 \prec 5 \prec 6
5	
6	
7	
8	

Algoritmo de Shamos e Hoey

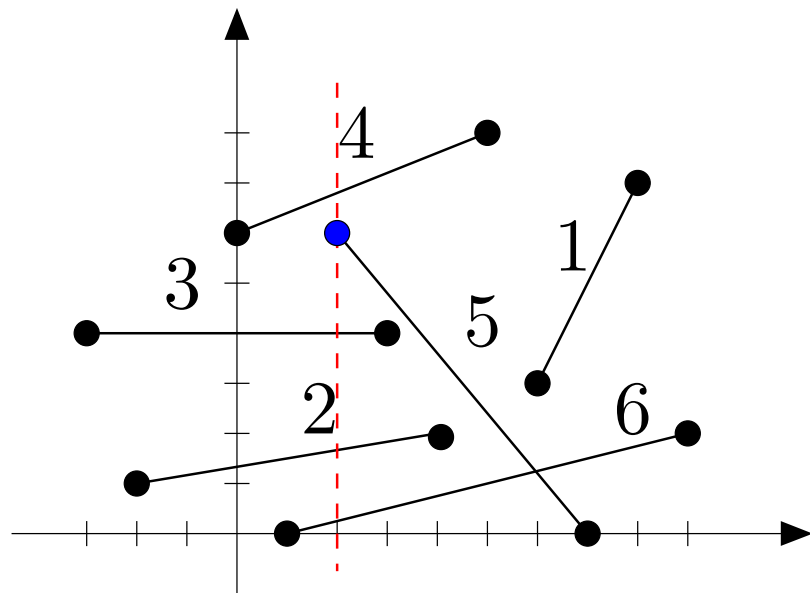


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$-\infty$	
-3	3
-2	3 \prec 2
0	4 \prec 3 \prec 2
1	4 \prec 3 \prec 2 \prec 6
2	4 \prec 5 \prec 3 \prec 2 \prec 6
3	4 \prec 5 \prec 2 \prec 6
4	4 \prec 5 \prec 6
5	
6	
7	
8	

Algoritmo de Shamos e Hoey

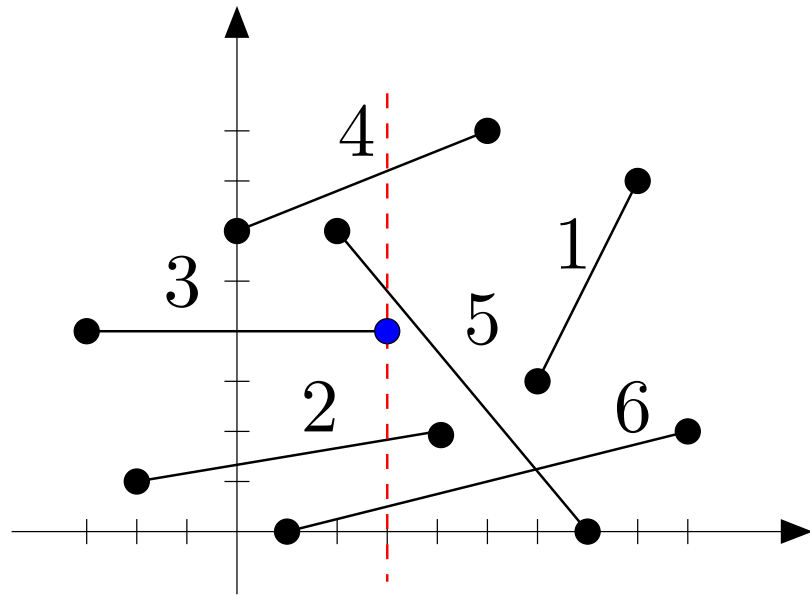


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$-\infty$	
-3	3
-2	3 \prec 2
0	4 \prec 3 \prec 2
1	4 \prec 3 \prec 2 \prec 6
2	4 \prec 5 \prec 3 \prec 2 \prec 6
3	4 \prec 5 \prec 2 \prec 6
4	4 \prec 5 \prec 6
5	
6	
7	
8	

Algoritmo de Shamos e Hoey

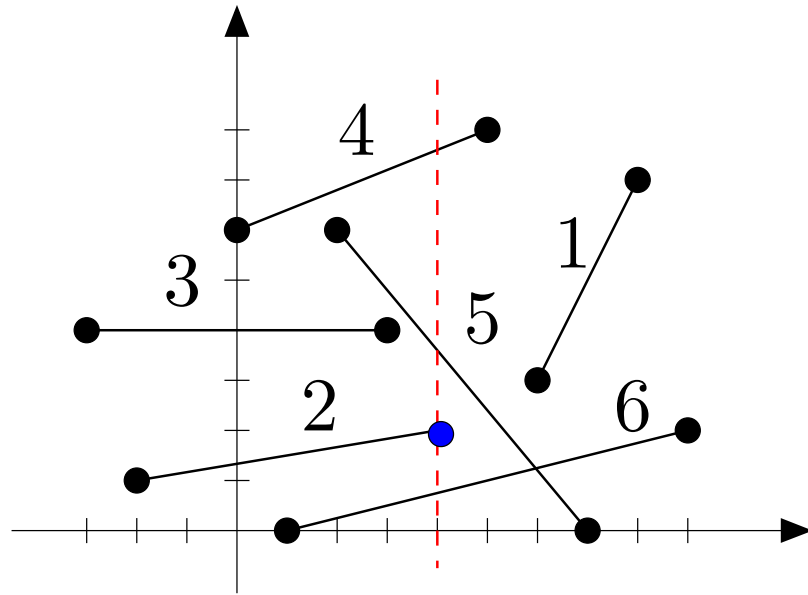


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$-\infty$	
-3	3
-2	3 \prec 2
0	4 \prec 3 \prec 2
1	4 \prec 3 \prec 2 \prec 6
2	4 \prec 5 \prec 3 \prec 2 \prec 6
3	4 \prec 5 \prec 2 \prec 6
4	4 \prec 5 \prec 6
5	
6	
7	
8	

Algoritmo de Shamos e Hoey

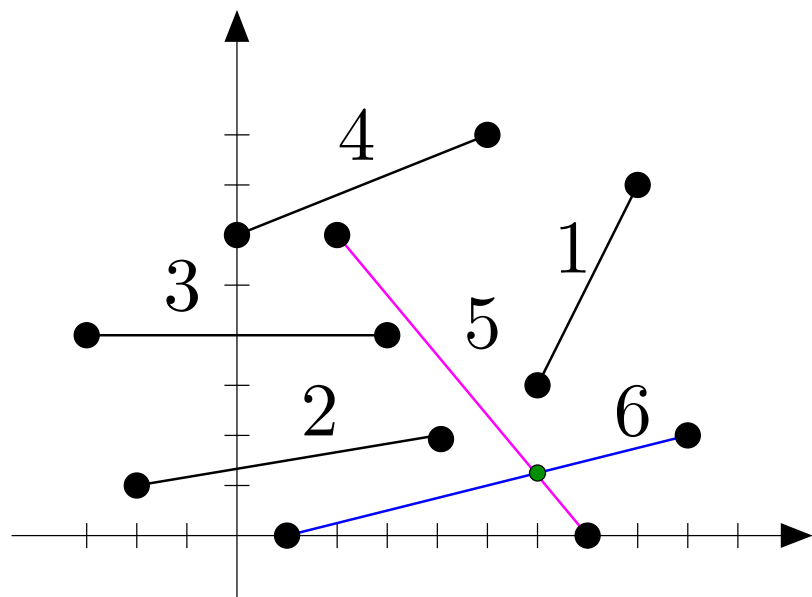


Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

$-\infty$	
-3	3
-2	3 \prec 2
0	4 \prec 3 \prec 2
1	4 \prec 3 \prec 2 \prec 6
2	4 \prec 5 \prec 3 \prec 2 \prec 6
3	4 \prec 5 \prec 2 \prec 6
4	4 \prec 5 \prec 6
5	
6	
7	
8	

Algoritmo de Shamos e Hoey



Alterações ocorrem
nos **extremos dos segmentos**.

Estes são
os **pontos eventos**.

Encontrou uma interseção!

$-\infty$	
-3	3
-2	3 \prec 2
0	4 \prec 3 \prec 2
1	4 \prec 3 \prec 2 \prec 6
2	4 \prec 5 \prec 3 \prec 2 \prec 6
3	4 \prec 5 \prec 2 \prec 6
4	4 \prec 5 \prec 6
5	
6	
7	
8	

Descrição combinatória da linha

Como guardar esse conjunto ordenado?

Descrição combinatória da linha

Como guardar esse conjunto ordenado?

Efetuiremos **inserções**, **remoções**, **predecessor** e **sucessor** neste conjunto.

Descrição combinatória da linha

Como guardar esse conjunto ordenado?

Efetuiremos **inserções**, **remoções**, **predecessor** e **sucessor** neste conjunto.

Por isso, boas escolhas de EDs são:
uma **árvore de busca binária balanceada (ABBB)**
ou uma **skip lists**.

Descrição combinatória da linha

Como guardar esse conjunto ordenado?

Efetuiremos **inserções**, **remoções**, **predecessor** e **sucessor** neste conjunto.

Por isso, boas escolhas de EDs são:
uma **árvore de busca binária balanceada (ABBB)**
ou uma **skip lists**.

Numa **ABBB**,
custo de pior caso por operação é $O(\lg m)$,
onde m é o número de elementos armazenados.

Numa **skip list**,
custo esperado por operação é $O(\lg m)$.

Algoritmo de Shamos e Hoey

Entrada: coleção $e[1..n], d[1..n]$ de segmentos.

Algoritmo de Shamos e Hoey

Entrada: coleção $e[1..n], d[1..n]$ de segmentos.

Saída: VERDADE se há dois segmentos na coleção que se intersectam, e FALSO caso contrário.

Algoritmo de Shamos e Hoey

Entrada: coleção $e[1..n], d[1..n]$ de segmentos.

Saída: VERDADE se há dois segmentos na coleção que se intersectam, e FALSO caso contrário.

Hipótese simplificadora:

Não há dois pontos extremos com a mesma X -coordenada.

Em particular, não há segmentos verticais,
nem dois segmentos com extremos coincidentes.

Montagem da fila de eventos

FILADEEVENTOS:

recebe $e[1..n]$ e $d[1..n]$ com extremos dos segmentos

Montagem da fila de eventos

FILADEEVENTOS:

recebe $e[1..n]$ e $d[1..n]$ com extremos dos segmentos

troca $e[i]$ por $d[i]$ para todo i tal que $e_X[i] > d_X[i]$

($e[i]$: extremo esquerdo do segmento i e $d[i]$ o direito)

Montagem da fila de eventos

FILADEEVENTOS:

recebe $e[1..n]$ e $d[1..n]$ com extremos dos segmentos

troca $e[i]$ por $d[i]$ para todo i tal que $e_X[i] > d_X[i]$

($e[i]$: extremo esquerdo do segmento i e $d[i]$ o direito)

devolve

$E[1..2n]$: pontos de $e[1..n]$ e $d[1..n]$
ordenados pelas suas X -coordenadas

Montagem da fila de eventos

FILADEEVENTOS:

recebe $e[1..n]$ e $d[1..n]$ com extremos dos segmentos

troca $e[i]$ por $d[i]$ para todo i tal que $e_X[i] > d_X[i]$

($e[i]$: extremo esquerdo do segmento i e $d[i]$ o direito)

devolve

$E[1..2n]$: pontos de $e[1..n]$ e $d[1..n]$
ordenados pelas suas X -coordenadas

$segm[1..2n]$:

$segm[p]$: índice do segmento do qual $E[p]$ é extremo

Montagem da fila de eventos

FILADEEVENTOS:

recebe $e[1..n]$ e $d[1..n]$ com extremos dos segmentos

troca $e[i]$ por $d[i]$ para todo i tal que $e_X[i] > d_X[i]$

($e[i]$: extremo esquerdo do segmento i e $d[i]$ o direito)

devolve

$E[1..2n]$: pontos de $e[1..n]$ e $d[1..n]$
ordenados pelas suas X -coordenadas

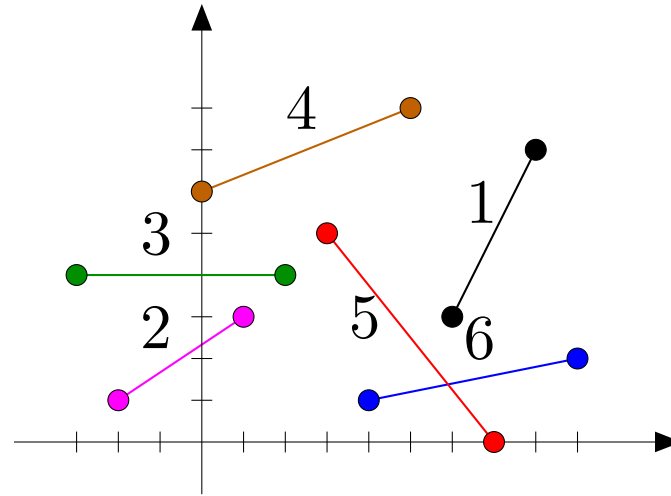
$segm[1..2n]$:

$segm[p]$: índice do segmento do qual $E[p]$ é extremo

$esq[1..2n]$:

$esq[p]$: VERDADE se $E[p]$ é extremo esquerdo de $segm[p]$
FALSO caso contrário.

Fila de eventos



E_X	-3	-2	0	1	2	3	4	5	6	7	8	9
E_Y	4	1	6	3	4	5	1	8	3	0	7	2
	1	2	3	4	5	6	7	8	9	10	11	12

$segm$	3	2	4	2	3	5	6	4	1	5	1	6
esq	V	V	V	F	F	V	V	F	V	F	F	F
	1	2	3	4	5	6	7	8	9	10	11	12

Processamento de ponto evento

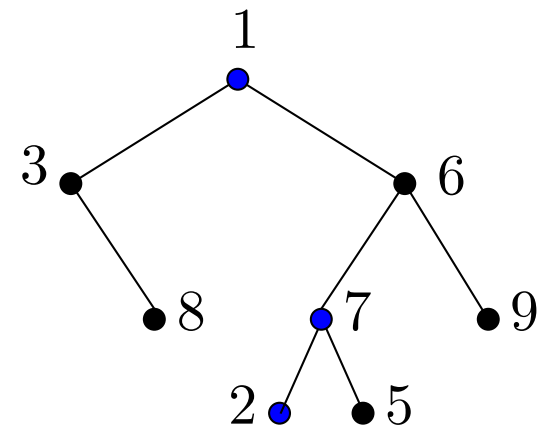
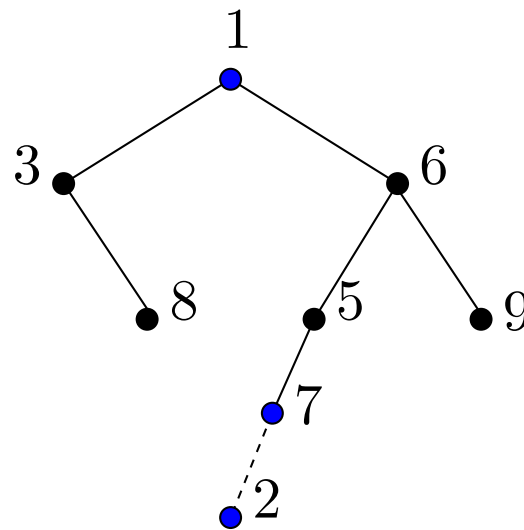
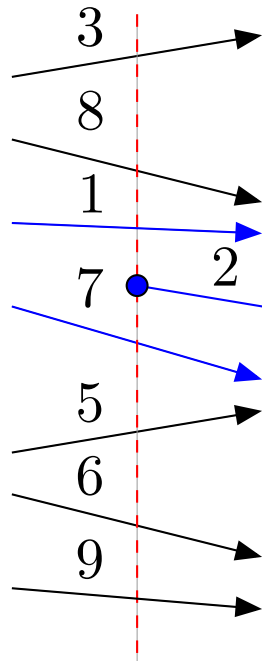
Dois tipos:

- **começo de segmento:** inclui o novo segmento na ABBB e verifica interseção com **seus dois novos “vizinhos”**.

Processamento de ponto evento

Dois tipos:

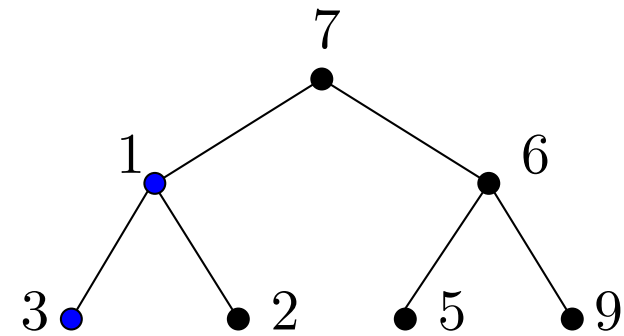
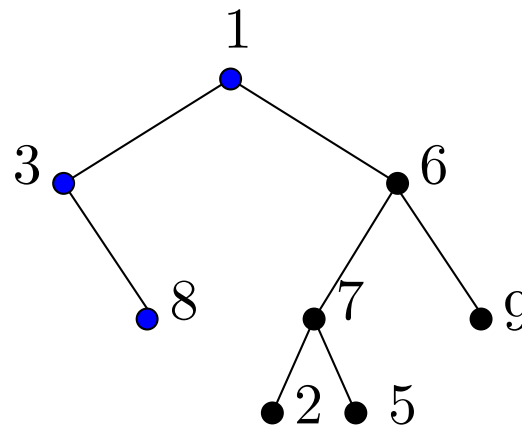
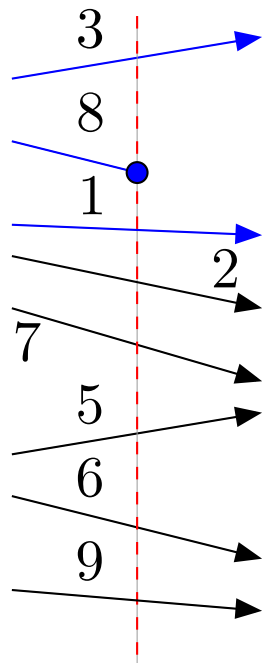
- **começo de segmento:** inclui o novo segmento na ABBB e verifica interseção com seus dois novos “vizinhos”.



Processamento de ponto evento

Dois tipos:

- **começo de segmento:** inclui o novo segmento na ABBB e verifica interseção com **seus dois novos “vizinhos”**.
- **fim de segmento:** remove o segmento da ABBB e verifica interseção entre **seus dois ex-vizinhos**.



Processamento de ponto evento

Dois tipos:

- **começo de segmento:** inclui o novo segmento na ABBB e verifica interseção com **seus dois novos “vizinhos”**.
- **fim de segmento:** remove o segmento da ABBB e verifica interseção entre **seus dois ex-vizinhos**.

Invariante: verificamos interseção entre quaisquer dois segmentos vizinhos na ABBB.

Processamento de ponto evento

Dois tipos:

- **começo de segmento:** inclui o novo segmento na ABBB e verifica interseção com **seus dois novos “vizinhos”**.
- **fim de segmento:** remove o segmento da ABBB e verifica interseção entre **seus dois ex-vizinhos**.

Invariante: verificamos interseção entre quaisquer dois segmentos vizinhos na ABBB.

Correção: se há dois segmentos que se intersectam, em algum momento, os dois serão vizinhos na ABBB.

Algoritmo de Shamos e Hoey

INTERSEÇÃO-SH(e, d, n)

```
1 ( $E, segm, esq$ )  $\leftarrow$  FILADEEVENTOS( $e, d, n$ )
2 CRIE( $T$ )
3 para  $p \leftarrow 1$  até  $2n$  faça
4    $i \leftarrow segm[p]$ 
5    $pred \leftarrow$  Predecessor( $T, E_X[p], E_Y[p]$ )
6    $suc \leftarrow$  Sucessor( $T, E_X[p], E_Y[p]$ )
7   se  $esq[p]$ 
8     então Insere( $T, i$ )
9     se ( $pred \neq \text{NIL}$  e INTER( $e, d, i, pred$ ))
10      ou ( $suc \neq \text{NIL}$  e INTER( $e, d, i, suc$ ))
11      então devolva VERDADE
12   senão Remove( $T, i$ )
13   se  $pred \neq \text{NIL}$  e  $suc \neq \text{NIL}$  e INTER( $e, d, pred, suc$ )
14     então devolva VERDADE
15 devolva FALSO
```

Consumo de tempo

O algoritmo executa $2n$ iterações.

Cada iteração faz uma chamada a **Predecessor**, uma a **Sucessor**, e uma a **Inserir** ou a **Remove**.

Na ABBB, em qualquer momento, há $O(n)$ segmentos.

Assim, cada uma destas operações consome tempo $O(\lg n)$.

As demais operações efetuadas em uma iteração consomem tempo $O(1)$ (mesmo as chamadas a INTER).

Logo o consumo de tempo por iteração é $O(\lg n)$, e **o algoritmo de Shamos e Hoey consome tempo $O(n \lg n)$.**

Como evitar a hipótese simplificadora?

Pontos extremos com mesma x -coordenada:

Se existir um segmento vertical,
deixe o extremo inferior no vetor e e o superior no vetor d .

Como evitar a hipótese simplificadora?

Pontos extremos com mesma x -coordenada:

Se existir um segmento vertical,
deixe o extremo inferior no vetor e e o superior no vetor d .

Se houver extremos repetidos, há interseção.

Como evitar a hipótese simplificadora?

Pontos extremos com mesma x -coordenada:

Se existir um segmento vertical,
deixe o extremo inferior no vetor e e o superior no vetor d .

Se houver extremos repetidos, há interseção.

Extremo **esquerdo** de um segmento:

- extremo cuja x -coordenada é menor.
- caso o segmento seja **vertical**,
chame de esquerdo o extremo
com y -coordenada menor.

Como evitar a hipótese simplificadora?

Pontos extremos com mesma x -coordenada:

Se existir um segmento vertical,
deixe o extremo inferior no vetor e e o superior no vetor d .

Se houver extremos repetidos, há interseção.

Extremo **esquerdo** de um segmento:

- extremo cuja x -coordenada é menor.
- caso o segmento seja **vertical**,
chame de esquerdo o extremo
com y -coordenada menor.

O outro extremo é o **direito**.

Como evitar a hipótese simplificadora?

Pontos extremos com mesma x -coordenada:

Se existir um segmento vertical,
deixe o extremo inferior no vetor e e o superior no vetor d .

Se houver extremos repetidos, há interseção.

Extremo **esquerdo** de um segmento:

- extremo cuja x -coordenada é menor.
- caso o segmento seja **vertical**,
chame de esquerdo o extremo
com y -coordenada menor.

O outro extremo é o **direito**.

Extremos-Ordenados (n, S) : ordena os extremos dos n
segmentos em S e já dá a resposta se houver repetição.

Detecção de interseção

Detecta-Interseção(n, S)

```
1   $E \leftarrow$  Extremos-Ordenados( $n, S$ )
2   $T \leftarrow \emptyset$      $\triangleright$  ABBB ou skip list
3  para cada  $p \in E$  faça
4     $s \leftarrow$  segmento( $p$ )
5     $pred \leftarrow$  Predecessor( $T, s$ )     $suc \leftarrow$  Sucessor( $T, s$ )
6    se  $p$  é extremo esquerdo de  $s$ 
7      então Insere( $T, s$ )
8          se ( $pred \neq_{\text{NIL}}$  e Intersecta( $s, pred$ ))
9              ou ( $suc \neq_{\text{NIL}}$  e Intersecta( $s, suc$ ))
10             então devolva VERDADE
11         senão Remove( $T, s$ )
12             se  $pred$  e  $suc \neq_{\text{NIL}}$  e Intersecta( $pred, suc$ )
13                 então devolva VERDADE
14 devolva FALSO
```

Inserção em ABB rubro-negra

INSIRAREC (T, x)

```
1  se  $T = \text{NIL}$ 
2    então  $q \leftarrow \text{NOVACÉLULA}(x, \text{NIL}, \text{NIL}, \text{RUBRO})$ 
3    devolva  $q$ 
4  se  $x < \text{info}(T)$        $\triangleright$  Vamos alterar aqui!
5    então  $\text{esq}(T) \leftarrow \text{INSIRAREC}(\text{esq}(T), x)$ 
6    senão  $\text{dir}(T) \leftarrow \text{INSIRAREC}(\text{dir}(T), x)$ 
7  se  $\text{RUBRO}(\text{dir}(T))$  e  $\text{NEGRO}(\text{esq}(T))$ 
8    então  $T \leftarrow \text{GIREESQ}(T)$ 
9  se  $\text{RUBRO}(\text{esq}(T))$  e  $\text{RUBRO}(\text{esq}(\text{esq}(T)))$ 
10   então  $T \leftarrow \text{GIREDIR}(T)$ 
11  se  $\text{RUBRO}(\text{esq}(T))$  e  $\text{RUBRO}(\text{dir}(T))$ 
12   então  $\text{TROQUECORES}(T)$ 
13  devolva  $T$ 
```

Inserção em ABB rubro-negra

INSIRAREC (T, e, d, i)

- 1 **se** $T = \text{NIL}$
- 2 **então** $q \leftarrow \text{NOVACÉLULA}(i, \text{NIL}, \text{NIL}, \text{RUBRO})$
- 3 **devolva** q
- 4 **se** $\text{ESQUERDA}(e[\text{segmento}(T)], d[\text{segmento}(T)]), e[i]$
- 5 **então** $\text{esq}(T) \leftarrow \text{INSIRAREC}(\text{esq}(T), i)$
- 6 **senão** $\text{dir}(T) \leftarrow \text{INSIRAREC}(\text{dir}(T), i)$
- 7 **se** $\text{RUBRO}(\text{dir}(T))$ e $\text{NEGRO}(\text{esq}(T))$
- 8 **então** $T \leftarrow \text{GIREESQ}(T)$
- 9 **se** $\text{RUBRO}(\text{esq}(T))$ e $\text{RUBRO}(\text{esq}(\text{esq}(T)))$
- 10 **então** $T \leftarrow \text{GIREDIR}(T)$
- 11 **se** $\text{RUBRO}(\text{esq}(T))$ e $\text{RUBRO}(\text{dir}(T))$
- 12 **então** $\text{TROQUECORES}(T)$
- 13 **devolva** T

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Você consegue projetar um algoritmo que consuma tempo $O(n \lg n)$ para este problema?

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Você consegue projetar um algoritmo que consuma tempo $O(n \lg n)$ para este problema?

No máximo, quantos pares teremos que imprimir?

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Você consegue projetar um algoritmo que consuma tempo $O(n \lg n)$ para este problema?

No máximo, quantos pares teremos que imprimir?

Algoritmos sensíveis à saída (*output sensitive*).

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Como adaptar o algoritmo de Shamos e Hoey?

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Como adaptar o algoritmo de Shamos e Hoey?

Novo tipo de ponto evento: as interseções.

Como tratá-las?

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Como adaptar o algoritmo de Shamos e Hoey?

Novo tipo de ponto evento: as interseções.

Como tratá-las?

Ao detectar cada uma, além de imprimi-la, a colocamos na fila de eventos (que é agora dinâmica).

Todas as interseções de segmentos

Problema: Dada uma coleção de n segmentos no plano, encontrar todos os pares de segmentos da coleção que se intersectam.

Como adaptar o algoritmo de Shamos e Hoey?

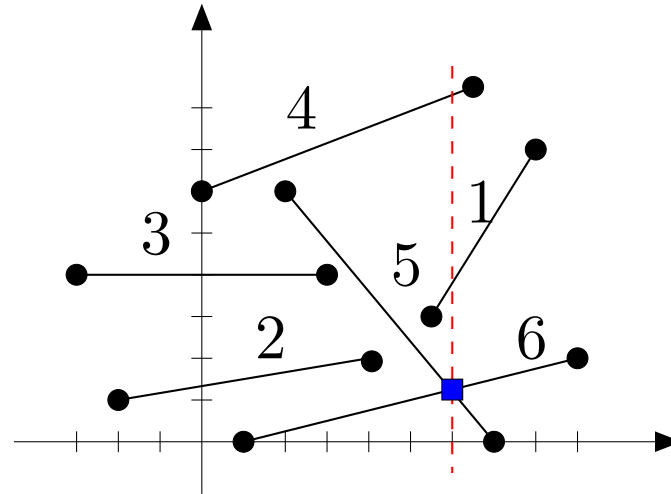
Novo tipo de ponto evento: as interseções.

Como tratá-las?

Ao detectar cada uma, além de imprimi-la, a colocamos na fila de eventos (que é agora dinâmica).

Ao processar um ponto evento que é uma interseção, deve-se inverter a ordem dos segmentos que se intersectam neste ponto.

Ponto evento: interseção



Antes do ponto evento: $4 \prec 1 \prec 5 \prec 6$

Depois do ponto evento: $4 \prec 1 \prec 6 \prec 5$