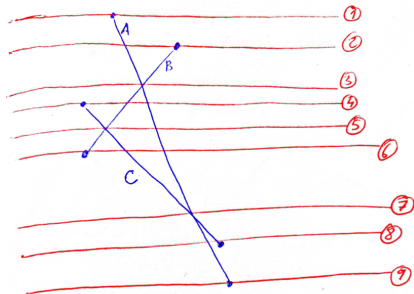


Idea:

- přejíždíme přímkou seshora dolů (nebo rotujeme polopřímku, ...),
- udržujeme průnik přímky s objekty,
- stačí měnit, když se něco zajímavého stane.



1:	A	4:	CBA	7:	AC
2:	AB	5:	BCA	8:	A
3:	BA	6:	CA	9:	∅

- U : halda událostí (dle y -souřadnice)
- P : (intervalový) strom průniků s objekty (dle x -souřadnice)

vlož předem známé události do U ;

```
u_prev = NULL;
```

```
while (!U.empty ())
```

```
{
```

```
    u = U.extract_min ();
```

```
    uprav výsledek dle stavu mezi u a u_prev;
```

```
    přidej/odeber/změň pořadí v  $P$  dle u;
```

```
    přidej/odeber nově zjištěné události do  $U$ ;
```

```
    u_prev = u;
```

```
}
```

Použití:

- obsah sjednocení či průniků
- obvod sjednocení
- body patřící do sjednocení
- počet průsečíků

Typická časová složitost $O(m \log m)$, kde m je počet událostí.

Zadání

Máme n úseček v rovině, vypište všechny jejich průsečíky.

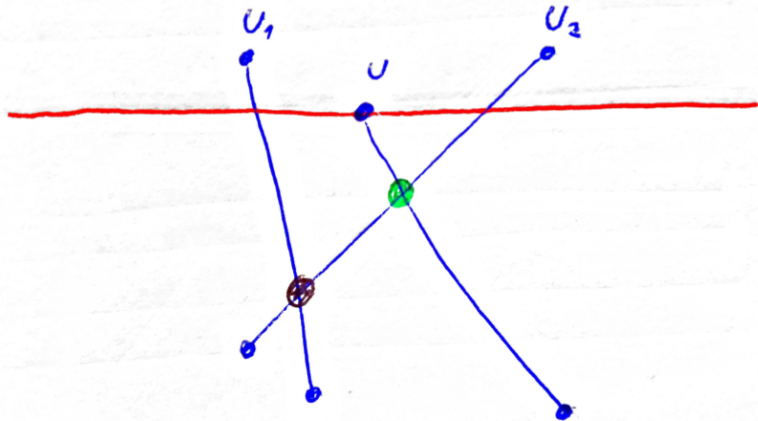
P = úsečky protínající scanline, dle pořadí průniků

Události:

- Koncové body úseček (předem známé).
- Průsečíky úseček sousedících v P (přidávané průběžně).

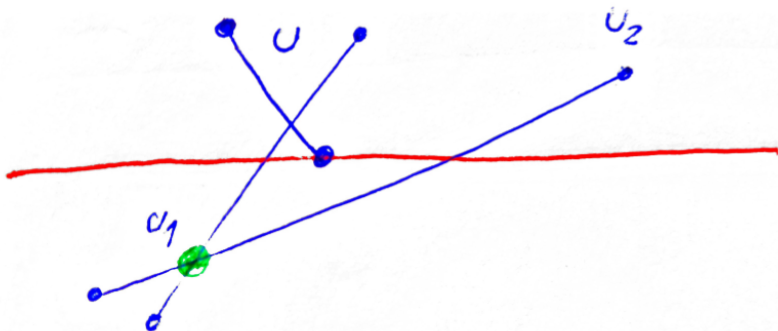
Událost: začátek úsečky u

- vlož u mezi u_1 a u_2 : předchozí a následující úsečka na scanline
- smaž událost: průsečík u_1 a u_2 , existuje-li
- přidej události: průsečík u s u_1 či u_2 , existuje-li.



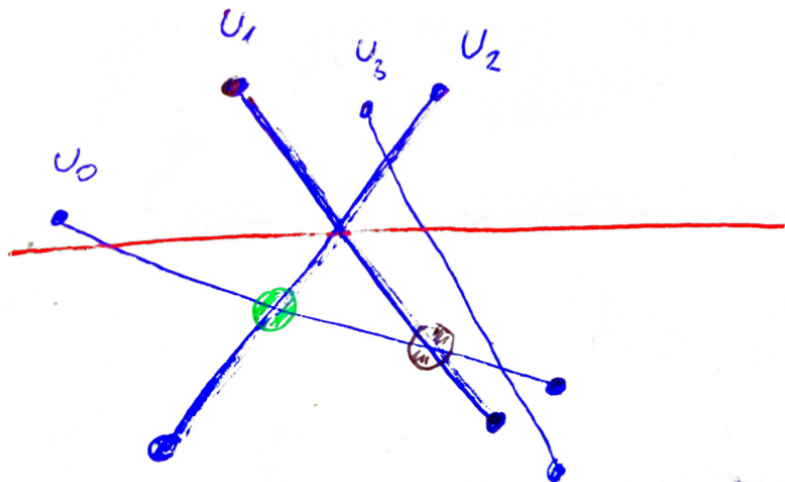
Událost: konec úsečky u

- smaž u (mezi u_1 a u_2 : předchozí a následující úsečka na scanline)
- přidej událost: průsečík u_1 a u_2 , existuje-li a je-li pod scanline



Událost: průsečík u_1 a u_2

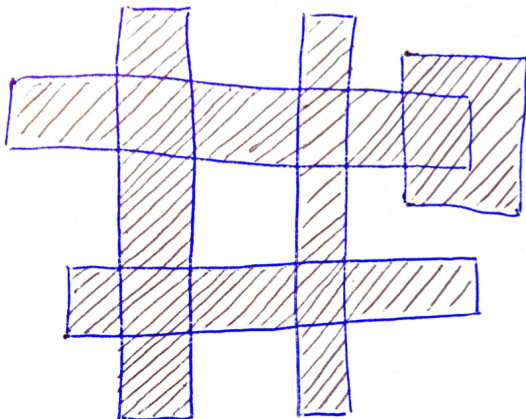
- prohod' pořadí u_1 a u_2 (předchodzí u_0 , následující u_3)
- smaž události: průsečík u_0 a u_1 , u_2 a u_3 .
- přidej události: průsečík u_0 a u_2 , průsečík u_1 a u_3 .

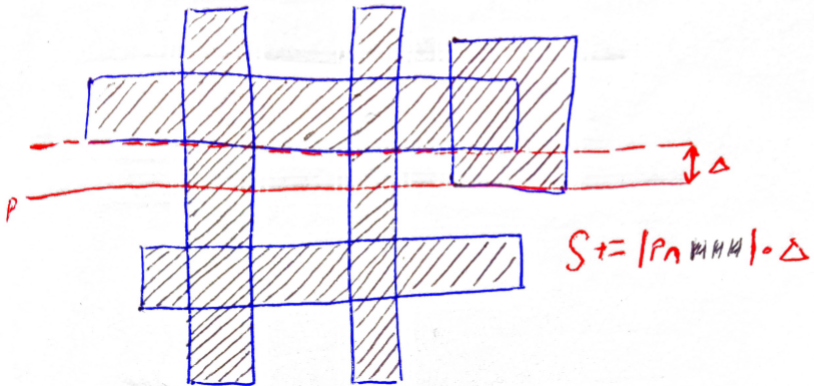


- Alternativa: události pro průsečíky nemazat, ale nepřidávat duplikáty.
- Složitost (n úseček, p průsečíků): $O((n + p) \log n)$.

Zadání

Máme n osově orientovaných obdélníků v rovině, určete obsah jejich sjednocení.





$$S_T = |P_n| \cdot \Delta$$

Události:

- Horní a dolní hrany obdélníků (předem známé).

Průniky v intervalovém stromu:

- Počet pokrytí celého intervalu (celý).
- Délka pokrytá kratšími obdélníky (části).

`pokryto = celý > 0 ? délka intervalu : části`

Obdélník od x_1 do x_2 :

- horní hrana: pokryj (kořen, x_1 , x_2 , $+1$)
- dolní hrana: pokryj (kořen, x_1 , x_2 , -1)

```
pokryj (v, x1, x2, by) {  
    if (disjunktní (v.interval, (x1,x2))) return;  
    if (podmnožina (v.interval, (x1,x2)))  
        {v.celý += by; return;}  
    pokryj (v.levý, x1, x2, by);  
    pokryj (v.pravý, x1, x2, by);  
    v.části = v.levý.pokryto + v.pravý.pokryto;  
}
```