

Intervall-Sweep

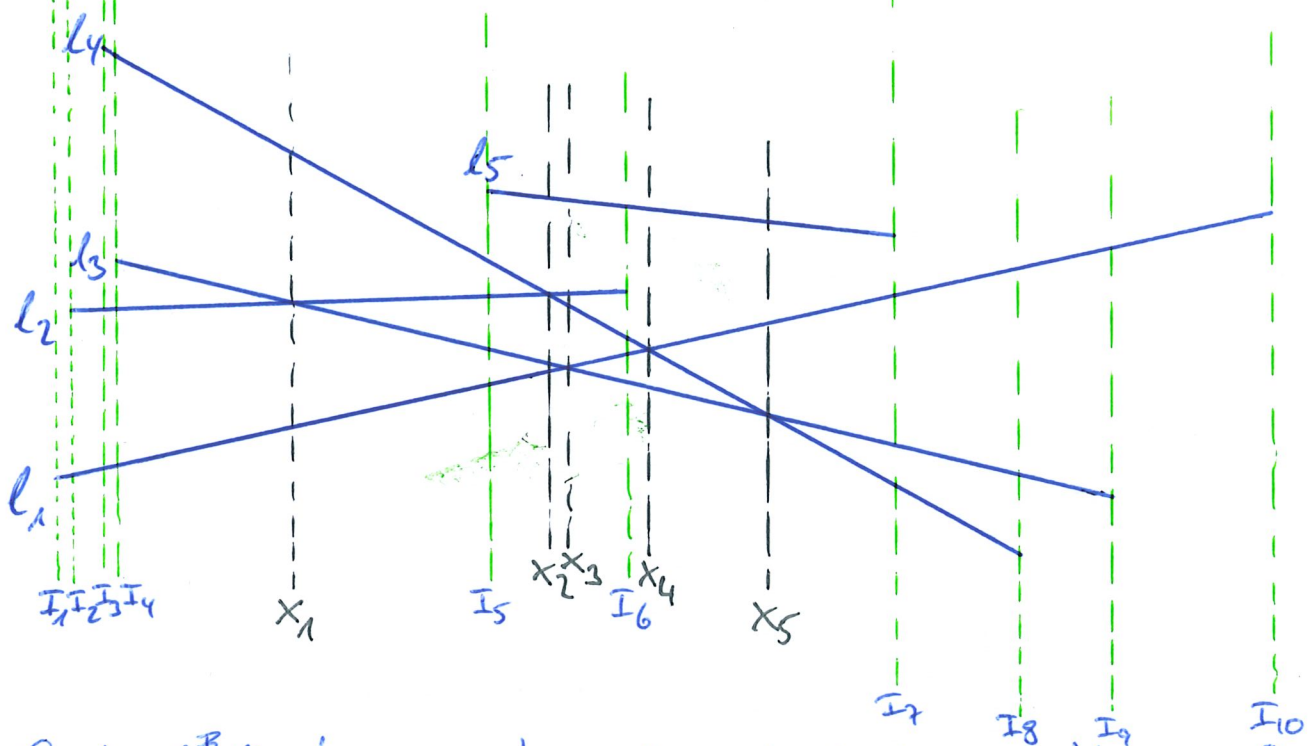
Idee: Scanne einmal von links nach rechts.

- Zähle dabei wieviele Intervalle sich aktuell überlappen.
- Wächst die Zahl auf 2, beginnt ein ~~z~~ auszugebendes Intervall. Geschieht nur, wenn ein Intervall beginnt.
- Sinkt die Zahl auf 1, endet ein auszugebendes Intervall. Geschieht nur, wenn ein Intervall endet.

In Pseudocode:

1. Sortiere Start- und Endpunkte; $counter = 0$; $start = 0$
2. Sei P diese sortierte Reihenfolge;
3. for ($p \in P$) do
4. if (p ist Startpunkt)
5. $counter = counter + 1$;
6. if ($counter = 2$) then
7. $start = p$;
8. else
9. $counter = counter - 1$;
10. if ($counter = 1$) then
11. report $[start, p]$

2D-Sweepline



grüne Gestrichelte Linien entsprechen Start-/Endpunkten von Segmenten.
 Start-Event-Queue: $[I_1, \dots, I_{10}]$

I_1 : (l_1)

I_2 : (l_1, l_2)

I_3 : (l_2, l_3)

I_4 : (l_2, l_3, l_4)

l_2 und l_3 schneiden sich!

Füge x_1 zur Queue hinzu:

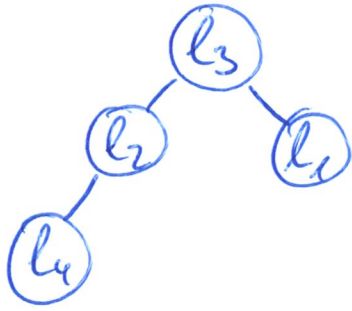
$\rightarrow [I_4, x_1, I_5, \dots, I_{10}]$

l_3 und l_4 schneiden sich!

Füge x_5 zur Queue hinzu:

$\rightarrow [x_1, I_5, I_6, x_5, I_7, I_8, \dots, I_{10}]$

X_1 : l_2 und l_3 tauschen



Dabei:

- l_1 und l_3 schneiden sich.

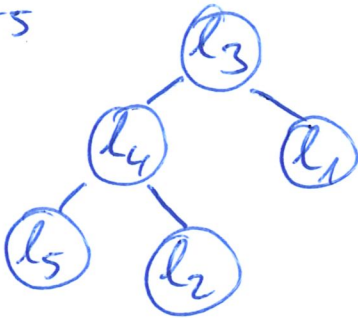
→ Füge x_3 hinzu

- l_2 und l_4 schneiden sich.

→ Füge x_2 hinzu

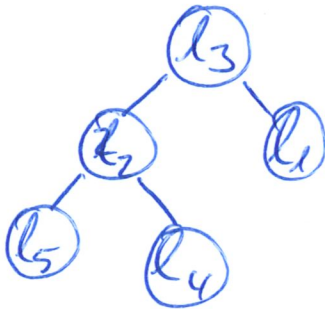
→ $[I_5, x_2, x_3, I_6, x_5, I_7, \dots, I_{10}]$

I_5



→ $[x_2, x_3, I_6, x_5, I_7, \dots, I_{10}]$

X_2 : l_2 und l_4 tauschen



Dabei:

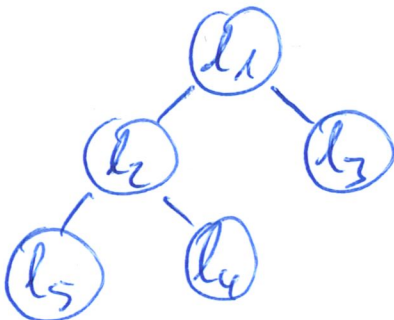
- l_4 und l_3 schneiden sich.

→ Event-Point ex. bereits

- l_2 und l_5 schneiden sich nicht.

→ $[x_3, I_6, x_5, I_7, \dots, I_{10}]$

X_3 : l_3 und l_1 tauschen



Dabei:

- l_1 und l_4 schneiden sich.

→ Füge x_4 hinzu

→ Aktuelle Queue:

$[I_6, x_4, x_5, I_7, \dots, I_{10}]$

usw.