

Shamos-Hoey

Queue (sortiert nach x)

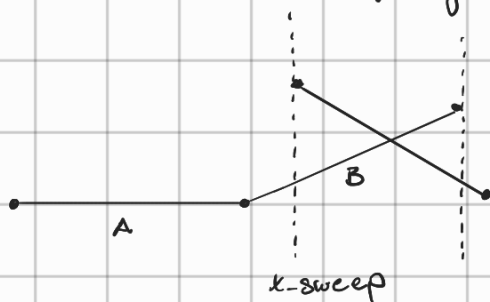
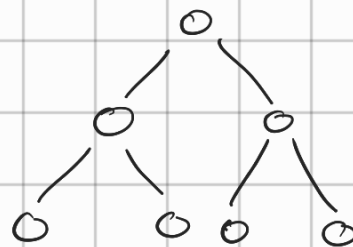
Sweep-Line-Status

$\{(x_{\max}, y), L, S_i\}$ → jeder Punkt entweder
linker oder rechter Punkt

→ Wenn x gleich, linker Endpunkt

$\{(x_{\min}, y), L, S_i\}$ vor rechten

→ Wenn x gleicher Typ
aufsteigend nach y •



Wichtige Fragen:

1. Wie sortiert man den Baum anhand der Queue?
2. Wie lautet die Bedingung für den Schnitt?

Antworten

1. Die Sortierung im Baum erfolgt durch $y = y_1 + \frac{y_2 - y_1}{x_2 - x_1} \cdot (x_{\text{sweep}} - x_1)$
• Beim z.B. binären Baum fängt man mit dem Vergleich bei Wurzel an u. klettert sich bis zur richtigen Stelle durch den Baum

2. Sei $S_1 = AB$ & $S_2 = CD$

des.: Orientierungstest

$CCW(C, D, A)$

$CCW(C, D, B)$

$CCW(A, B, C)$

$CCW(A, B, D)$

Formel: $val = (y_2 - y_1)(x_3 - x_2) - (x_2 - x_1)(y_3 - y_2)$

⇒ Schneiden, wenn $(v_1 > 0 \text{ \& } v_2 < 0 \text{ \& } v_1 < 0 \text{ \& } v_2 > 0)$

$(v_3 > 0 \text{ \& } v_4 < 0 \text{ \& } v_3 < 0 \text{ \& } v_4 > 0)$

⇒ Berühren, wenn $v_1 = 0 \text{ \& } v_2 = 0 \text{ \& } v_3 = 0 \text{ \& } v_4 = 0$

↪ Bsp: $CCW(C, D, A) = 0 \rightsquigarrow x_A$ zwischen x_C, x_D & y_A zw. y_C, y_D

⇒ Wenn $v_{1,2,3,4} = 0$ überprüfen ↪ Liegt Punkt auf Segment der unendl. Gerade?
auf Überlappung

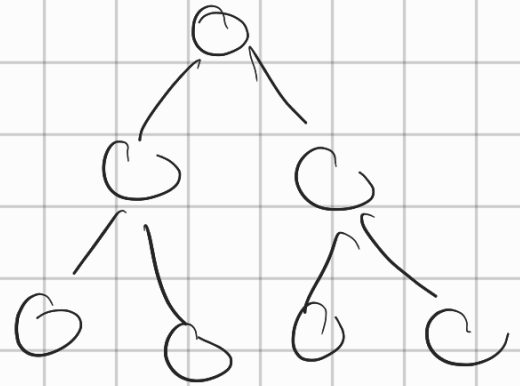
• $\max(x_A, x_B) \geq \min(x_C, x_D) \text{ \& } \max(x_C, x_D) \geq \min(x_A, x_B)$

Bentley-Ottmann Algorithmus

- fügt 3. Event in die Queue ein (Schnittpunkt-Event)
- Veränderung der Datenstruktur
 - Implementierung einer prio. Queue durch ein min-Heap

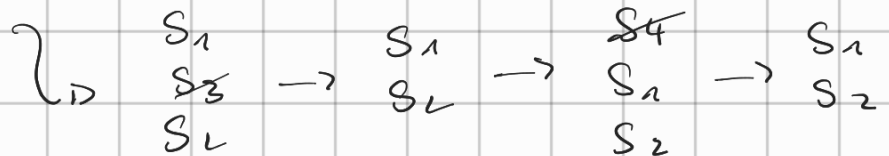
Wichtige Fragen:

1. Wann fertig \leadsto Wenn Queue leer
2. Was passiert am Schnitt?
 - Schnitt Event wird entfernt
 - S_A & S_B werden vertauscht und beide auf neue Schnitte mit ihren neuen Nachbarn getestet



*Wichtig bevor Schnitt-Event in Heap geworfen wird überprüfen,

ob Event bereits existiert



3. Wie werden die Schnittpunkte berechnet?

1. $D = (x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)$

2. $x = \frac{(x_1 y_2 - y_1 x_2)(x_3 - x_4) - (x_1 - x_2)(x_3 y_4 - y_3 x_4)}{D}$

$$y = \frac{(x_1 y_2 - y_1 x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 y_4 - y_3 x_4)}{D}$$

Allgemeiner Ablauf (Shamos Hoey)

1. Linker Endpunkt

- Füge S in Baum
- Prüfe ob S nachbarn schneidet

2. Rechter Endpunkt

Lösche S aus Baum +

- Finde Nachbarn von S & prüfe für die max. 2 Nachbarn, ob sie sich mit ihren Nachbarn schneiden.

3. Abbruch bei Schnitt

4. Wenn Queue leer \rightarrow kein Schnitt.

Allgemeiner Ablauf (Bentley)

1. Linker Endpunkt

- Füge S in Baum
- Prüfe ob S nachbarn schneidet
 - ↳ Wenn ja Schnittpunkt berechnen + neues Event

2. Rechter Endpunkt

Lösche S aus Baum +

- Finde Nachbarn von S & prüfe für die max. 2 Nachbarn, ob sie sich mit ihren Nachbarn schneiden.

3. Bei Schnitt

- Swappe S_1 & S_2 lösche Schnittevent
- prüfe S_1 & S_2 mit neuen Nachbarn auf Schnitt.