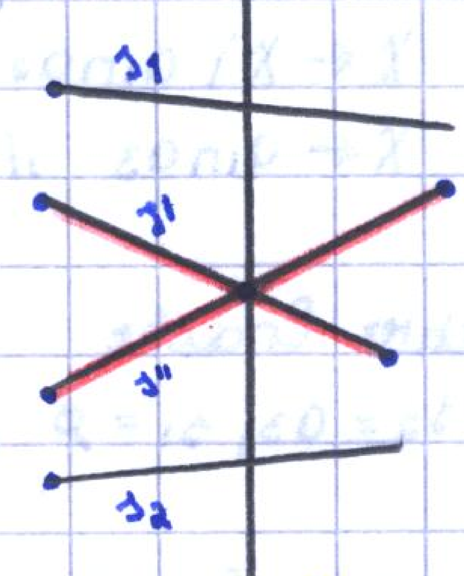
3.2.9. Sweep-Alg. für Segment-Schnitt:

```
SWEEP(S)                                  // S = Menge von n Segmenten
                                          // Ausgabe: alle SchnittPkte.
X-Struktur   X ← ∅
Y-Struktur   Y ← ∅
double xpos ← -∞;                         // Pos. des SL.
forall s∈S do
      X.insert (s.left);                  // linker Endpkt ≙ s.left
      X.insert (s.right);                 // rechter Endpkt ≙ s.right
od;
// Hauptschleife: (ähnlich ∀ Sweep-Alg.).
while (¬X.empty()) {
      p ← X.findemin();
      X.delete(p)
      xpos ← p.xcord();                   // schiebt SL zum Pkt p.
      // Fallunterscheidung gemäß verschiedenen Events:
      switch (p) {
            case: p linker Endpkt von s
                  Y.insert(s);
                  s1 ← Y.succ(s);         // ex. nicht immer
                  s2 ← Y.pred(s);         //
                  X.delete(s1∩s2);        // Invariante! s1, s2 keine Nachbarn mehr.
                  X.insert(s1∩s);         // ex. nicht immer
                  X.insert(s2∩s);         //
            case: p rechter Endpkt von s
                  s2 ← Y.succ(s);
                  s1 ← Y.pred(s);
                  Y.delete(s);
                  X.insert(s1∩s2);        // s1∩s2 rechts von SL
            case: p SchnittPkt von s' und s".
                  s1 ← Y.succ(s');
                  s2 ← Y.pred(s");
                  Y.swap(s', s");
                  X.delete(s1∩s');
                  X.delete(s2∩s");
                  X.insert(s1∩s");
                  X.insert(s'∩s2);
                  Ausgabe: "p = s'∩s"."
      end switch
od
```



3.2.10. Laufzeit:

balancierter Suchbaum nach y-Koord, der höchst. n Segm. speichert ⇒ Höhe = $O(\log n)$

1) Alle Operationen auf X und Y brauchen höchstens Zeit $O(\log n)$
2) Daraus folgt: Initialisierung benötigt Zeit $O(n \log n)$
   (2n Endpkte einfügen)
3) Hauptschleife $2n + s$ mal ausgeführt, wobei $s = \#$ SchnittPkte.
   Jeder Schleifendurchlauf kostet $O(\log n)$, da in jedem Fall konstant viele Operationen auf X- und Y-Struktur und konstant viele Schnitttests ausgeführt werden.
   In jm Fall: insert ≤ 2 mal, delete ≤ 2 mal, restliche Operationen: konst. Zeit
   ⇒ pro Schleifendurchlauf ≤ $4 \cdot \log n$ + restl. konst. Operationen.
   ⇒ $O(\log n)$ pro Schleifendurchlauf

⇒ **Gesamtlaufzeit ist somit: $O((n+s) \cdot \log n)$**

Zu X-Str: balanciert Suchbaum.

Erster Pkt einfügen ⇒ $O(1)$

Zweiter Pkt einfügen ⇒ $O(1)$

⋮

n Pkte drin ⇒ Höhe: $O(\log n)$

Weitere n fehlen noch! Pro jeden weiteren aus n übrigen brauchen wir $O(\log n)$ Zeit

Gesamt n Pkte, die noch reinmussen ⇒ $n \cdot O(\log n) = O(n \log n)$.