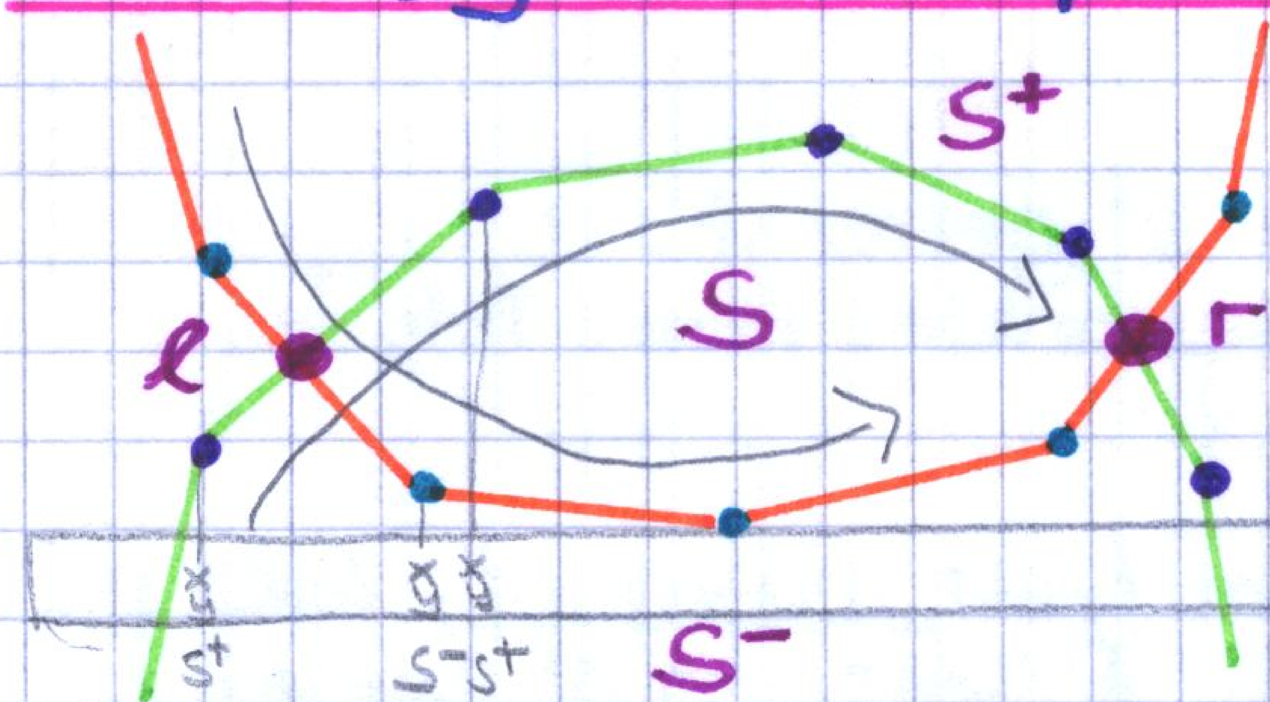
- **Laufzeit:** für S^+ bzw S^-
  S^+ kann in Zeit $\underbrace{O(m)}_{D,D^-\ 1\text{ weil } m\text{ Pkte/Geraden}} + \underbrace{O(m\log m)}_{\text{Graham's Scan}} = O(m\log m)$ berechnet werden.
  Falls $l_1, \dots, l_m$ nach Steigung sortiert
  $\Rightarrow O(m)$ (~> siehe Graham's Scan)
  S^- analog

- **Berechnung der Schnittpkte von S^+ und S^-:**



r und l sind neue Pkte !!
Jeweils mit Merge die Pkte aus S^+ und S^- in ein Feld einfügen und gleichzeitig die y-Koord in zwei Int Variablen speichern (diese heißen s^+ und s^-). Dabei ist die eine immer kleiner als die andere, bis sich das irgendwann umdreht ⇒ Schnittpkt kurz davor

VS: Die Ränder sind sortiert
Berechnung durch Mischen (indem S in der Mitte durchgeschnitten wird, dann erhält man eine steigende und eine fallende Folge)
Laufzeit: $O(n)$

- **Laufzeit insgesamt:**
  Der Schnitt von n Halbebenen kann in Zeit $O(n\log n)$ berechnet werden.
  Falls die Geraden nach Steigung sortiert sind
  $\Rightarrow O(n)$

1.2.3 Divide & Conquer - Algorithmus

Problem:
- Geg.: 2 konv Polygone $P = (p_1, \dots, p_m)$, $Q = (q_1, \dots, q_l)$, abgeschl
  (voriger Abschnitt: 2 konvexe offene Polygone) Eckenfolgen sind gegen UZS gegeben !!
- Ges.: $P \cap Q$ bzw die Folge der Ecken gegen den UZS

- **Idee:**
  Zerlege die Ebene in Regionen, sodass $\forall$ Regionen R, $(P \cap Q) \cap R$ einfach zu berechnen ist.

- **Definition:**
  Zeichne durch jede Ecke von P und Q eine senkrechte Gerade.
  Jeweils 2 benachbarte Senkrechte definieren eine Region, nämlich die Fläche dazwischen.
  $\Rightarrow R \cap P$ und $R \cap Q$ sind für jedes R ein Trapezoid, d.h. sie haben konstante Größe (4Eck).