3.4.5. Point-Location allgemein (unabh. v. Voronoi-Diagramm).

⟼ 3.4.5.1. Problem:

Geg: Beliebige planare Unterteilung der Ebene (Subdivision)
Ges: point location.
dh. finde für beliebigen Pkt p das Gebiet in dem p liegt.
⇔ finde die Kante der Unterteilung, die von einem vertikalen Strahl von p aus nach unten zuerst getroffen wird.
dh. die die man "sieht", wenn man von p nach unten schaut.
(→ vertikales Sichtbarkeitsproblem).

Schreibe den Namen (Nummer, ID, ...) jedes Gebiets an seine untere Folge von Kanten.
⇒ wenn untere Kante ausgeg. wird, so weiß man in welchem Gebiet p liegt.
⇒ daher sind die Aussagen äquivalent.

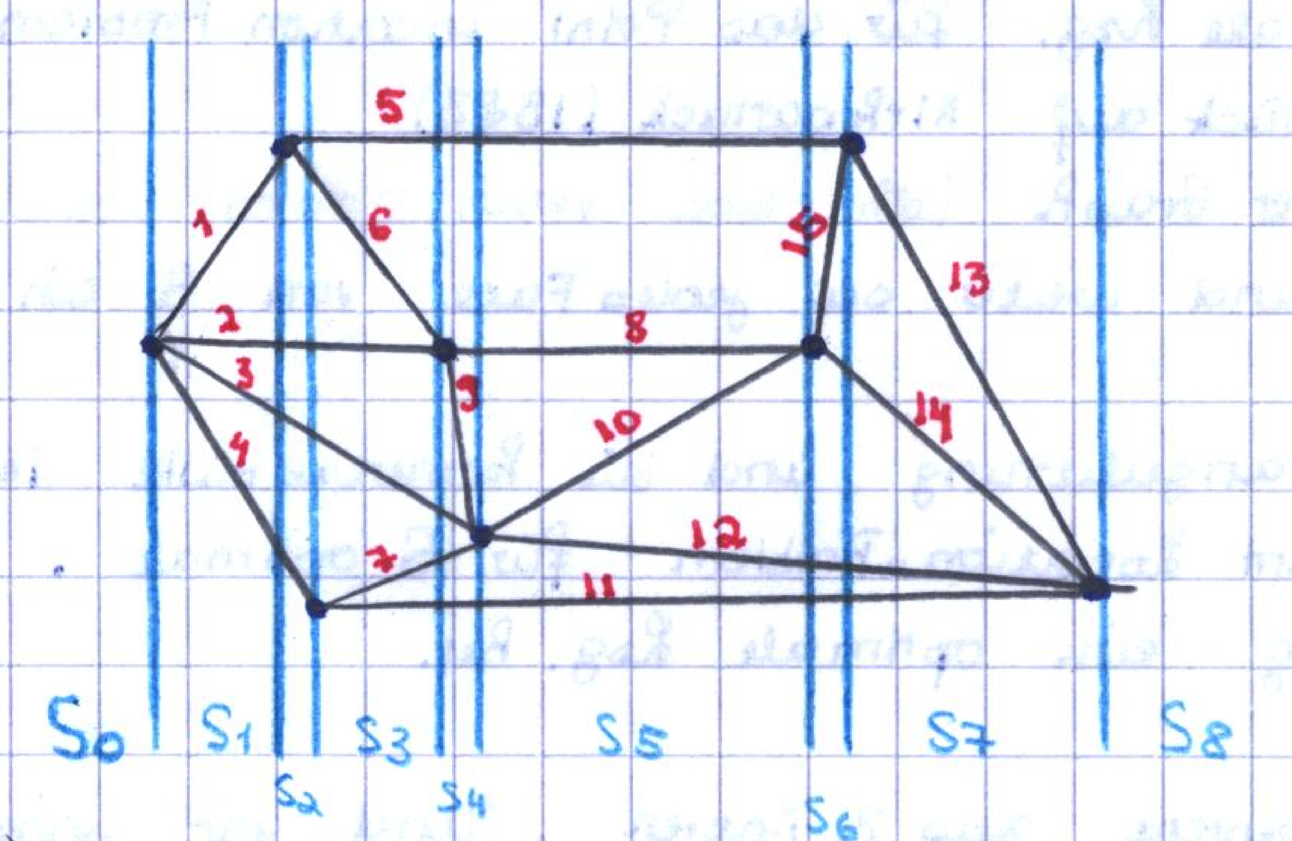
⟼ 3.4.5.2. Streifenmethode: allgemein:

die Streifenmethode zerlegt dieses 2-dim. Suchproblem in 2 1-dimensionale.

⟼ 3.4.5.2 Streifenmethode: Idee:

Zerlege die Ebene in vertikale Streifen $S_0, S_1, \ldots, S_n$ durch n vertikale Geraden jeweils durch einen Knoten von G.
G = planare Unterteilung
n = # Knoten von G.

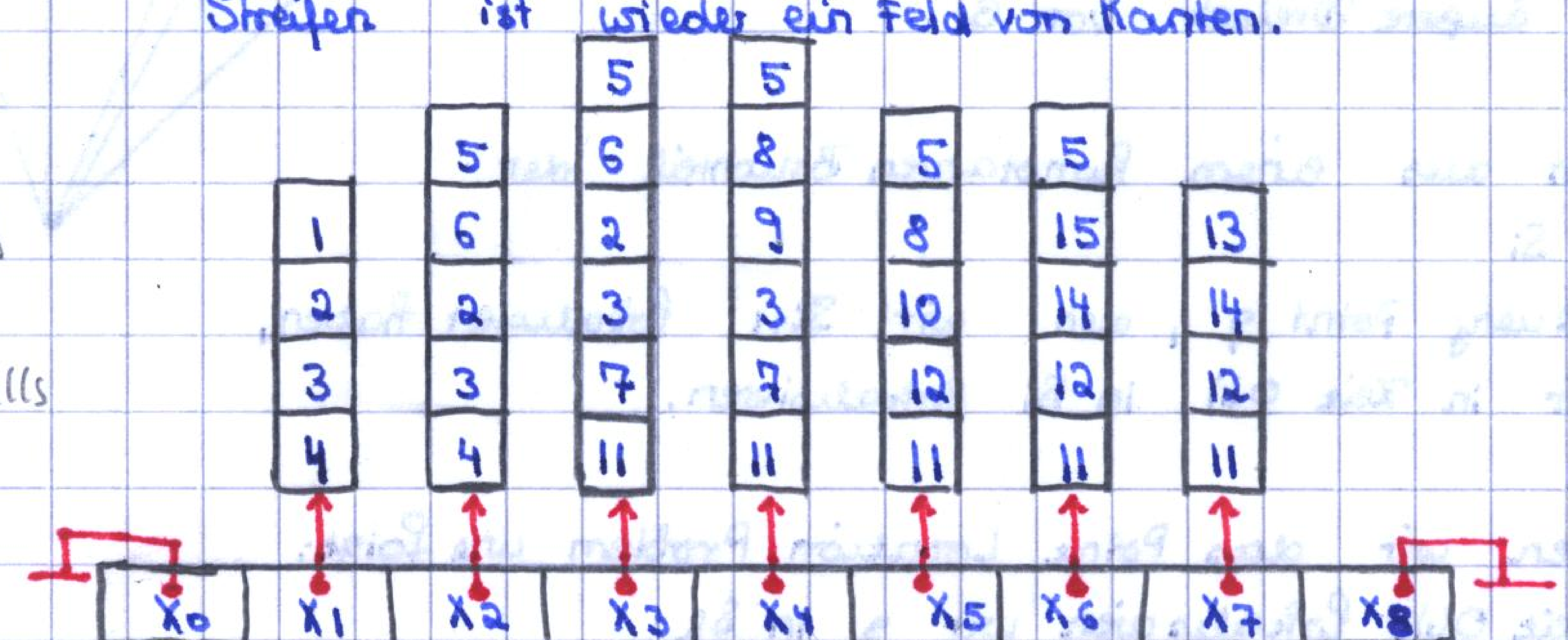
Preprocessing: $\Theta(n^2)$ im schlechtesten Fall.



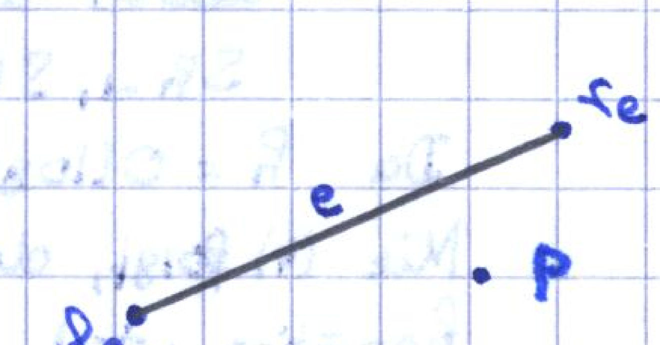
Speichere die Streifen in einem Feld S[0..n].
Schritt 1: Finde den Streifen S[i], der p enthält durch binäre Suche nach p.xcord() in S
→ kostet Zeit $O(\log n)$.

Darstellung jedes Streifens S[i]:
Eine Folge der Kanten von G, die S_i überqueren von unten nach oben (gemäß ihrer Lage in S_i) sortiert.
Bem: Kanten schneiden sich nicht innerhalb eines Streifens, dh. jeder Streifen ist wieder ein Feld von Kanten.

Ein Feld von 2 dim Pkten bzw. Intervallen.
$x_0 := [-\infty, x_0], [x_0 x_1], [x_1, x_2], \ldots, [x_8, \infty]$
Zweites Feld von Zeigern:
Zeiger zeigen auf Felder von ebenfalls 2dim Pktn.



Schritt 2: Binärsuche auf Streifen S[i] selbst.
genauer: Sortiere p in Folge der Kanten ein.
Liegt oberhalb, auf oder unterhalb einer Kante e
⇝ orientation $(l_e, r_e, p) = ?$



$\left.\begin{array}{l}\text{Da insgesamt } O(n) \text{ Kanten} \\ \text{(G ist planarer Graph)}\end{array}\right\} \Rightarrow$ kostet auch Zeit $O(\log n)$

Zu Laufzeit: binäre Suche ist $T(n) = T\left(\frac{n}{2}\right) + c_1 \Rightarrow f(n) = c_1, \log_b a = \log_2 1 = 0$
$\Rightarrow f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \cdot \log n) = \Theta(\log n)$