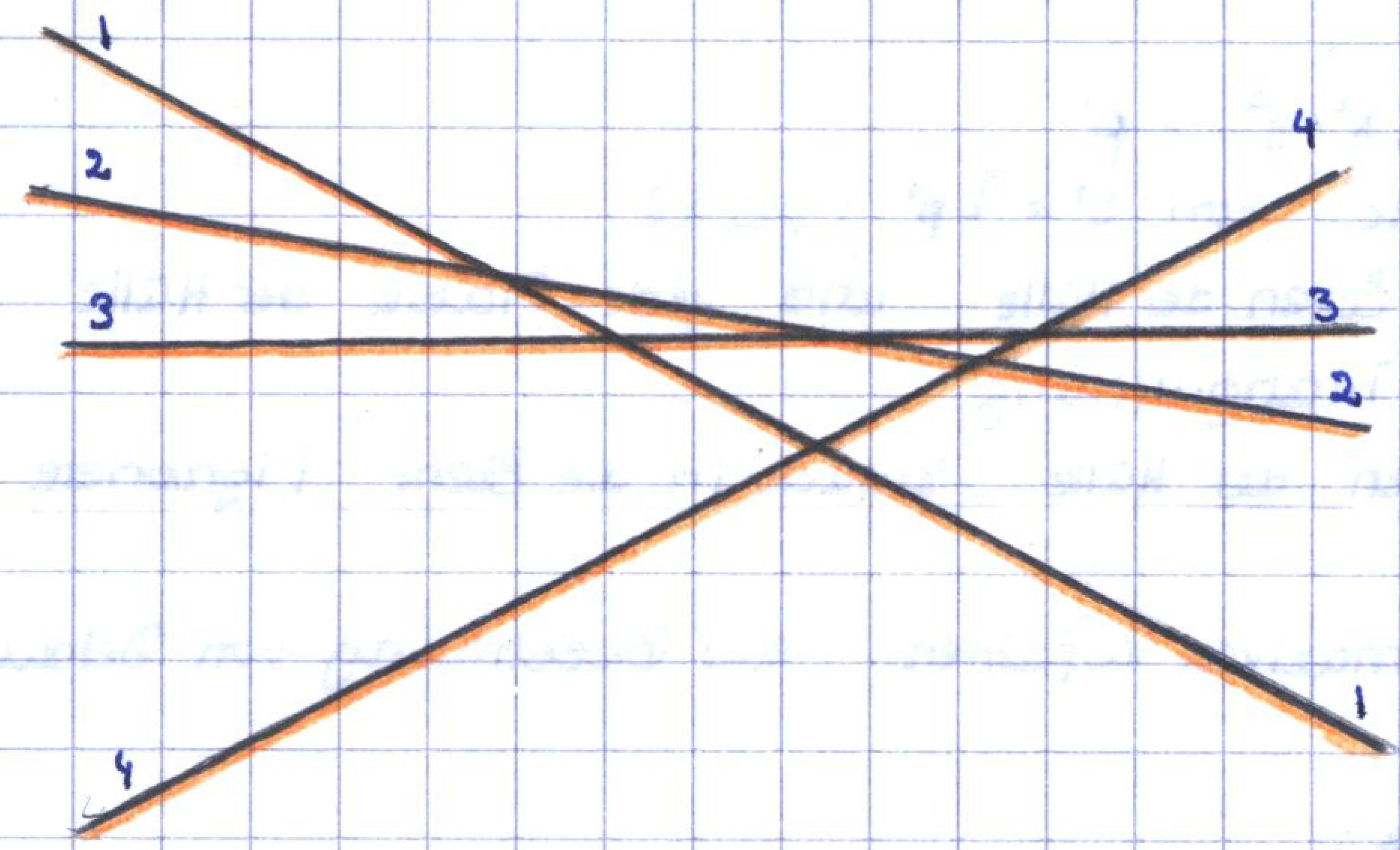


→ 6.4.1.6. Laufzeit: $O(n^2 \log n)$

→ 6.4.1.7 Bem. Man kann G auch in Zeit $O(n^2)$ berechnen.

→ Algorithmus später

($\Rightarrow$ dann Laufzeit: $O(n^2)$).

6.4.2. Problem 2

→ 6.4.2.1. Problem: Geg: $p_1 .. p_n \in \mathbb{R}^2$

Finde 3 paarweise verschiedene Pkte $p_i p_j p_k$ so, dass die Fläche des Dreiecks $\Delta p_i p_j p_k$ minimal ist.

Es gilt offensichtlich: 3 Pkte auf gemeinsamer Geraden $\Leftrightarrow$ kleinste Fläche $(=0)$.

→ 6.4.2.2. Einfache Lsg:

Teste alle Dreiecke (Minimumsuche)

$\Rightarrow O(n^3)$ Laufzeit.

→ 6.4.2.3. Bessere Lsg:

Betrachte alle Paare (p_i, p_j), $i \neq j$, definiere Gerade ℓ_{ij}

Sei $near(p_i, p_j)$ = Pkt $\in S \setminus \{p_i, p_j\}$ mit min. Abstand zu ℓ_{ij}.

$\Rightarrow \Delta(p_i, p_j, near(p_i p_j))$ hat min. Fläche unter allen Dreiecken $(p_i p_j, \cdot)$

$\Rightarrow$ Lineare Suche auf der Menge $\Delta(p_i, p_j, near(p_i p_j))$ liefert das Ergebnis.

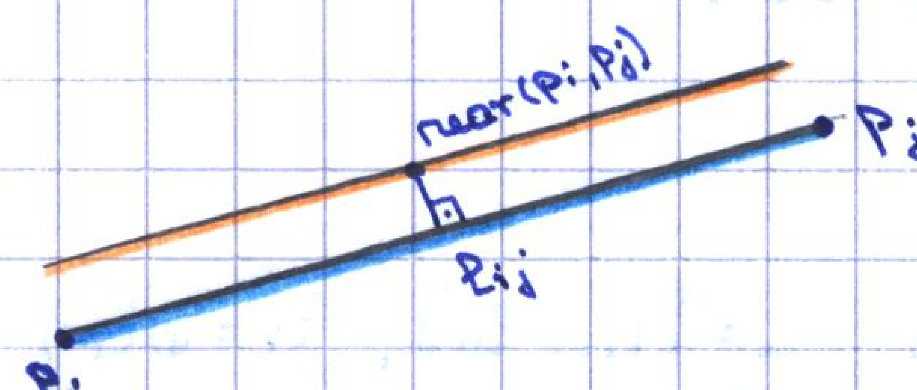
$\Rightarrow$ Laufzeit $O(n^2)$, wenn $near(p_i p_j)$ bekannt

Nun: Berechnung von $near(p_i p_j)$ $\forall$ $(p_i p_j)$

a) trivial: $O(n^3)$ (lin. Suche $\forall$ Paare)

b) Dualität.

→ 6.4.2.4 Dualität:



Betrachte Gerade ℓ durch $p_k = near(p_i, p_j)$ parallel zu ℓ_{ij}

$\Rightarrow$ dann enthält der Streifen zw. ℓ_{ij} und ℓ keinen Pkt, dh.

Beim Verschieben von ℓ_{ij} nach ℓ wird kein Pkt getroffen.

Im Dualen: Betrachte $\ell_i := D(p_i)$, $\ell_j := D(p_j)$ schneiden sich im Pkt

$p_{ij} := D^{-1}(\ell_{ij})$. Die Parallele ℓ geht über in einen Pkt $p = D^{-1}(\ell)$.

1). p hat gleiche x-Koordinate wie p_{ij} (dual zu parallelen Ge.)

2) p liegt auf $\ell_k = D(p_k)$

3) Strecke $\overline{p_{ij}\, p}$ schneidet keine Gerade außer am Rand.