1.2 Algorithmus I: Gift Wrapping.

Intuitiv: Pktmenge wird mit Geschenkpapier eingewickelt.

Zunächst: $S \subset \mathbb{R}^2$. Voraussetzung: $|S| < \infty$

Ges: $CH(S) = p_1 .. p_n$.

1.2.1. Idee: 1). Startpkt p_1 : Pkt mit der kleinsten y-Koordinate.
(Falls mehrere wähle den linkesten. Dh. Minimum in p_n - lexikogr. Ordnung.
(Übung : p_1 ist Ecke der konvexen Hülle).

2) Wie findet man p_2? (Nachfolger von p_1 auf konv. Hülle).
Betrachte horizontalen Strahl l durch p_1 (nach rechts). Drehen den Strahl gg. den Uhrzeigersinn bis wir einen weiteren Pkt von S treffen. (Ann. $|S| > 1$).
So erhalten wir p_2.
Genauer: $\forall q \in S \setminus \{p_1\}$ sei α_q der Winkel zwischen l und $\overline{pq}$.
Wähle p_2 so, dass α_{p_2} minimal. Falls mehrere Pkte mit minimalen Winkel, wähle den Pkt mit maximaler Entfernung zu p_1.
⇝ lineare Suche in S nach Minimum bzgl. oben definierter Ordnung.

3). Setze Algor. bei p_2 fort, dh. l wird nun der Strahl, der
- in p_2 beginnt.
- in Richtung p_1p_2 zeigt.

⇝ Suche Minimum in $S \setminus \{p_1, p_2\}$.

4) Wiederhole bis p_1 wieder erreicht wird.

1.2.2. Korrektheit : Übung.

1.2.3. Laufzeit: $CH(S) = p_1 .. p_k$, $|S| = n$.
p_1: lineare Suche in S kostet $O(n)$ (lin. Suche nach Min. bzgl. p_y)
$p_2 .. p_k$: lin. Suche in S kostet $O(n)$ (lin. Suche nach Winkelordnung).
⇒ Gesamtlaufzeit: $O(k \cdot n)$.

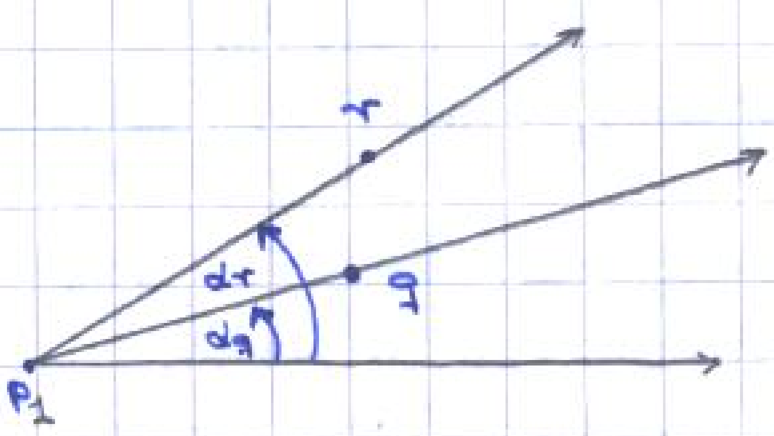
1.2.4. Satz: Sei S eine Menge von n Pkten im $\mathbb{R}^2$ und k die Zahl der Ecken der konvexen Hülle CH(S). Dann kann CH(S) in Zeit $O(k \cdot n)$ berechnet werden. Bew. siehe oben.

1.2.5. Bem: $k \in \{1..n\}$
Worst Case : $k = n$ (zB. alle Pkte aus S liegen auf einem gemeinsamen Kreis, das Innere ist leer). ⇒ Laufzeit: $O(n^2)$.
Bester Fall: k = const. ⇒ Laufzeit : $O(k \cdot n) = O(n)$.

1.2.6. Details der Implementierung:

1. Wie findet man Pkt q so, dass α_q minimal ist?
⇝ Wir müssen Winkel vergleichen.

```
q ← undefiniert
αq ← ∞
forall p ∈ S \ {p1} do
    if αp < αq then
        q ← p
        αq ← αp
    fi
od
```



Im Allg. 3 Pkte p, q, r → vergleiche Winkel
Naiv: Berechne Winkel aus Koordinaten der Pkte mit trigonometrischen Fktnen.
⇒ Nachteile:
- langsam
- Präzisionsprobleme
- Robustheitsproblem (Definitionsbereich der trigonometrischen Fktnen ist eingeschränkt).

⇒ Besser: Andere Betrachtungsweise:
Dazu betrachte Dreieck p, q, r
Drei mögliche Orientierungen:

a) positiv orientiert (left-turn) ⇒ $\alpha_q < \alpha_r$

b) Orientierung 0 (collinear) ⇒ $\alpha_q = \alpha_r$

c) negativ orientiert. (right-turn) ⇒ $\alpha_q > \alpha_r$