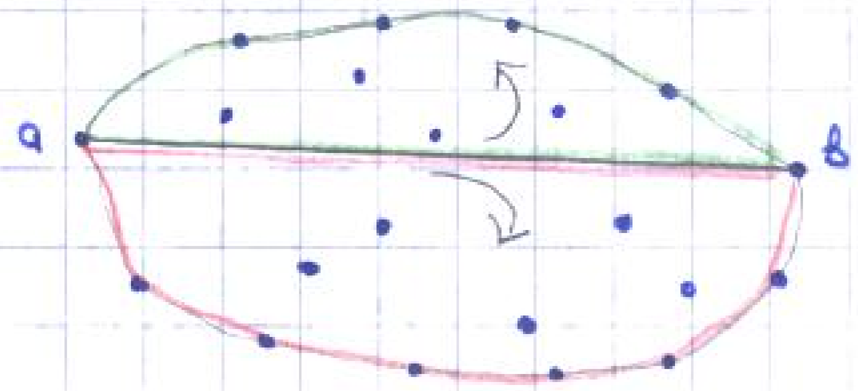
1.3. Algorithmus II: Graham's Scan.

1.3.1. Idee: Berechne "obere" und "untere" Hülle getrennt.

Sei a der linkeste und b der rechteste Pkt von S, dh. a = Minimum bzgl. x-y-Sortierung und b = Maximum bzgl. xy-Sortierung.

Obere Hülle = Der Teil von CH(S), der oberhalb von $\overline{ab}$ liegt (inclusive a,b). $= CH(S_1), S_1 \subset S$.

Untere Hülle = Der Teil von CH(S), der unterhalb von $\overline{ab}$ liegt (incl. a,b) $= CH(S_2), S_2 \subset S$.



Beobachtung: $S_1 = \{p \in S : \text{orientation}(a,b,p) \geq 0\}$
und $S_2 = \{p \in S : \text{orientation}(a,b,p) \leq 0\}$.

Sei S_1 lexikografisch nach xy-sortiert ($\rightarrow n \log n$)
$\rightsquigarrow$ sortierte Folge $q_1 \dots q_n = S_1$
$\Rightarrow a = q_1$ und $b = q_n$
Berechne obere Hülle als Teilfolge $x_1 .. x_m$ von $q_1 .. q_n$
(dh. im Uhrzeigersinn).

Beobachtung:

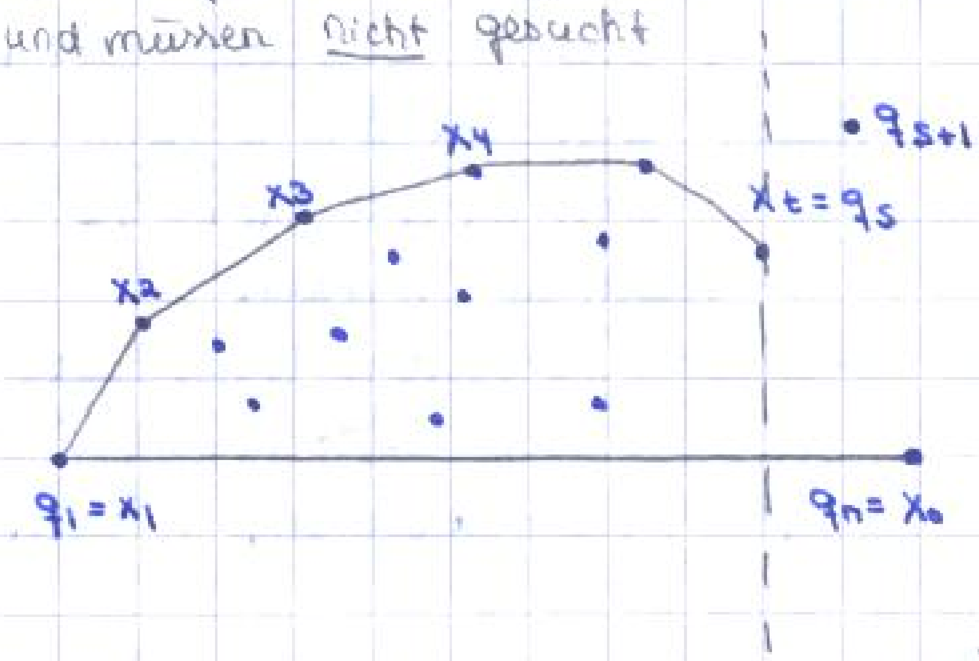
1). Jeweils drei aufeinanderfolgende Ecken x_i, x_{i+1}, x_{i+2} bilden einen Rightturn, dh. $\text{orientation}(x_i, x_{i+1}, x_{i+2}) < 0$.
2). (b,a,p) ist Rightturn $\forall p \in S_1$.
(dh. $\forall p \in S_1 : \text{orientation}(q_n, q_1, p) < 0$).

1.3.2. Algorithmus:

verwalte stack $x_0, x_1 .. x_t$ von potentiellen Ecken der oberen Hülle.
Am Ende: genau die Ecken der oberen Hülle.
Initialisierung: stack $S = q_n, q_1, q_2$. Bem: alle drei Pkte wg. lexik. Sort. genau festgelegt und müssen nicht gesucht werden.
Schleife: Betrachte nur $q_3 .. q_{n-1}$.

Sei q_s aktueller Pkt
Invariante: $x_0, x_1 .. x_t$ ist Teilfolge von $q_n, q_1 .. q_s$

mit: 1) $t \geq 2$, $x_0 = q_n$, $x_1 = q_1$, $x_t = q_s$
besser x_1 statt x_0 2) $x_0, x_1 .. x_t$ ist konvexes Polygon
3) obere Hülle von S ist Teilfolge von $x_1 .. x_t\, q_{s+1} .. q_n$.



Schritt: $s \mapsto s+1$

```
while (x_{t-1}, x_t, q_{s+1}) nicht Rightturn do
      pop(x_t)
od
push(q_{s+1}).
```

Beobachtung: stack speichert obere Hülle der Pkte $q_1 .. q_s$ (die bis jetzt gewesen).

Die Funktion:

UPPER_HULL($q_1 .. q_n$)   besser: $q_1 \dots q_m \rightarrow b = q_m$ (weil $|S| = n$)
(Vorbed. 1). $q_1 .. q_n$ lexik. nach xy sortiert
2) $q_2 .. q_{n-1}$ liegen oberhalb $\overline{q_1 q_n}$).
Stack von Pktn. S
(stack <point> S;)
S.push(q_n);
S.push(q_1);
S.push(q_2);
(Ann. $n \geq 2$).

→