- for $s = 3$ to $m-1$ do
  - $x \leftarrow S.\mathrm{top}()$ $\quad O(1)$
  - $y \leftarrow S.\mathrm{top\_pred}()$
  - while orientation $(y, x, q_s) \geq 0$ do $\quad$ ↓dh left-turn oder colinear, also kein right-turn
    - $S.\mathrm{pop}()$ $\quad O(1)$
    - $x \leftarrow y$ $\quad O(1)$
    - $y \leftarrow S.\mathrm{top\_pred}()$ $\quad O(1)$ $\quad \Big\} \; O(m-3)$
  - od
  - $S.\mathrm{push}(q_s)$ $\quad O(1)$
- od
- return S $\quad$ S enthält die obere Hülle

- Invariante:

Es ist $x_0, x_1, \dots, x_t$ ist Teilfolge von $q_m, q_1, \dots, q_s$ mit:

- $\left.\begin{array}{l} x_0 = q_m \\ x_1 = q_1 \\ x_t = q_s \end{array}\right\} t \geq 2$ dh der Stack ist nie leer
- ~~x_0~~ $x_1, \dots, x_t$ konvexes Polygon
- obere Hülle von S ist Teilfolge von: $\overset{q_1}{x_1}, \dots, \overset{q_s}{x_t}, q_{s+1}, \dots, q_m$

ist immer erfüllt.

$\Rightarrow$ Korrektheit des Algorithmus

- Algorithmus CONVEX-HULL (S): in diesem Alg ist S eine Liste !!
  - $S.\mathrm{sort}(xy)$ $\quad O(n \log n)$ dominiert.
  - $a \leftarrow S.\min$ $\quad O(1)$
  - $b \leftarrow S.\max$ $\quad O(1)$
  - forall $p \in S$ do
    - if (orientation $(a,b,p) > 0$) $\quad S_1.\mathrm{append}(p)$ "rechts" einfügen
    - if (orientation $(a,b,p) < 0$) $\quad S_2.\mathrm{append}(p)$
  - od $\quad$ "links" einfügen $\quad O(n)$
  - $S_1.$ ~~append~~ push (a) $\quad O(1)$
  - $S_1.\mathrm{append}(b)$ $\quad O(1)$
  - $S_2.\mathrm{push}(a)$ $\quad O(1)$
  - $S_2.\mathrm{append}(b)$ $\quad O(1)$
  - $H_1 \leftarrow$ UPPER-HULL (S_1) $\quad O(m)$
  - $H_2 \leftarrow$ LOWER-HULL (S_2) $\quad O(n-m)$
  - H_1 umdrehen $\quad O(m)$
  - H_2 an H_1 anhängen $\quad O(1)$
  - a und b einmal löschen $\quad O(1)$

- Laufzeit: ohne Sortieren !!

$|S| = n$

Betrachtung der while-Schleife:

- Der Rumpf hat Kosten $O(1)$
- Da jeder Pkt höchstens einmal aus S entfernt werden kann, wird die while Schleife höchstens $|S| = n$ mal durchlaufen

$\Rightarrow$ insgesamt: $O(n)$

Das ist optimal !!

Denn: mit Sortierung $\Rightarrow O(n \log n)$ und wir wissen: Berechnung von $CH(S)$ ist mind. so schwer wie Sortieren !!

- Bemerkung

Wenn # Ecken von $CH(S) = h < \log n$ $\Rightarrow$ Alg I besser als Alg II

Alg I: $O(n \cdot h)$

Alg II: $O(n \cdot \log n)$