

(2) Bei Termination erhält Stack S die obere Hülle.

Voraussetzung: die Invariante ist für den letzten Pkt ($s=n$) erfüllt.

d.h. alle Pkte sind abgearbeitet.

Wissen also: $x_1 \dots x_t$ ist TF von $q_1 \dots q_n$ mit $bea: \text{läuft ab } 1$, weil $x_0 = x_t = q_n$.

1) $t \geq 2$, $x_0 = q_n$, $x_1 = q_1$ und $x_t = q_n$

2) $x_1 \dots x_t$ konv. Polygon

3) obere Hülle ist TF von $x_1 \dots x_t$.

z.z. obere Hülle = $x_1 \dots x_t$ also nicht echte Teilmenge.

Ann: nein, obere Hülle $\subset x_1 \dots x_t$

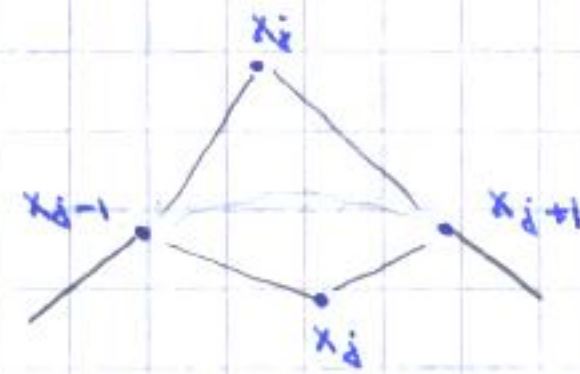
$\Rightarrow \exists x_j \in \{x_1 \dots x_t\}$ mit $x_j \notin$ obere Hülle

\Rightarrow 1. Fall: orientation(x_{j-1}, x_j, x_{j+1}) = +1 \Rightarrow \hat{z} zu 2)

2. Fall: orientation(x_{j-1}, x_j, x_{j+1}) = -1 \Rightarrow \hat{z} zu 2).

\Rightarrow Ann. falsch

\Rightarrow Beh.



13.5 Laufzeit:

Beobachtung: Jeder Pkt kann höchstens einmal aus S entfernt werden.

Rumpf hat Kosten $O(1)$ (orientation, konstante Zahl von stack-Operationen).

Da jeder Pkt höchstens einmal aus S entfernt werden kann \Rightarrow

\Rightarrow while-Schleife: wird höchstens n mal eingetreten. \Rightarrow forschleife kostet uns $O(n)$.

In der while-Schleife konst. Operationen, außerhalb auch

\Rightarrow Gesamtkosten der UPPER-Pkt: $O(n)$.

(Analog berechnet man die untere Hülle).

Die Funktion insgesamt:

CONVEX_HULL(S)

1. S.sort(x_y);

a \leftarrow S.min();

b \leftarrow S.max();

2. forall p \in S do eigentlich S1 {a, b}

3. if (orientation(a, b, p) > 0) S1.append(p)

4. if (orientation(a, b, p) < 0) S2.append(p)

5. od

6. S1.push(a);

S1.append(b);

7. S2.push(a);

S2.append(b);

8. H1 \leftarrow UPPER_HULL(S1)

9. H2 \leftarrow LOWER_HULL(S2)

10. H1 umdrehen

11. H2 an H1 anhängen

12. doppeltes Vorkommen von a/b löschen.

$O(n \log n)$ ($|S|=n$)

$O(1)$

$O(1)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(1)$

$O(n)$

$O(1)$

$O(m) = O(n)$

$O(n-m) = O(n)$

$O(n)$

$O(1)$

$O(n)$

(S1, S2 Listen \Rightarrow append kostet $O(1)$, $n \times \Rightarrow O(n)$)

(falls S1 Liste \Rightarrow einfügen am Ende kostet $O(n)$)

Zusammenfassung:

CONVEX_HULL(S)

1) Sortiere & lexik.

2) a \leftarrow min \wedge b \leftarrow max

3) S1 \leftarrow {p \in S: b, a, p Rightturn} \cup {a, b}

4) S2 \leftarrow {p \in S: b, a, p Leftturn} \cup {a, b}

5) UPPER_HULL(S1)

6) LOWER_HULL(S2)

7) " Zusammenkleben

Zeit:

$O(n \log n)$

$O(1)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$

$O(n)$