

## Übungsblatt 1 - Konvexe Hüllen

**Ausgabe:** Mittwoch, 16. April 2014

**Abgabe:** Dienstag, 22. April 2014

### 1 Einfacher Algorithmus für konvexe Hülle

In der Vorlesung wurde der Algorithmus `FIRSTCONVEXHULL(P)` vorgestellt, der für eine gegebene Punktmenge  $P$  die konvexe Hülle berechnet. Hierzu werden im letzten Schritt die berechneten Kanten in eine sortierte Knotenliste  $L$  von  $CH(P)$  überführt. In der Vorlesung wurde gezeigt, dass dies in  $O(n^2)$  Zeit möglich ist. Zeigen Sie, dass dieser Schritt sogar in  $O(n \log n)$  Zeit berechnet werden kann.

### 2 Algorithmus Gift Wrapping

In der Vorlesung wurde der Algorithmus *Gift Wrapping* für die Berechnung der konvexen Hülle vorgestellt. Zeigen Sie folgenden Satz und gehen Sie dabei insbesondere auf die Korrektheit des Algorithmus ein.

**Satz 1.** *Die konvexe Hülle  $CH(P)$  von  $n$  Punkten  $P$  in  $\mathbb{R}^2$  lässt sich mit Gift Wrapping (auch Jarvis' March) in  $O(n \cdot h)$  Zeit berechnen, wobei  $h = |CH(P)|$ .*

Welche degenerierten Fälle können auftreten? Wie kann man mit diesen Fällen umgehen?

### 3 Algorithmus Chan Hull

Gegeben sei ein konvexes Polygon  $P$  mit  $n$  Knoten und ein Punkt  $p$  außerhalb von  $P$ ; siehe Abbildung 1.

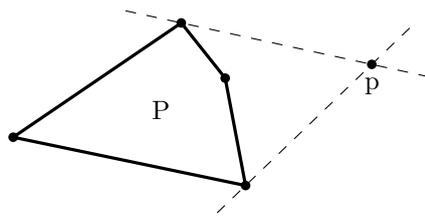


Abbildung 1: Illustration von Aufgabe 3

1. Wie kann eine Tangente an  $P$ , die durch  $p$  führt, in  $O(\log n)$  Zeit berechnet werden, wenn die Ecken von  $P$  als (im Uhrzeigersinn) sortierte Liste gegeben sind?
2. An welcher Stelle im Algorithmus *Chan Hull* wird diese Berechnung benötigt?

### 4 Optimalität!

Von einem Algorithmus, der die konvexe Hülle einer gegebenen Punktmenge berechnet, fordern wir, dass er die Punkte als (im Uhrzeigersinn) sortierte Liste ausgibt.

- a) Zeigen Sie, dass jeder Algorithmus zur Berechnung der konvexen Hülle von  $n$  Punkten im schlimmsten Fall eine Laufzeit von  $\Omega(n \log n)$  hat, was bedeutet, dass *Graham Scan* optimal im Sinne der asymptotischen Laufzeit ist.  
*Hinweis:* Benutzen Sie, dass die *Sortierung* von  $n$  Schlüsseln (in gewissen Rechnermodellen) eine Laufzeit von  $\Omega(n \log n)$  benötigt.
- b) Weshalb stellt die Laufzeit von *Gift Wrapping* keinen Widerspruch zum Ergebnis aus Teilaufgabe a) dar?
- c) Gegeben sei ein einfaches, nicht notwendigerweise konvexes Polygon in der üblichen Listenrepräsentation. Geben Sie einen Algorithmus an, der die konvexe Hülle der Eckenmenge dieses Polygons in  $\mathcal{O}(n)$  Zeit berechnet. Erläutern Sie, weshalb dies keinen Widerspruch zum Ergebnis aus Teilaufgabe a) darstellt.