Karlsruher Institut für Technologie
Fakultät für Informatik
ITI Wagner

Algorithmische Geometrie
Sommersemester 2012
Martin Nöllenburg/Andreas Gemsa

Übungsblatt 1 - Konvexe Hüllen

Ausgabe: Dienstag, 17. April 2012
Abgabe: Dienstag, 24. April 2012

1 Einstieg

In einigen Verfahren zur Berechnung der konvexen Hülle muss bestimmt werden ob ein Punkt $p \in \mathbb{R}^2$ links oder rechts von einer gegebenen Linie liegt. In der Vorlesung wurde erwähnt, dass man das mit Hilfe der Berechnung der Determinante einer bestimmten Matrix durchführen kann. Im Folgenden soll diese Behauptung bewiesen werden.

Seien die Punkte $p = (p_x, p_y)$, $q = (q_x, q_y)$ und $r = (r_x, r_y)$ gegeben. Die gerichtete Gerade g verlaufe durch die Punkte p und q. Wir nehmen an, dass r nicht auf g liegt.

a) Beweise, dass das Vorzeichen der Determinante $\det(A)$ der Matrix

$$A = \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}$$

anzeigt ob r sich links oder rechts von g befindet.

b) Da sich die drei Punkte p, q und r in allgemeiner Lage befinden induzieren sie ein Dreieck. Zeige, dass $|\det(A)|$ der doppelte Flächeninhalt dieses Dreiecks ist.

2 Algorithmus zur Berechnugn der konvexen Hülle

Wir betrachten nun eine alternative zu dem in der Vorlesung vorgestellten Algorithmus zu Berechnung der konvexen Hülle von P Punkten in der Ebene.

Die grundlegende Idee des Algorithmus' ist wie folgt: Bestimme den rechtesten Punkt der Eingabemenge. Nenne diese Punkt p_1. Lege eine vertikale Gerade durch p_1 und rotiere sie im Uhrzeigersinn um p_1 bis die Gerade einen zweiten Punkt, p_2, trifft. Rotiere die Gerade weiter, aber jetzt um p_2 bis die Gerade einen dritten Punkt p_3 trifft. Wiederhole das Verfahren so lange bis die Gerade wieder p_1 trifft.

bitte umblättern

a) Gebe für den Algorithmus eine Beschreibung in Pseudocode an.

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

c) Zeige, dass dieser Algorithmus die konvexe Hülle *korrekt* berechnet.

d) Zeige, dass dieser Algorithmus in $O(n \cdot h)$ implementiert werden kann, wobei h die Anzahl der Punkte auf der konvexen Hülle ist.

3 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) Zeige, dass jeder Algorithmus zur Berechnung der konvexen Hülle von n Punkten eine Laufzeit von $\Omega(n \log n)$ benötigt, was bedeutet, dass der Algorithmus aus der Vorlesung optimal im Sinne der asymptotischen Laufzeit ist.

   Hinweis: Benutze, dass die *Sortierung* von n Schlüsseln (in gewissen Rechnermodellen) eine Laufzeit von $\Omega(n \log n)$ benötigt.

b) Gegeben sei ein einfaches, nicht notwendigerweise konvexes Polygon in der üblichen Listenrepräsentation. Gebe einen Algorithmus an, der die konvexe Hülle der Eckenmenge dieses Polygons in $\mathcal{O}(n)$ Zeit berechnet. Erläutere, warum dies keinen Widerspruch zum Ergebnis aus Teilaufgabe a) darstellt.