Technische Universität Berlin
Fakultät II – Institut für Mathematik
Prof. Günter M. Ziegler
Axel Werner

Wintersemester 2004/2005
Algorithmische Geometrie

4. Übungsblatt

Abgabe am 25.11.2004

Aufgabe 13: **4 Punkte**

Sei P ein Polygon (ohne Löcher) und $p \in P$. Definiere $\mathrm{star}(p\,; P) := \{q \in P \mid p \text{ sieht } q\}$ (p sieht q, wenn das Intervall $\overline{pq}$ ganz in P liegt). P ist *sternförmig*, wenn es einen *Sternpunkt* $p \in P$ gibt, d.h. einen Punkt $p \in P$, so dass $\mathrm{star}(p\,; P) = P$. Der *Kern* von P ist $\ker(P) := \{p \in P \mid \mathrm{star}(p\,; P) = P\}$ (also die Menge aller Sternpunkte).

i. Zeige: $\mathrm{star}(p\,; P)$ ist ein Polygon.

ii. Ist $\ker(P)$ konvex? Falls ja, gib einen Beweis an, sonst ein Gegenbeispiel.

iii. Zeige: P ist konvex $\Leftrightarrow \ker(P) = P$.

iv. Sei $Q = \mathrm{star}(p\,; P)$ (nach i. wieder ein Polygon). Zeige: $\ker(P) \subseteq \ker(Q)$.

Aufgabe 14: **4 Punkte**

Es sei P ein Polygon mit n Ecken. Gib einen Algorithmus zur Berechnung von $\ker(P)$ (s. Aufgabe 13) an, der Laufzeit $O(n \log n)$ hat.

Aufgabe 15: **4 Punkte**

Gegeben sei eine Menge von n Punkten in der Ebene. Die paarweisen Strecken zwischen den Punkten induzieren eine Unterteilung der konvexen Hülle der Punkte in konvexe Polygone. Für $i \in \{3, \ldots, n\}$ sei p_i die Anzahl der dabei entstehenden i-Ecke. Entwirf einen effizienten Algorithmus, welcher $(p_3, \ldots, p_n)$ berechnet. Was ist die Laufzeit bzw. der Speicherplatzbedarf dieses Algorithmus?
Tip: Modifiziere den Sweep-Algorithmus zum Berechnen der Schnitte zwischen Strecken.

Aufgabe 16: **4 Punkte**

Für ein Polygon P heißen die Zusammenhangskomponenten der Menge $\mathrm{conv}(P) \setminus P$ die *Taschen* von P. Es sei nun ein Polygon P_1 mit m Ecken gegeben, sowie eine Triangulierung von P_1 und der Taschen von P_1. Weiterhin sei P_2 ein konvexes Polygon mit m Ecken. Zeige, dass sich der Schnitt $P_1 \cap P_2$ in $O(m+n)$ Zeit berechnen lässt.

Programmieraufgabe 2: **Abgabe bis 9.12.2004**

Wir wollen den Algorithmus aus Aufgabe 15 implementieren – am besten in zwei Schritten:

1. CGAL stellt eine Implementierung eines geradlinigen, planaren Graphen mittels einer DCEL zur Verfügung. Auf der homepage der Vorlesung gibt es ein Programm, das eine solche graphisch darstellt; modifiziere es, füge Ecken und Kanten ein, lies Regionen aus etc.

2. Implementiere einen Algorithmus, der n Punkte in der Ebene trianguliert. Input soll eine Liste von Punkten sein, Output eine DCEL, welche die in Aufgabe 15 beschriebene Unterteilung der konvexen Hülle der Punkte repräsentiert, sowie ein Vektor, der die Anzahlen p_i der entstandenen i-Ecke enthält.