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

Wintersemester 2004/2005
Algorithmische Geometrie

1. Übungsblatt

Abgabe am 4.11.2004

Aufgabe 1: **4 Punkte**

Wir betrachten die folgenden beiden Varianten von Chan's Algorithmus: In der Schleife in Step 3 berechne g nach der Vorschrift

$$\text{(i)} \quad g = 2^t \qquad \text{bzw.} \qquad \text{(ii)} \quad g = t.$$

Zeige, dass dann für die Gesamtlaufzeit die folgenden Schranken gelten:

$$\text{(i)} \quad O(n \cdot (\log h)^2) \qquad \text{bzw.} \qquad \text{(ii)} \quad \Theta(n\, h \log h).$$

Aufgabe 2: **4 Punkte**

Sei P ein (nicht notwendigerweise konvexes) Polygon mit $n > 3$ Ecken $p_0, \ldots, p_{n-1}$ in sortierter Reihenfolge. Eine *Diagonale* in P ist ein Liniensegment $p_i p_j$ ($i \neq j$), welches zwei nicht benachbarte Ecken (also $|i - j| \neq 1$ und $\{i, j\} \neq \{0, n-1\}$) verbindet und ganz (bis auf die Endpunkte) im Inneren liegt.
Zeige:

1. P besitzt immer eine Diagonale $p_{i-1}p_{i+1}$ für ein i (und Indizes modulo n).

2. Ist P konvex, dann existiert mindestens ein i, so dass $\bigcirc(p_{i-1}, p_i, p_{i+1})$ ganz P enthält. Hierbei ist $\bigcirc(p_{i-1}, p_i, p_{i+1})$ der durch p_{i-1}, p_i und p_{i+1} aufgespannte Kreis.

Aufgabe 3: **4 Punkte**

Gegeben seien zwei Strecken $S_1 = p_1p_2$ und $S_2 = q_1q_2$ im $\mathbb{R}^2$. Beschreibe einen Algorithmus, der effizient testet, ob S_1 und S_2 sich schneiden. Dabei sollen keine Divisionen verwendet werden. Welchen Vorteil hat diese Einschränkung?

Aufgabe 4: **4 Punkte**

Gegeben seien zwei Polygone P_1 und P_2 mit jeweils mindestens 3 Ecken, welche keine überlappenden Kanten haben. Wie groß ist die maximale Anzahl der Schnittpunkte der beiden, falls

i. beide konvex sind,

ii. beide nicht notwendigerweise konvex sind.

Gib Beispiele an, bei denen die obere Schranke angenommen wird. (Bei (ii.) hängt die Antwort davon ab, ob $|P_1|$ und $|P_2|$ gerade oder ungerade sind!)

Bitte die Rückseite beachten!

Programmieraufgabe 1: **Abgabe bis 11.11.2004**

Wir wollen den Algorithmus von Chan zur Berechnung der konvexen Hülle von Punkten in der Ebene mit CGAL implementieren. Dazu kann das Rahmenprogramm zur Konvexe-Hülle-Berechnung von der homepage verwendet werden. Die Programmierung des Algorithmus selbst erfolgt am besten in zwei Schritten:

i. Zunächst sollte Graham's scan in einem eigenen Unterprogramm implementiert und sorgfältig getestet werden.

ii. Danach wird dieses Unterprogramm entsprechend dem 1. Schritt in Chan's Algorithmus angepasst und die Schritte 2 und 3 implementiert, die dann den umgebauten Graham-Algorithmus verwenden.

Testet die verschiedenen Algorithmen (Slow, Graham und Chan) mit den verschiedensten Eingabedaten (sehr viele Punkte; Konfigurationen, bei denen die konvexe Hülle alle Eingabepunkte enthält etc.) und vergleicht die tatsächliche Laufzeit.