

Algorithmische Geometrie

Helmut Alt (Claudia Dieckmann, Sven Scholz)

Abgabe 03.07.2009 **vor** der Vorlesung

Aufgabe 1 Delaunay-Triangulierung

20 Punkte

Schreiben Sie ein Programm, das die Delaunay-Triangulierung einer Menge $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ von n Punkten in der Ebene randomisiert inkrementell berechnet.

Verwenden Sie dazu folgenden Algorithmus:

$\text{Trian}(P^{(i-1)})$ sei die Delaunay-Triangulierung nach den ersten $i - 1$ Schritten.

- Wähle p_i zufällig aus $P \setminus P^{(i-1)}$ aus.
- $P^{(i)} = P^{(i-1)} \cup \{p_i\}$
- Finde das Dreieck $\{q, r, s\}$, in dem p_i liegt.
- Füge die Strecken $\overline{p_i q}$, $\overline{p_i r}$, $\overline{p_i s}$ ein.
- Für jedes neu erzeugte Dreieck teste, ob im Inneren des Umkreises ein Punkt aus P liegt.
- Falls ja, "flippe" die entsprechende Kante.
- Fahre so lange fort, bis keine Kanten mehr "geflippt" werden müssen.

Kommentieren und testen Sie Ihr Programm ausreichend und erstellen Sie eine ausführbare .jar Datei, die Sie ihrem Tutor per Mail schicken.

Sie brauchen die Laufzeit des Algorithmus **nicht** analysieren.