

Übungsblatt 7 - Point Location

Ausgabe: 24. Mai 2011

Abgabe: 31. Mai 2011

1 Keine Vorberechnung

In der Vorlesung wurde ein Algorithmus besprochen, der das *point location* Problem mit Hilfe eines Vorberechnungsschritts löst. Im Folgenden werden wir annehmen, dass sowohl die Polygon-Unterteilungen als auch der Punkt für die Punktanfrage gleichzeitig mitgeteilt werden. Deshalb lohnt sich der Vorberechnungsschritt nicht mehr. [**Gilt auch für Aufgaben 2 und 3**].

Sei P ein einfaches Polygon bestehend aus n Knoten und sei q der Anfrage-Punkt. Es folgt eine schriftliche Beschreibung eines Algorithmus' der bestimmt ob q im Inneren von P liegt:

Sei $\rho := \{(q_x + \lambda, q_y) : \lambda > 0\}$ (horizontale Kante die durch die Punkt q verläuft). Ermittle für jede Kante e des Polygons P ob sie ρ schneidet. Ist die Gesamtanzahl der Kanten aus P die ρ schneiden ungerade, so liegt q im Inneren von P .

- Zeige die Korrektheit des Algorithmus.
- Erkläre wie man mit degenierten Fällen umgeht (Ein Beispiel für einen degenierten Fall ist, dass ρ einen Endpunkt einer Kante schneidet.)
- Bestimme die Laufzeit dieses Verfahrens.

2 Spezielle Polygone

Sei P ein *konvexes* Polygon bestehend aus n Knoten. Die Knoten sind in sortierten Reihenfolge (bzgl. des Rands des Polygons) gegeben.

- Zeige, dass es man in $\mathcal{O}(\log n)$ Zeit bestimmen kann, ob ein gegebener Punkt q innerhalb von P liegt.
- Kann man das Resultat aus a) für den Fall verallgemeinern, dass P kein konvexes sondern ein y -monotones Polygon ist? [Erläutern]

Bitte wenden

3 Spezielle Polygone II

Ein Polygon P heißt *sternförmig*, wenn es einen Punkt p im Inneren von P gibt, so dass für jeden anderen Punkt q im Inneren von P gilt, dass die Verbindungsstrecke pq vollständig in P enthalten ist. Nehme im Folgenden an, dass solch ein Punkt p zusammen mit einem sternförmigen Polygon P gegeben ist.

- a) Zeige, dass man in $\mathcal{O}(\log n)$ Zeit bestimmen kann ob ein Punkt q im Inneren von P liegt oder nicht.
- b) Was ist, wenn der Punkt p nicht gegeben ist? Kann man das Verfahren aus a) so anpassen, dass es auch ohne den Punkt p funktioniert? [Erläutern]

4 Das Ray-Shooting Problem

Das *ray shooting* Problem tritt häufig beim Rendern von computergenerierten 3D-Grafiken auf. Die 2D Variante lautet wie folgt: Verwalte eine Menge S bestehend aus n sich nicht schneidenden Streckensegmenten so, dass man schnell eine Anfrage vom Typ: “Gegeben ein query-Strahl ρ —eine Halbgerade die in einem Punkt startet. Finde das erste Segment aus S das von ρ geschnitten wird.”

Im Folgenden betrachten wir nur das *vertical ray shooting* Problem bei dem der query-Strahl nur vertikal “geschossen” wird. Das bedeutet insbesondere, dass für solch eine Anfrage nur der Startpunkt definiert werden muss.

- a) Gebe eine Datenstruktur für das vertical ray shooting Problem an. Gebe eine möglichst scharfe Schranke für die worst-case Laufzeit und den worst-case Speicherplatzverbrauch an.
- b) Kann man den Ansatz aus a) so modifizieren, dass er auch funktioniert, wenn die Streckensegmente sich schneiden dürfen?