Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **WS 10/11**
TU Braunschweig

Prof. Dr. Sándor Fekete
Henning Hasemann

Algorithmische Geometrie
Übung 4 vom 10. 01. 2011

Abgabe der Lösungen am Montag, den 24. 01. 11, vor der Übung im SN19.4.
Bitte die Blätter vorne deutlich mit eigenem Namen versehen!

Aufgabe 1 (Voronoi): Sei $P = \{p_1, p_2, \dots, p_n\}$ eine Punktmenge und p_2 ein Punkt aus P mit kleinstem Abstand zu p_1. Zeige: Die Voronoi-Zellen von p_1 und p_2 in $Vor(P)$ grenzen aneinander (d.h. sie teilen sich eine Voronoi-Kante).

(Hinweis: Du darfst folgende Annahme treffen: Wenn zwei Voronoi-Zellen $V(p_1)$ und $V(p_2)$ nicht benachbart sind, werden sie von mindestens zwei weiteren Voronoi-Zellen $V(p_3)$ und $V(p_4)$ "getrennt", so dass $|p_3 - p_4| < |p_1 - p_2|$ gilt.)

(40 P.)

Aufgabe 2 (Untere Schranke für Voronoi): Angenommen, ein Algorithmus ALG könnte das Voronoi-Diagramm einer Punktmenge P mit $|P| = n$ mit Komplexität $O(f(n)) \subset O(n \log n)$ (also schneller als $O(n \log n)$) berechnen.
Beweise, dass dies nicht möglich ist, indem du zeigst, dass man ALG benutzen könnte um n Zahlen in $O(n + f(n))$ zu sortieren.

(40 P.)

Aufgabe 3 (Voronoi Lookup): Gegeben: Eine Punktmenge P mit $n \in \mathbb{N}$ Punkten und ihr Voronoidiagramm $Vor(P)$. Gib ein Verfahren zur Konstruktion einer Datenstruktur an, die es erlaubt, zu einem beliebigen Punkt $q \in \mathbb{R}^2$ die nächste Site zu finden (d.h. festzustellen, in welcher Voronoi-Zelle er sich befindet).

Das Erzeugen der Datenstruktur darf beliebig aufwendig sein, danach soll das Finden der nächsten Sites für beliebige q je in Zeit $O(\log n)$ erfolgen können. Erläutere, warum dein Verfahren die Lookup-Zeit von $O(\log n)$ einhält.

(Hinweis: In dem Fall, dass q auf einer Voronoi-Kante oder einem Voronoi-Knoten liegt, ist die nächste Site nicht eindeutig bestimmt. Dann sollte der Lookup einfach irgendeine der nächsten Sites liefern.)

(40 P.)