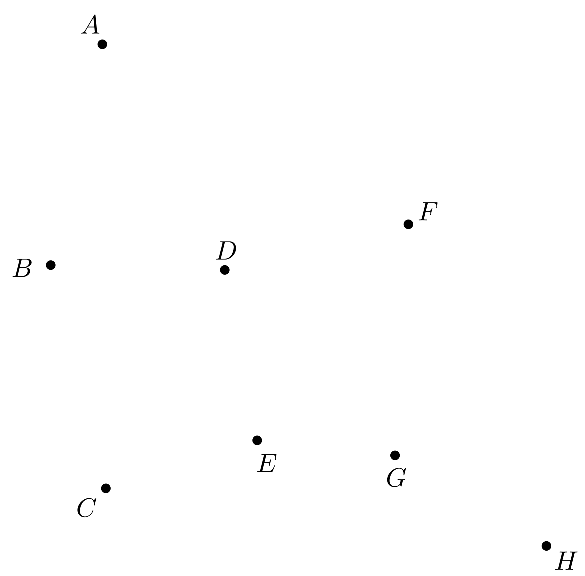
Grundlagen der Algorithmischen Geometrie SS 2016
Übungsblatt 12
Universität Bonn, Institut für Informatik I

- *Dieser Übungszettel wird **nicht** mehr besprochen. Er ist nur zur persönlichen Vertiefung des Stoffes der letzten beiden Wochen gedacht.*

Aufgabe 1: Farthest Point Delaunay-Triangulation

Gegeben sei eine Menge P von n Punkten in allgemeiner Lage in der Ebene. Dabei soll allgemeine Lage bedeuten, dass keine vier Punkte auf einem Kreis liegen. Die Farthest-Point-Delaunay-Triangulation von P, kurz FPDT(P), ist wie folgt definiert: Drei Punkte p, q und $r \in P$ bilden ein Dreieck, falls der Umkreis dieses Dreiecks alle Punkte aus $P \setminus \{p, q, r\}$ enthält.

a) Ermitteln Sie FPDT(P) für die gegebene Punktmenge.



b) Beweisen Sie: FPDT(P) ist eine Triangulation der konvexen Hülle von P.

Aufgabe 2: Bisektor-Variante

Der *Bisektor* zwischen zwei geometrischen Objekten O_1 und O_2 in der euklidischen Ebene ist die Menge aller Punkte (b_x, b_y), die den gleichen euklidischen Abstand zu den beiden Objekten haben. Der euklidische Abstand $d((b_x, b_y), O)$ zwischen einem Punkt (b_x, b_y) und einem Objekt O ist dabei durch den kürzesten euklidischen Abstand zu einem Punkt des Objektes gegeben.

Wie sieht der Bisektor zwischen einem Kreis und einem Punkt im Inneren des Kreises aus? Begründen Sie ihre Antwort.

Aufgabe 3: Voronoi-Diagramme und Delaunay-Triangulation

Wir betrachten das Einfügen des Punktes p_i, wenn die Delaunay-Triangulation der Punktmenge $S_{i-1} = \{p_1, \ldots, p_{i-1}\}$ schon berechnet wurde.

a) Beweisen oder widerlegen Sie: Für einen Punkt p_i außerhalb der konvexen Hülle $\mathrm{ch}(S_{i-1})$ ist jede Kante, die p_i mit einem von p_i aus sichtbaren Eckpunkt von $\mathrm{ch}(S_{i-1})$ verbindet, eine Delaunay-Kante in $\mathrm{DT}(S_i) = \mathrm{DT}(S_{i-1} \cup \{p_i\})$.

b) Beweisen oder widerlegen Sie: Wenn man zu $\mathrm{DT}(S_{i-1})$ alle in a) betrachteten Kanten hinzunimmt, erhält man die Delaunay-Triangulation $\mathrm{DT}(S_i)$.

Aufgabe 4: Pledge-Algorithmus

Betrachten Sie den Pledge-Algorithmus zur Navigation in einem unbekannten Labyrinth.

Der Algorithmus folgt jedem Stück Mauer von jeder Seite höchstens einmal. Versuchen Sie, ein Labyrinth zu konstruieren, in dem der Algorithmus möglichst großen Anteilen der Mauer folgt. Sie bestimmen die Startrichtung des Algorithmus.

Ist es möglich ein Labyrinth zu konstruieren, in dem der Algorithmus jedem Stück Mauer von min. einer Seite folgt? Ist es möglich, ein Labyrinth zu konstruieren, in dem der Algorithmus jedem Stück Mauer von beiden Seiten vollständig folgt?