

Aufgabensammlung zum Buch „Algorithmen und Datenstrukturen“ (Kapitel 7)

1 Aufgaben aus dem Buch

Zu folgenden Aufgaben, die direkt aus dem Buch entnommen sind, gibt es an der Universität Freiburg am Lehrstuhl Ottmann Musterlösungen. In der Version mit Lösungen sind diese angegeben. Hinter der fortlaufenden Aufgabennummer steht in Klammern die Nummer der Aufgabe im Buch.

Aufgabe 1 (Aufgabe 7.9):

Gegeben sei die Menge $\{A, B, C, D, E, F\}$ von Intervallen mit

$$A = [2, 3], B = [5, 9], C = [1, 4], D = [3, 7], E = [6, 8] \text{ und } F = [8, 10].$$

1. Geben Sie einen Intervallbaum möglichst geringer Höhe zur Speicherung dieser Intervallmenge an.
2. Führen Sie eine Aufspießanfrage für den Punkt $x = 3$ durch und geben Sie an, in welcher Reihenfolge die aufgespießten Intervalle entdeckt werden (ausgehend vom Intervallbaum aus a)).

2 Ähnlich Aufgaben

Bei den folgenden Aufgaben handelt es sich um Aufgaben die an der ETH Zürich, am Institut für Theoretische Informatik und an der Universität Freiburg im Institut für Informatik in diversen Vorlesungen gestellt wurden. Inhaltlich sind diese Aufgaben mit dem behandelten Stoff im Buch verwandt. Zu allen Aufgaben gibt es Musterlösungen, die allerdings nur in der Version mit Lösungen enthalten sind.

Aufgabe 2:

Für den Algorithmus zur Berechnung der konvexen Hülle müssen wir testen, ob ein Punkt r auf der rechten Seite von der geraden \overline{pq} liegt.

1. Beweisen Sie, daß das Vorzeichen der Determinante

$$D = \det \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}$$

entscheidet, ob r auf der rechten oder linken Seite liegt.

2. Zeigen Sie, daß $|D|$ doppelt so groß ist, wie die Dreiecksfläche von p, q und r .

Nutzen Sie hierbei den folgenden Sachverhalt: Für den Inhalt A eines unregelmäßigen n -Ecks in der Ebene mit den (entgegen dem Uhrzeigersinn gelesenen) Endpunkten $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) = (x_0, y_0)$ gilt:

$$A = \frac{1}{2} \sum_{k=0}^{n-1} (x_k y_{k+1} - x_{k+1} y_k)$$

Im Uhrzeigersinn gelesene Endpunkte ergeben den Wert $-A$.

Aufgabe 3:

Zeigen Sie den Satz von Euler

Sei G ein kreuzungsfreier Graph in der Ebene, v die Anzahl der Knoten, e die Anzahl der Kanten, f die Anzahl der Flächen (inklusive der unbegrenzten äußeren Fläche) und c die Anzahl der Zusammenhangskomponenten. Dann gilt $v - e + f = c + 1$.

durch vollständige Induktion.

Aufgabe 4:

Wenn es im Vergleich zur Anzahl der Eingabepunkte n wenig Eckpunkte k in der konvexen Hülle gibt so ist es günstiger, einen Algorithmus zu konstruieren, der sowohl in Abhängigkeit von k als auch n steht. Betrachte o.B.d.A. nur die Berechnung obere Hülle. Die Idee ist es, ausgehend vom linken Punkt die Punkte auf der Hülle nacheinander zu bestimmen.

1. Beschreiben Sie die Vorbereitungen und den Gang über die Hülle genauer.
2. Was ist die erzielte Laufzeit Ihrer Implementation?

Aufgabe 5:

Für n orthogonale Objekte in der Ebene bieten sich Divide-And-Conquer Algorithmen zur Lösung von geometrischen Problemen an. Es gilt das folgende algorithmische Prinzip: *Falls Objektmenge klein genug, löse das Problem direkt, sonst zerteile das Problem in (möglichst) gleichgroße Teilprobleme (divide) und löse die Teilprobleme rekursiv (conquer), anschließend füge die Teillösungen zu einer Gesamtlösung zusammen (merge)*. Der Vorteil von DAC Algorithmen ist ihre Einfachheit, selten werden komplizierte Datenstrukturen verwendet.

Anstatt die Objektmenge in nur zwei Teilmengen aufzuteilen, zerlegt eine vertikale Gerade die geometrischen Objekte allgemein in drei Teile: Die Objekte die vollständig links bzw. rechts von der Geraden liegen und die Objekte, die von der Geraden geschnitten werden. Für das folgende Problem beschreiben Sie die sich ergebenden Fallunterscheidungen im *Merge*-Schritt und zeigen Sie, daß Ihr Algorithmus $O(n \log n)$ Zeit und $O(n)$ Platz benötigt.

Test auf orthogonaler Segmentschnitt: Sei S Menge horizontaler und vertikaler Segmente in getrennter Darstellung (L, R, V) , d.h. eine Menge L von linken, eine Menge R rechten Endpunkten horizontaler Segmente und eine Menge V vertikaler Segmente. Gesucht ist die Antwort auf die Frage, ob es in S ein Paar sich schneidender Segmente gibt.

Aufgabe 6:

Die *optimale* Triangulation eines Polygons $P = (v_0, \dots, v_{n-1}, v_n)$ mit $n + 1$ Knoten ist definiert für eine Gewichtsfunktion w , die durch die Seiten und Flächen der Triangulation festgelegt ist. Das Problem ist es, eine Triangulation zu finden, die die Summe der Gewichte minimiert. Eine mögliche Gewichtsfunktion ist z.B. $w(\Delta v_i v_j v_k) = |v_i v_j| + |v_j v_k| + |v_k v_i|$, wobei $|v_i v_j|$ der euklidische Abstand von $|v_i|$ zu $|v_j|$ ist.

Die entscheidende Betrachtung ist die Substruktur einer optimalen Triangulation: Eine optimale Triangulation, die das Dreieck $\Delta v_0 v_k v_n$ enthält, zerfällt in zwei Polygone (v_0, v_1, \dots, v_k) und $(v_k, v_{k+1}, \dots, v_n)$, deren geerbte Triangulation wiederum optimal ist.

Sei $t[i, j]$ das Gewicht einer optimalen Triangulation des Polygons (v_{i-1}, \dots, v_j) , dann ist initial $t[i, i] = 0$. Beschreiben Sie, wie die Tabelle der Werte $t[i, j]$, $1 \leq i < j \leq n$ dynamisch aktualisiert werden kann, um einen $O(n^3)$ Algorithmus zur w -optimalen Triangulation eines Polygons zu erhalten.

Aufgabe 7:

Beantworten Sie die folgenden zwei Fragen zur Triangulation von Polygonen und begründen Sie Ihre Antwort!

1. Kann man zu jedem Eckpunkt eines einfachen Polygons eine Diagonale finden?
2. Läßt sich jedes einfache Polygon so triangulieren, daß jedes Dreieck höchstens zwei Nachbardreiecke hat?

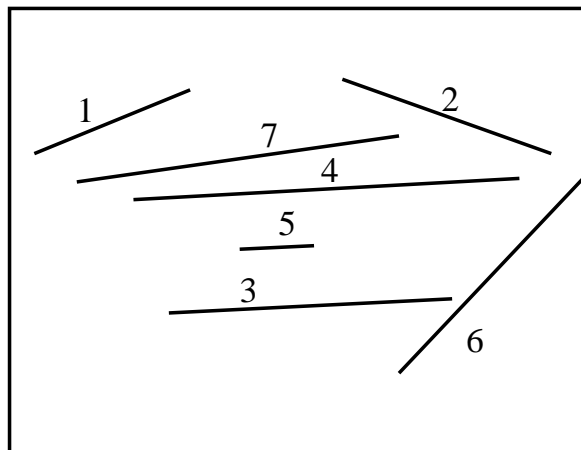
Aufgabe 8:

Zum Verständnis der lineare Optimierung dienen die zwei folgenden Aufgaben:

1. Wieviel potentielle Extrempunkte gibt es in einem linearen Programm mit n Variablen x_1, \dots, x_n mit $x_i \geq 0$ und m weiteren Nebenbedingungen maximal?
2. Minimieren Sie graphisch die Zielfunktion $x - y$ in der xy -Ebene mit den Nebenbedingungen $1 \leq x \leq 3, 0 \leq y$ und $2y - x \leq 3$.

Aufgabe 9:

Finden Sie die Trapezzerlegung inklusive der assoziierten Datenstruktur für die folgende Segmentmenge in der Ebene.



In dem inkrementellen Algorithmus sollen die Segmente in der Reihenfolge ihrer Numerierung behandelt werden.

Aufgabe 10:

Randomisieren Sie die Berechnung zur konvexen Hülle. Der Algorithmus startet mit einem beliebigen Dreieck $KH_3 = p_1p_2p_3$. Sei z ein innerer Punkt und $p_i, i > 3$, ein Punkt außerhalb der derzeitig berechneten

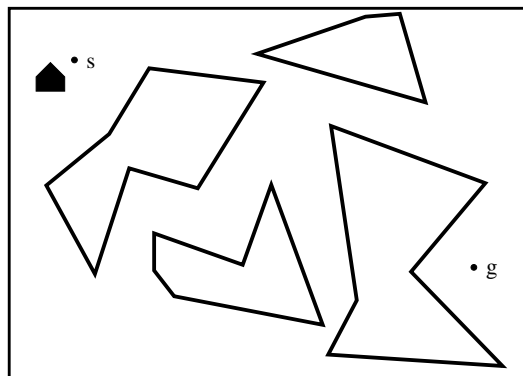
konvexen Hülle KH_{i-1} . Dann hat $\overline{zp_i}$ einen Schnitt mit einem Segment e in KH_{i-1} . Durch einen Kantenlauf von e aus werden so lange Kanten eliminiert, bis die Tangentenpunkte a und b von KH_{i-1} zu p_i erreicht werden. KH_i ergibt sich aus dem Zusammenschluß von p_i und KH_{i-1} . Dabei werden weiter Schnittsegmente aufgesucht und KH_{i-1} letztendlich um p_i erweitert.

Die Segmente werden durch Verweise (Zeiger) in einem Array A verwaltet: Für jeden noch nicht eingefügten Punkt $p_j \in \{p_i, \dots, p_n\}$ wird diejenige Kante innerhalb von KH_{i-1} gespeichert, die von $\overline{zp_j}$ geschnitten wird. Falls ein Knoten schon innerhalb der konvexen Hülle liegt, so ist der Verweis leer. Die von den Punkten aus adressierten Segmente verweisen wiederum auf die jeweiligen Punkte p_j zurück.

1. Beschreiben Sie, wie bei gegebenem Punkt p_i außerhalb der Hülle KH_{i-1} und gegebenen Tangentenpunkten a und b das Array A aktualisiert werden kann.
2. Wir sagen ein Punkt p_i mit $i < j$ steht mit p_j im Konflikt, wenn $\overline{zp_j}$ eine Kante von KH_{i-1} schneidet, die beim Einfügen von p_i entfernt wird. Die Laufzeit des Verfahrens ist durch die Anzahl der Konfliktpaare beschränkt. Geben Sie ein worst-case Beispiel an, in dem die Anzahl aller Konfliktpaare quadratisch ist.

Aufgabe 11:

Planen Sie in drei Schritten einen Weg von Startpunkt s zum Zielpunkt g für den gegebenen Roboter (Referenzpunkt sei dessen Spitze) in der folgenden Umgebung:

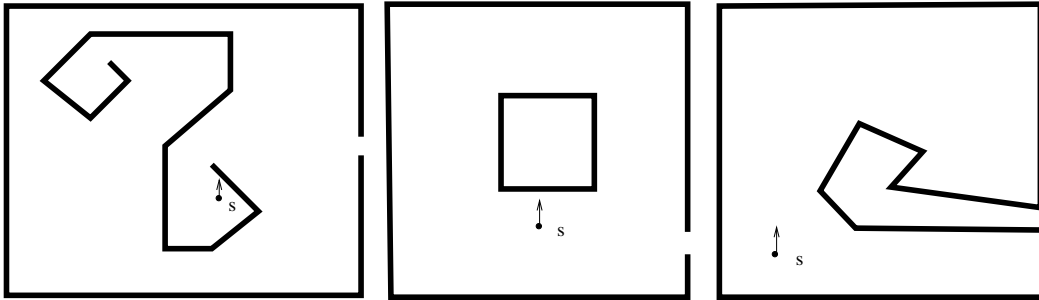


1. Projektion des Konfigurations- in den Arbeitsraum.
2. Trapezzerlegung des Bereichs inklusive der Elimination von Objekttrapezen.
3. Planung des Weges entlang der Trapez- und Seitenmitten.

Aufgabe 12:

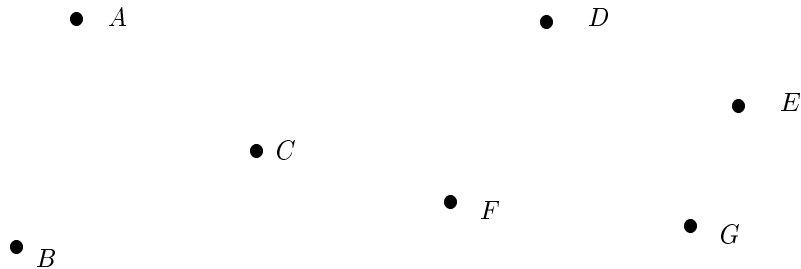
Gegeben sei ein punktförmiger Roboter, der nur über einen Tastsensor und einen Winkelzähler verfügt. Dieser Zähler kann auf 0 zurückgesetzt werden und gibt danach zu jedem Zeitpunkt die Summe der Winkel der ausgeführten Drehbewegungen an. Dabei werden Linksbewegungen positiv, Rechtsbewegungen negativ gezählt. Bei einer Wandberührung dreht sich der Roboter nach rechts. Der Roboter kann einer Richtung und einer Wand folgen, z.B. mit `repeat folge Richtung until Wandkontakt` oder mit `repeat folge Wand until Winkelzähler = 0`. Versuchen Sie sich an einem Algorithmus, der in einem polygonalen Labyrinth einen Ausweg findet, soweit er existiert.

Nehmen Sie als Beispiele die folgenden einfache Labyrinth mit Startpunkt s :



Aufgabe 13:

Gegeben sei die folgende Menge P von 7 Punkten in der Ebene:



1. Geben Sie das Voronoi-Diagramm von P an.
2. Geben Sie eine doppelt verkettete Kantenliste an, beschränkt auf alle Kanten der Voronoi-Regionen der Punkte A , B und C aus $\text{Vor}(\{A, B, C, D, E, F, G\})$.
3. Zeigen Sie graphisch, wie man aus den Voronoi-Diagramm für die Mengen $\{A, B, C\}$ und $\{D, E, F, G\}$ das Voronoi-Diagramm für die gesamte Punktmenge konstruieren kann (Merge-Schritt des Divide-and-conquer Algorithmus).

Aufgabe 14:

Zeigen Sie: Für ein Voronoi-Diagramm $\text{Vor}(P)$ zur Punktmenge $P = \{p_1, \dots, p_n\}$ gelten die folgenden beiden Aussagen:

1. Ein Punkt q in der Ebene ist ein Knoten in $\text{Vor}(P)$ genau dann, wenn der größte leere Kreis $C_P(q)$ bezüglich P um q drei oder mehrere Punkte p aus P auf seinem Rand besitzt.
2. Die Mittelsenkrechte (der Bisektor) zwischen den Punkten p_i und p_j (beide aus P) definiert eine Kante in $\text{Vor}(P)$ genau dann, wenn es einen Punkt q in der Ebene gibt, so daß der größte leere Kreis $C_P(q)$ sowohl p_i und p_j auf dem Rand enthält, aber keinen weiteren Punkt aus P .

Aufgabe 15:

Zeigen Sie: Die obere konvexe Hülle \mathcal{UH} einer Punktmenge P entspricht im Dualraum der unteren Einhüllenden \mathcal{LE} (für lower envelope) von P^* . Dabei ist $\mathcal{LE}(P^*)$ durch das Minimum der y Werte aller p^* aus P^* gegeben.

Aufgabe 16:

Betrachten Sie das Paraboloid $z = x^2 + y^2$ im 3-dimensionalen Raum und einen Punkt $p = (p_x, p_y, 0)$. Weiterhin sei die folgende dem Punkt p assoziierte nicht vertikale Ebene $z = 2p_x x + 2p_y y - (p_x^2 + p_y^2)$ mit $h(p)$ bezeichnet.

1. Zeigen Sie: Die Ebene $h(p)$ ist eine Tangentialebene an das Paraboloid zum Punkt p und schneidet eine (in z Richtung gelegene) Vertikale durch den Punkt $q = (q_x, q_y, 0)$ laufende Gerade G in dem Punkt $s = (q_x, q_y, 2p_x q_x + 2p_y q_y - (p_x^2 + p_y^2))$.
2. Zeigen Sie: Die vertikale Distanz zwischen s und dem Schnittpunkt q' von G mit dem Paraboloid ist das Quadrat der Distanz zwischen p und q .
3. Zu einer Punktmenge $P = \{p_1, \dots, p_n\}$ sei H definiert als Menge $\{h(p) | p \in P\}$ der Tangentialebenen an das Paraboloid. Desweiteren sei die obere Einhüllende $\mathcal{UE}(H)$ die Fläche, die sich durch das Maximum der z -Werte aller beteiligter Ebenen in H zusammensetzt. Zeigen Sie: Die Projektion von $\mathcal{UE}(H)$ auf die xy -Ebene ist das Voronoi-Diagramm von P . Veranschaulichen Sie sich dabei die Situation in einer Dimension niedriger.

Aufgabe 17:

Sei P ein Menge von n Punkten in der Ebene, von denen keine vier auf einem Kreis liegen, und $p_1, p_2, p_3 \in P$. Beweisen Sie, daß die Delaunay-Triangulation der duale Graph des Voronoi-Diagramms ist.

Aufgabe 18:

Sei P ein Menge von n Punkten in der Ebene, von denen keine vier auf einem Kreis liegen, und $p_1, p_2, p_3 \in P$. Sei $\overline{p_1 p_2}$ eine Delaunaykante mit angrenzenden Dreiecken $D(p_1, p_2, p_3)$ und $D(p_1, p_2, p_4)$. Zeigen Sie, daß die Voronoi-Kante zwischen p_1 und p_2 das Liniensegment zwischen den Umkreismittelpunkten von $D(p_1, p_2, p_3)$ und $D(p_1, p_2, p_4)$ ist.

Aufgabe 19:

Beweisen Sie, daß

1. die Delaunay-Triangulation den maximalen Umkreis minimiert;
2. die Delaunay-Triangulation den maximalen enthaltenden Kreis minimiert.

Aufgabe 20:

Sei P ein Menge von n Punkten in der Ebene, von denen keine vier auf einem Kreis liegen. Der *Gabriel-Graph* $G(P)$ von P ist wie folgt definiert: Eine Kante $e = \langle p_1, p_2 \rangle$ mit $p_1, p_2 \in P$ gehört zu $G(P)$, falls für alle $p_3 \in P \setminus \{p_1, p_2\}$ gilt, daß

$$(d_2(p_1, p_3))^2 + (d_2(p_2, p_3))^2 \geq (d_2(p_1, p_2))^2$$

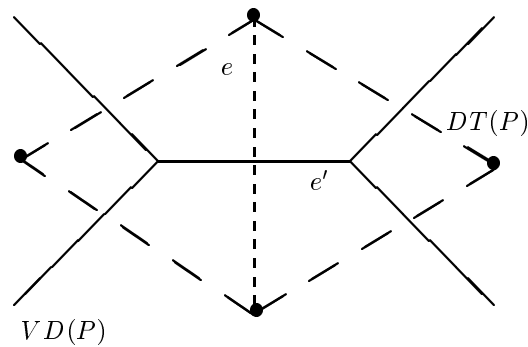


Abbildung 1: Eine Kante der Delaunay-Triangulation schneidet die duale Voronoi-Kante.

ist, wobei d_2 den euklidischen Abstand zwischen zwei Punkten bezeichne.

1. Zeigen Sie, daß der minimal spannende Baum von P ein Teilgraph des Gabriel-Graphen ist.
2. Zeigen Sie, daß jede Kante des Gabriel-Graphen auch eine Kante der Delaunay-Triangulation ist.
3. Zeigen Sie, daß $e \in DT(P)$ genau dann eine Kante von $G(P)$ ist, falls e die Kante e' des Voronoi-Diagramms schneidet—wobei e' die zu e duale Kante ist (vgl. Abbildung 1).
4. Geben Sie einen Algorithmus an, der in einer Zeit von $O(n)$ den Gabriel-Graphen berechnet, falls die Delaunay-Triangulation gegeben ist.

Aufgabe 21:

Gegeben sei die Menge $\{A, B, C, D, E, F\}$ von Intervallen mit $A = [2, 3]$, $B = [5, 9]$, $C = [1, 4]$, $D = [3, 7]$, $E = [6, 8]$ und $F = [8, 10]$.

1. Geben Sie einen Intervallbaum möglichst geringer Höhe zur Speicherung dieser Intervallmenge an.
2. Führen Sie eine Aufspießanfrage für den Punkt $x = 3$ durch und geben Sie an, in welcher Reihenfolge die aufgespießten Intervalle entdeckt werden (ausgehend vom Intervallbaum aus (a))