GRUNDZÜGE DER ALGORITHMISCHEN GEOMETRIE

Klausur

18. Juli 2008, 10:15 - 12:15 Uhr

Name: ....

Matrikelnummer: ....

Anzahl beschriebener Blätter
(ohne Aufgabenblatt): ....

Bitte beschriften Sie jedes Blatt mit Name und Matrikelnummer!

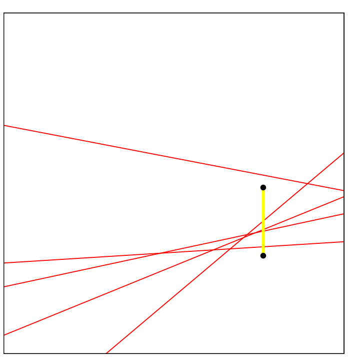
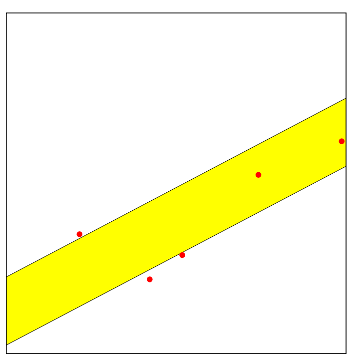
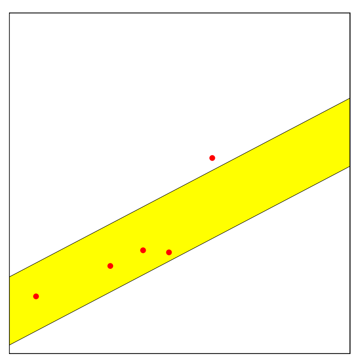
Bearbeitungszeit: 120 Minuten
Zugelassene Hilfsmittel: Keine!
Gesamtzahl Aufgaben: 6
Gesamtpunktzahl: 49

1	2	3	4	5	6	$\sum$

Aufgabe 1: (5 Punkte)
Sei $\mathscr{D}$ die Dualitätsabbildung, die Punkte wie folgt auf nicht-vertikale Geraden abbildet und umgekehrt nicht-vertikale Geraden auf Punkte:

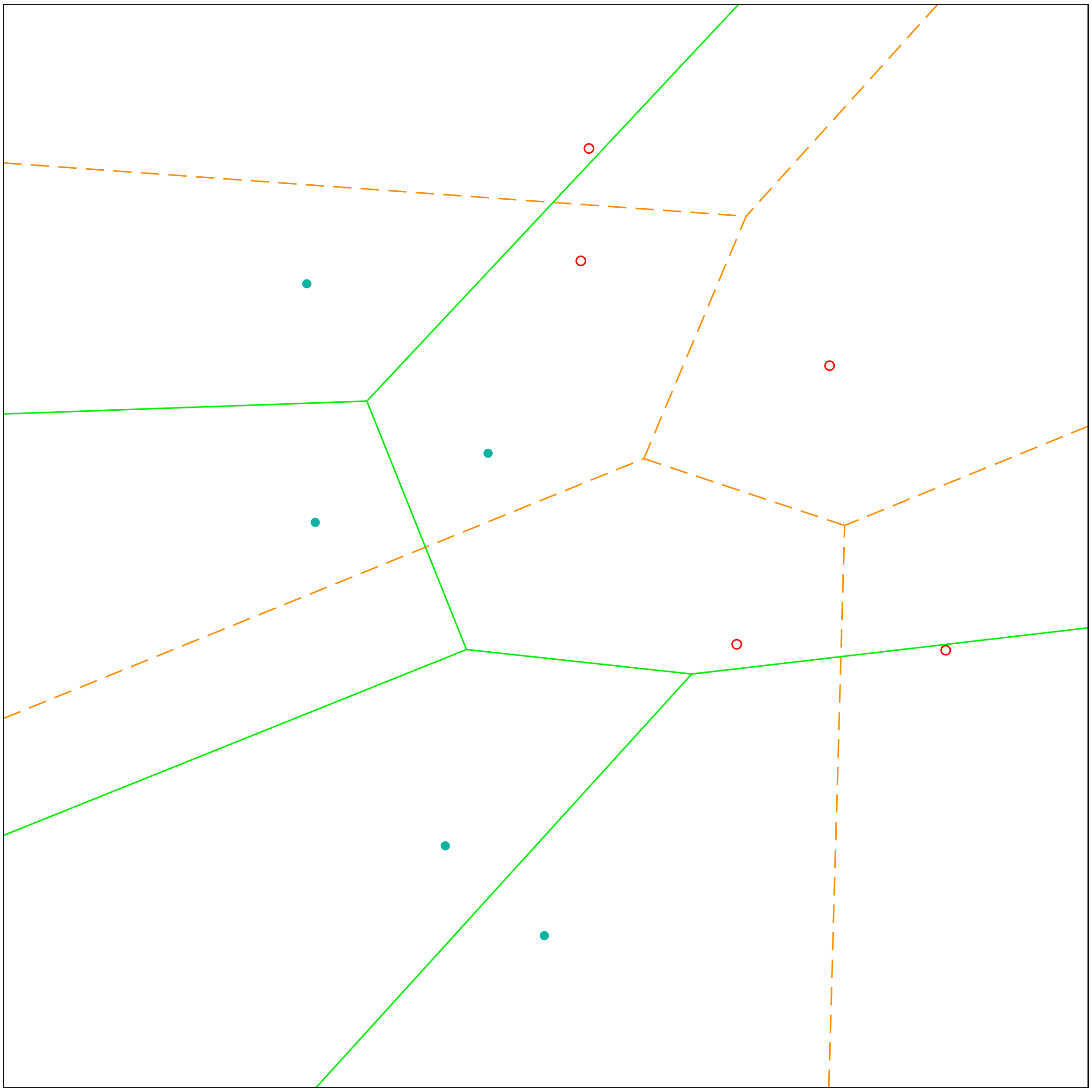
$$p = (p_x, p_y) \mapsto \mathscr{D}(p) : Y = -p_x \cdot X + p_y$$
$$\ell : Y = aX + b \mapsto \mathscr{D}(\ell) = (a, b)$$

Welche der beiden rechts abgebildeten Konfigurationen von fünf Punkten ist unter dieser Dualität dual zu der links abgebildeten Konfiguration von fünf Geraden. Die beiden links abgebildeten Punkte mit gleicher x-Koordinate (und die dazu dualen Geraden rechts) dienen der Orientierung. Begründen Sie ihre Antwort.

Aufgabe 2: (8 Punkte)

In der folgenden Abbildung sind die Teile der Voronoidiagramme einer Punktmenge L und einer Punktmenge R (gestrichelt) innerhalb eines Quadrates dargestellt. Die Punkte in L liegen links der Vertikalen, die das Quadrat halbiert, die Punkte R rechts davon. Skizzieren Sie den ungefähren Verlauf des Bisektors $\mathcal{B}(L,R)$!



Aufgabe 3: (7 Punkte)

(a) Ein zusammenhängender eingebetteter planarer Graph mit n Knoten, e Kanten und f Flächen erfüllt die Euler-Formel

$$n - e + f = 2$$

Beweisen Sie, dass ein (zusammenhängender) einfacher planarer Graph mit n Knoten maximal $3n - 6$ Kanten besitzt.

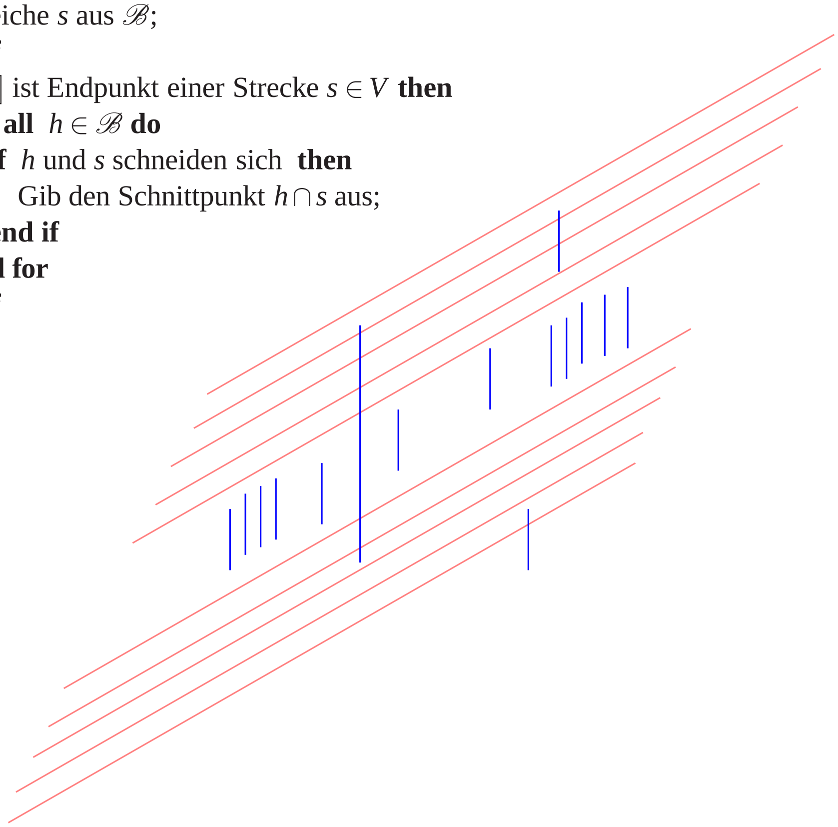
(b) Der Grad eines Punktes p in einer Triangulation einer Punktmenge ist die Anzahl der Kanten, die in p enden. Zeigen Sie, dass es in jeder Delaunaytriangulation einen Knoten vom Grad höchstens 5 gibt.

Aufgabe 4: (9 Punkte)
Sei V eine Menge von n vertikalen Strecken und sei N eine Menge von n nichtvertikalen Strecken. Die Strecken in N seien paarweise disjunkt, d.h., je zwei Strecken in N haben keinen Punkt gemeinsam. Ferner seien die Strecken in $V \cup N$ in allgemeiner Lage, d.h, die x-Koordinaten der Endpunkte der Strecken in N und die der unteren Endpunkte der Strecken in V seien jeweils paarweise verschieden.

Im Folgenden sind drei verschiedene Algorithmen angegeben, die jeweils alle Schnittpunkte zwischen den Strecken in V und N berechnen. Sei k die Anzahl der Schnittpunkte. Bestimmen Sie für jeden Algorithmus dessen asymptotische Laufzeit *im schlechtesten Fall* in Abhängigkeit von k und n.

Algorithmus **A**:

Speichere die beiden Endpunkte der Strecken in N und die unteren Punkte der Strecken in V nach x-Koordinaten sortiert in einem Feld $A[1..2|N|+|V|]$;
$\mathscr{B}$ = leerer BBSB, in dem später die Strecken aus N nach y-Koordinaten sortiert gespeichert werden;
for $i = 1$ to $2|N|+|V|$ **do**
  if $A[i]$ ist linker Endpunkt einer Strecke $s \in N$ **then**
    füge s in $\mathscr{B}$ ein;
  end if
  if $A[i]$ ist rechter Endpunkt einer Strecke $s \in N$ **then**
    streiche s aus $\mathscr{B}$;
  end if
  if $A[i]$ ist Endpunkt einer Strecke $s \in V$ **then**
    for all $h \in \mathscr{B}$ **do**
      if h und s schneiden sich **then**
        Gib den Schnittpunkt $h \cap s$ aus;
      end if
    end for
  end if
end for



Algorithmus **B**:

Speichere die beiden Endpunkte der Strecken in N und die unteren Punkte der Strecken in V nach x-Koordinaten sortiert in einem Feld $A[1..2|N|+|V|]$;
$\mathscr{B}$ = leerer BBSB, in dem später die Strecken aus N nach y-Koordinaten sortiert gespeichert werden;
for $i = 1$ to $2|N|+|V|$ **do**
  if $A[i]$ ist linker Endpunkt einer Strecke $s \in N$ **then**
    füge s in $\mathscr{B}$ ein;

end if
if $A[i]$ ist rechter Endpunkt einer Strecke $s \in N$ **then**
  streiche s aus $\mathscr{B}$;
end if
if $A[i]$ ist Endpunkt einer Strecke $s \in V$ **then**
  Sei $A[i] = (a,b)$ und sei (a,c) der obere Endpunkt von s;
  Bestimme die Strecke h in $\mathscr{B}$, deren y-Koordinate für $x = a$ minimal ist unter den Strecken in $\mathscr{B}$, deren y-Koordinate mindestens so groß ist wie b;
  d = y-Koordinate von h für $x = a$;
  while $d \leq c$ **do**
    Gib den Schnittpunkt $h \cap s$ aus;
    h = Nachfolger von h in $\mathscr{B}$;
    d = y-Koordinate von h für $x = a$;
  end while
end if
end for

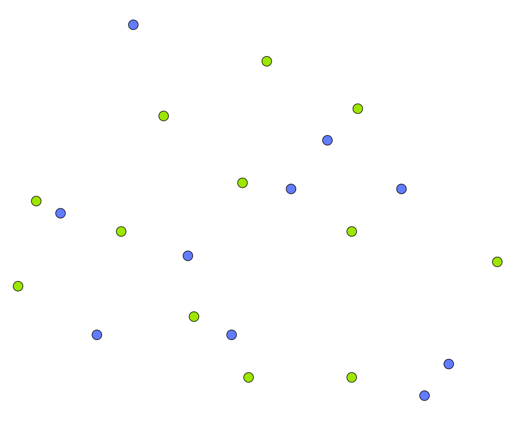
Algorithmus **C**:

Bestimme eine Rotation φ, so dass $\varphi(N \cup V)$ keine vertikalen Strecken enthält;
$N' = \varphi(N)$;
$V' = \varphi(V)$;
$S = \mathit{BentleyOttmannPlaneSweep}(N' \cup V')$;
Gib $\varphi^{-1}(S)$ aus;

Hinweise

- Nehmen Sie an, dass eine geeignete Rotation in **C** in Zeit $O(n \log n)$ gefunden werden kann.
- Der Algorithmus $\mathit{BentleyOttmannPlaneSweep}(L)$ ist der aus der Vorlesung bekannte Plane-Sweep-Algorithmus für den allgemeinen Fall ohne vertikale Strecken.
- Ein BBSB ist ein balancierter binärer Suchbaum, in dem man in Zeit $O(1)$ Vorgänger bzw. Nachfolger eines bereits lokalisierten Elementes finden kann, also beispielsweise ein entsprechend erweiterter AVL-Baum.

Aufgabe 5: (11 Punkte)
Gegeben seien eine Menge B von n blauen Punkten und eine Menge G von n grünen Punkten in der Ebene in allgemeiner Lage, d.h., mit paarweise verschiedenen x-Koordinaten und paarweise verschiedenen y-Koordinaten. Ein Punkt p liegt *westlich* von einem Punkt q, falls $p_x < q_x$ und *südwestlich* von q, falls $p_x < q_x$ und $p_y < q_y$.



(a) Geben Sie einen asymptotisch möglichst effizienten Algorithmus an, der alle blauen Punkte bestimmt, für die es höchstens einen grünen Punkt gibt, der südwestlich des blauen Punktes liegt. Welche asymptotische Laufzeit erreicht ihr Verfahren?

(b) Geben Sie einen asymptotisch möglichst effizienten Algorithmus an, der alle blauen Punkte bestimmt, für die es höchstens einen grünen Punkt gibt, der westlich des blauen Punktes liegt. Welche asymptotische Laufzeit erreicht ihr Verfahren?

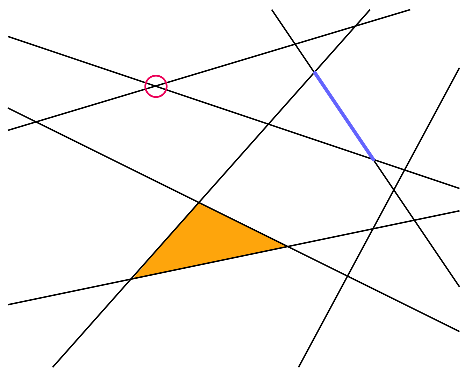
Zur Korrektheit der Algorithmen brauchen Sie ausnahmsweise nichts zu sagen. ☺

Aufgabe 6: (9 Punkte)
Sei L eine Menge von n Geraden in allgemeiner Lage in der Ebene. Ferner bezeichne $\mathcal{A}(L)$ das Arrangement der Geraden in L.

(a) Aus wie vielen Knoten besteht $\mathcal{A}(L)$? (ohne Beweis)

(b) Aus wie vielen Strecken und Strahlen besteht $\mathcal{A}(L)$? (ohne Beweis)

(c) Aus wie vielen Zellen besteht $\mathcal{A}(L)$? (ohne Beweis)



(d) Wir konstruieren $\mathcal{A}(L)$ randomisiert inkrementell: Zunächst permutieren wir die Eingabefolge der Geraden zufällig und konstruieren $\mathcal{A}(L)$, in dem wir die Geraden in der durch die Permutation bestimmten Reihenfolge eine nach der anderen zum bisher konstruierten Arrangement hinzufügen. Alle Permutationen seien gleichwahrscheinlich. Wir betrachten einen beliebigen, aber festen Knoten v des Arrangements $\mathcal{A}(L)$. Wie wahrscheinlich ist es, dass v erst beim Hinzufügen der letzten Geraden entsteht? Begründen Sie ihre Antwort.