Schriftliche Prüfung im Fach

Grundzüge der Algorithmischen Geometrie

2. Februar 2012, 9:00 - 11:00 Uhr

Name, Vorname: ____________________

Matrikelnummer: ____________________

Anzahl Doppelbögen: ____________________

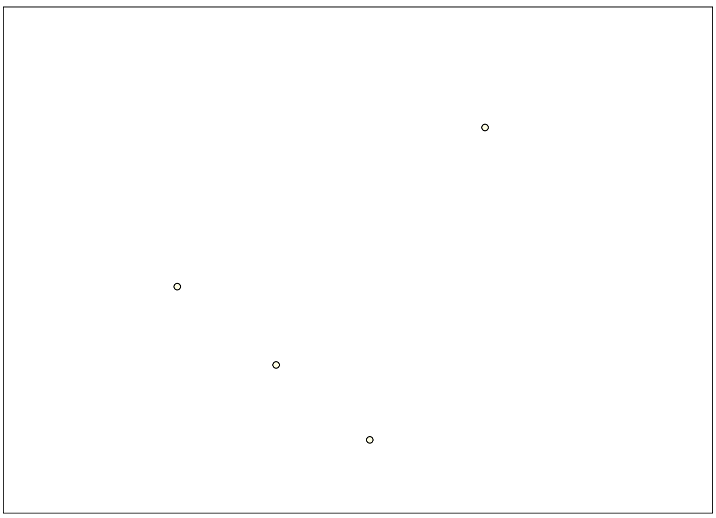
Bearbeitungszeit:	120 Min.
Zugelassene Hilfsmittel:	Keine!
Gesamtzahl Aufgaben:	6
Gesamtpunktzahl:	46

Bitte jeden beschriebenen Doppelbogen mit Matrikelnummer und Namen beschriften! Viel Erfolg!

1	2	3	4	5	6

Aufgabe 1 (6 PUNKTE)

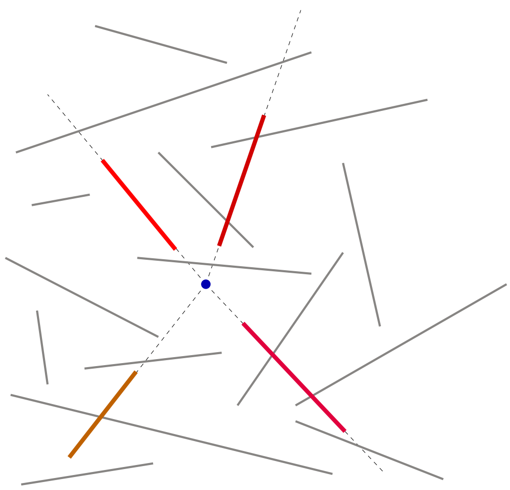
Skizzieren Sie das Voronoidiagramm der vier Punkte in der folgenden Abbildung innerhalb des umschließenden Rechtecks:



Aufgabe 2 (10 PUNKTE)

Sei $\mathscr{S} = \{s_1, s_2, \ldots, s_n\}$ eine Menge von n paarweise disjunkten Strecken in der Ebene. Ferner sei $\mathscr{R} = \{r_1, r_2, \ldots, r_m\}$ eine Menge von m paarweise disjunkten Strecken, deren unterstützende Geraden $\ell(r_i)$ sich alle in einem Punkt p schneiden. Sie dürfen annehmen, dass p in keiner der Strecken in $\mathscr{R}$ enthalten ist.

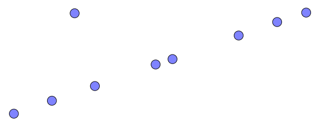
Geben Sie einen möglichst effizienten Plane-Sweep Algorithmus an, bei dem (konzeptuell) eine Gerade durch p um p rotiert wird und alle k Schnittpunkte zwischen Strecken in $\mathscr{S}$ und Strecken in $\mathscr{R}$ berechnet werden. Analysieren Sie Ihren Algorithmus. Ihr Algorithmus sollte Laufzeit in $O((n+m)\log(n+m)+k)$ erreichen. Zur Korrektheit des Algorithmus brauchen Sie nichts zu sagen.



Aufgabe 3 (5 PUNKTE)
Sei $\mathscr{D}$ die Dualitätsabbildung, die Punkte wie folgt auf nicht-vertikale Geraden abbildet und umgekehrt nicht-vertikale Geraden auf Punkte:

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

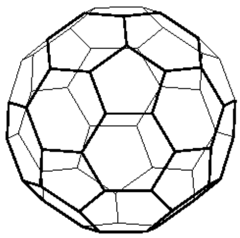
Sei P eine Menge von k Punkten mit paarweise verschiedenen x-Koordinaten, die alle bis auf einen auf einer Geraden liegen. Ferner sei $L = \{\mathscr{D}(p) \mid p \in P\}$ die Menge der zu den Punkten in P dualen Geraden und $\mathcal{A}(L)$ das Arrangement der Geraden in L.



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

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

Aufgabe 4 (6 PUNKTE)



Sei $P \subset \mathbb{R}^3$ eine Menge von $n \geq 4$ Punkten im dreidimensionalen Raum. Sei $CH(P)$ die konvexe Hülle von P, also das kleinste konvexe Polytop, das alle Punkte in P enthält. Alle Punkte in P seien Extrempunkte.

(a) Aus wie vielen Kanten besteht die konvexe Hülle $CH(P)$ maximal? (ohne Beweis)

(b) Zeigen Sie dann, dass die konvexe Hülle $CH(P)$ eine Seitenfläche enthalten muss, die von höchstens 5 Kanten begrenzt wird.

Aufgabe 5 (10 PUNKTE)
Welche der folgenden Behauptungen sind wahr, welche falsch (jeweils ohne Beweis!)?
2 Punkte für jede richtige Antwort, 1 Punkt Abzug für jede falsche.

wahr	falsch	
○	○	Gegeben sei eine Menge S von n Strecken. Man kann in Zeit $O(n \log n)$ testen, ob die Strecken in S paarweise disjunkt sind.
○	○	Man kann die Delaunay-Triangulation einer Menge von n Punkten in der Ebene nur dann in Zeit $O(n \log n)$ bestimmen, wenn die Punkte bereits nach x-Koordinate sortiert vorliegen.
○	○	Wenn die n Punkte einer Punktmenge bereits nach x-Koordinate sortiert vorliegen, so kann man mit einem auf Plane-Sweep und Rotating-Calipers basierenden Algorithmus ein Punktepaar mit maximalem Abstand in linearer Zeit finden.
○	○	Sei P eine Menge von n Punkten in allgemeiner Lage in der Ebene. Dann gehören alle Kanten des Gabriel Graphen von P zur Kantenmenge der Delaunay-Triangulation von P.
○	○	Ein zweidimensionales Lineares Programm mit n Nebenbedingungen kann man mit einem randomisierten Algorithmus in erwarteter Laufzeit in $O(n)$ lösen.

Aufgabe 6 (9 PUNKTE)
Sei P eine Menge von n Punkten in der Ebene. Geben Sie einen Teile-und-Herrsche Algorithmus an, der die maximalen Punkte in P aufsteigend nach x-Koordinate sortiert in Zeit $O(n \log n)$ berechnet. Zur Erinnerung: Ein Punkt $p \in P$ heißt maximal, falls $\{q \in P - \{p\} \mid q_x \geq p_x \text{ und } q_y \geq p_y\} = \emptyset$.
Begründen Sie die Korrektheit Ihres Algorithmus. Ihr Algorithmus muss die geforderte Laufzeit erzielen, Sie müssen dies aber nicht nachweisen!