Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Sommersemester 2025

Dr. Arne Schmidt

Klausur
Einführung in Algorithmische Geometrie
31.07.2025

Nachname: ....

Vorname: ....

Matr.-Nr.: ....

Studiengang: ....

☐ Bachelor ☐ Master ☐ Andere: ....

Hinweise:

- Bitte das Deckblatt in Druckschrift vollständig ausfüllen.
- Die Klausur besteht aus 13 Blättern, bitte auf Vollständigkeit überprüfen.
- Die Bearbeitungszeit für die Klausur beträgt 120 Minuten.

- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- Antworten, die *nicht* gewertet werden sollen, bitte deutlich durchstreichen.
- Kein Tippex verwenden.
- Werden mehrere Antworten gegeben, werten wir die mit der geringsten Punktzahl.
- Mit *Bleistift* oder in *Rot* geschriebene Klausurteile können nicht gewertet werden.

Aufgabe	1	2	3	4	5	6	7	Σ
Max	10	15	22	12	10	21	10	100
Erreicht								

Aufgabe 1: Kurzfragen **(5*2 Punkte)**

a) Seien f eine Fläche und e eine Kante in einer DCEL. Welche Aussagen sind korrekt?

- ☐ IncidentFace(OuterBoundary(f)) = f
- ☐ OuterBoundary(IncidentFace(e)) = e
- ☐ InnerBoundary(IncidentFace(e)) = e

b) Welche Aussagen zu einem kd-Tree auf einer k-dimensionalen Punktmenge sind korrekt?

- ☐ Die Höhe des Baumes in $O(\log n)$.
- ☐ Der Speicherbedarf ist in $O(n)$.
- ☐ Der kd-Tree kann in Zeit $O(n \log n)$ erstellt werden.

c) Betrachte die Triangulation $\mathcal{T}(\mathcal{A})$ für ein Arrangement $\mathcal{A}$. Welche Aussagen sind korrekt?

- ☐ Jede Fläche in $\mathcal{T}(\mathcal{A})$ (auch die unbegrenzte Fläche) berührt exakt 3 Knoten.
- ☐ Jeder Knotengrad in $\mathcal{T}(\mathcal{A})$ ist mindestens 3.
- ☐ Man kann immer 3 Knoten in $\mathcal{T}(\mathcal{A})$ auswählen, die untereinander nicht adjazent sind.

d) Seien P_1 und P_2 zwei Polygone mit n_1 bzw. n_2 vielen Knoten. Welche Aussagen zu Minkowski-Summen sind korrekt?

- ☐ $P_1 \bigoplus P_2$ kann in Zeit $O(n_1 + n_2)$ bestimmt werden.
- ☐ Sind P_1 und P_2 konvex, dann ist auch $P_1 \bigoplus P_2$ konvex.
- ☐ Sind P_1 und P_2 Dreiecke, dann ist auch $P_1 \bigoplus P_2$ ein Dreieck.

e) Welche Aussagen zum Art-Gallery-Problem sind korrekt?

- ☐ Die Kardinalität eines größten unabhängigen Witness-Sets entspricht der Kardinalität eines kleinsten Guard-Sets
- ☐ $\lfloor \frac{n}{3} \rfloor$ Vertex-Guards sind manchmal notwendig und immer ausreichend.
- ☐ Decken Guards den Rand eines Polygons ab, so ist das gesamte Polygon abgedeckt.

Aufgabe 2: Range Queries **(5+4+6 Punkte)**

a) Betrachte die Punktmenge P in Abbildung 1. Zeichne den kd-Tree zu P, indem die Trennlinien in die Abbildung gezeichnet werden und die Graphenstruktur angegeben wird.

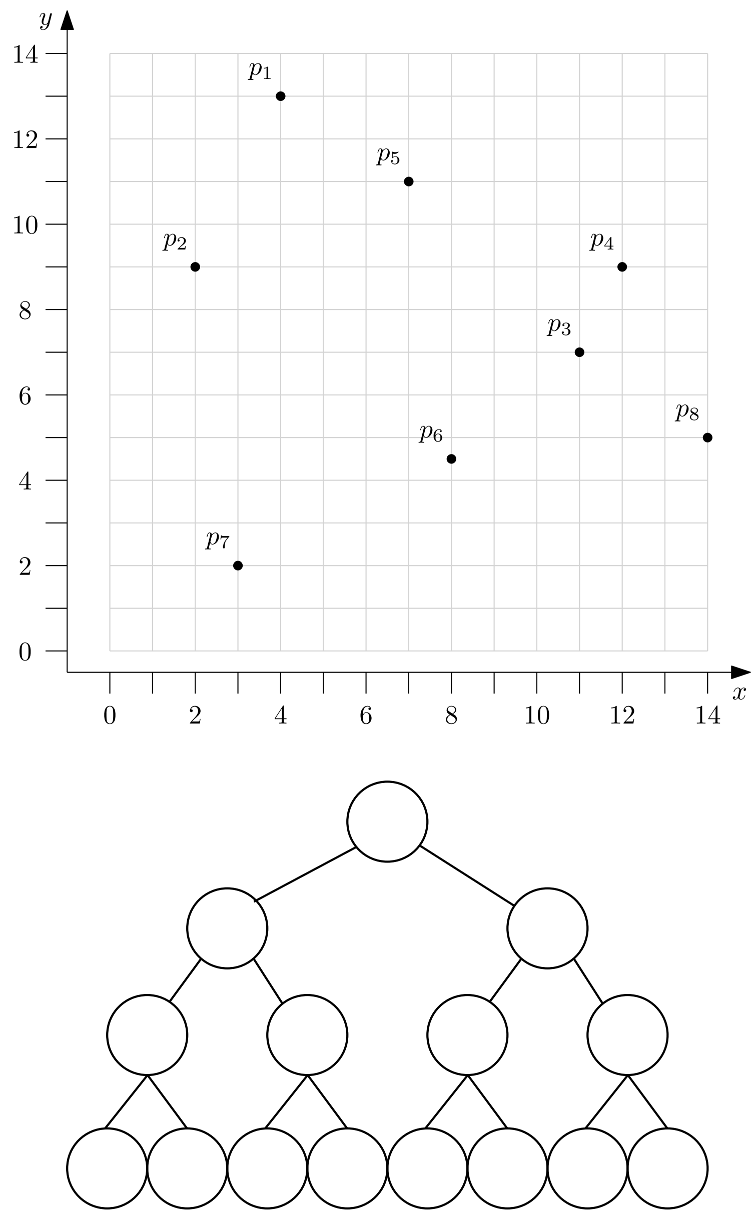


Abbildung 1: Oben: Punktmenge P. Unten: Dazugehöriger kd-Tree.

b) Führe eine Range-Query mit Rechteck $R = [0, 10] \times [0, 10]$ und dem kd-Tree aus Teil a) aus. Liste alle in R enthaltenen Punkte auf und beschrifte die abgelaufenen Knoten folgendermaßen:

(C) Der Bereich vom Knoten ist vollständig in R enthalten.

(S) Der Bereich vom Knoten schneidet R (aber nicht komplett enthalten).

(X) Der Bereich vom Knoten schneidet R nicht.

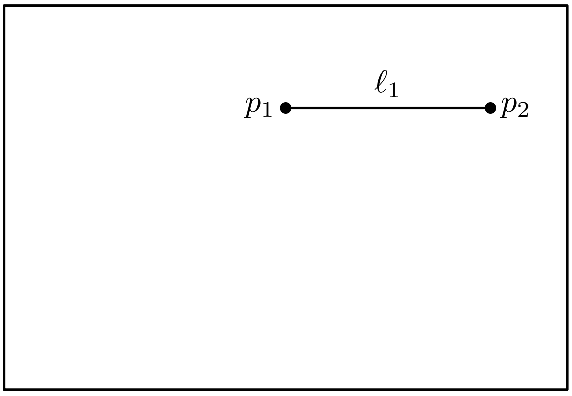
In R enthaltene Punkte:

c) Angenommen, wir führen mit einem kd-Tree eine Range-Query mit einem Dreieck mit zwei achsenparallelen Seiten anstatt einem Rechteck aus.

   Zeige: Die Range-Query mit Dreiecken benötigt im Worst-Case $\Omega(n)$ Zeit, selbst wenn keine Punkte im Dreieck liegen und die Punkte in allgemeiner Lage sind.

Aufgabe 3: Lokalisierung **(7+3+6+6 Punkte)**

a) Betrachte die Menge an drei Segmenten S in Abbildung 2 (unten). Skizziere die Trapezoidal Map in die Abbildung zu jedem neu eingefügten Segment und gib die zugehörige Suchstruktur $\mathcal{D}(S)$ an.



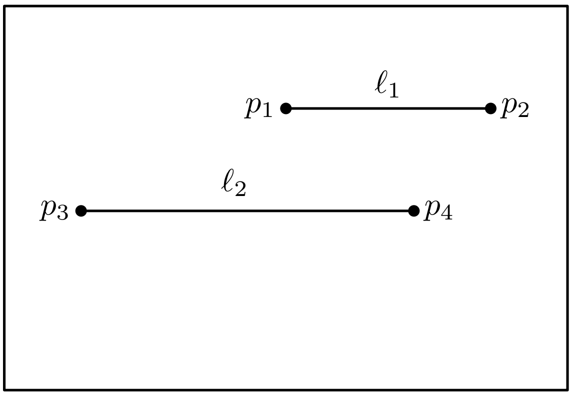


Abbildung 2: Links: Inkrementell eingefügte Kanten. Rechts: DAG der Trapezoidal Map.

b) Zeichne einen Punkt z in die Abbildung 2 mit den zwei Segmenten, sodass der Suchpfad bzgl. z dem längsten Suchpfad in $\mathcal{D}(S)$ entspricht. Gib außerdem den Suchpfad bzgl. z an.

c) Sei S eine Menge von n nicht-schneidenden Segmenten in allgemeiner Lage. Argumentiere mit Hilfe eines Plane-Sweeps, dass es höchstens $3n + 1$ viele Trapeze in der Trapezoidal Map geben kann.

(Hinweis: Es darf angenommen werden, dass die Segmente keinen gemeinsamen Endpunkt haben.)

d) Sei $P = (p_1, \ldots, p_n)$ ein konvexes Polygon.

Zeige: Das Entscheidungsproblem, ob ein gegebener Punkt $q \in \mathbb{R}^2$ innerhalb von P liegt, kann in Zeit $O(\log n)$ Zeit gelöst werden.

Aufgabe 4: Motion Planning **(6+6 Punkte)**

a) Betrachte den Work-Space mit Hindernissen $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ in Abbildung 3. Darin soll ein entlang des Vektors (x, y) verschobener Roboter $\mathcal{R}(x, y)$ bewegt werden, wobei $\mathcal{R}(0, 0)$ das Polygon mit Knoten $(1, 0), (0, 1), (-1, 0), (0, -1)$ ist. Markiere den freien Configuration-Space in Abbildung 3.

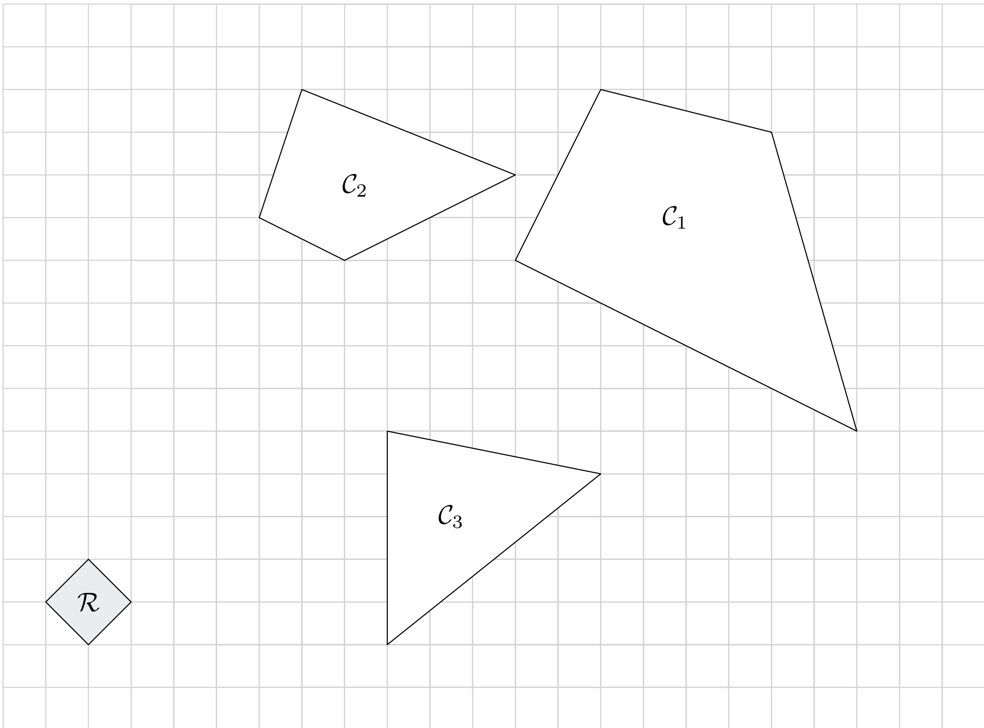


Abbildung 3

b) Ein Punktroboter soll sich im Work-Space aus Abbildung 4 bewegen. Zeichne die Roadmap in die Abbildung ein und markiere darauf einen Pfad vom Startpunkt s zum Zielpunkt t.

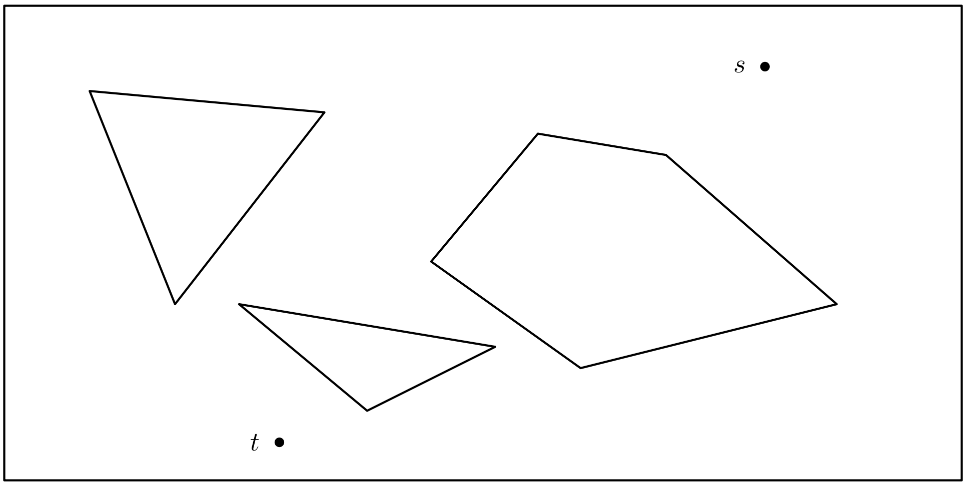


Abbildung 4

Aufgabe 5: Polyederformel **(4+6 Punkte)**

Sei A ein Arrangement mit n Knoten, m Kanten und f Flächen. Aus der Vorlesung kennen wir die *Polyederformel*:

$$n - m + f \geq 2$$

a) Zeige: In A gilt $f \leq 2n - 4$.

b) Zeige: Wenn jeder Kreis von Kanten in A gerade Länge besitzt, dann existiert in A einen Knoten vom Grad höchstens 3.

Aufgabe 6: Art Gallery Problem **(3+4+6+8 Punkte)**

a) Skizziere das Sichbarkeitspolygon des Punktes p in Abbildung 5.

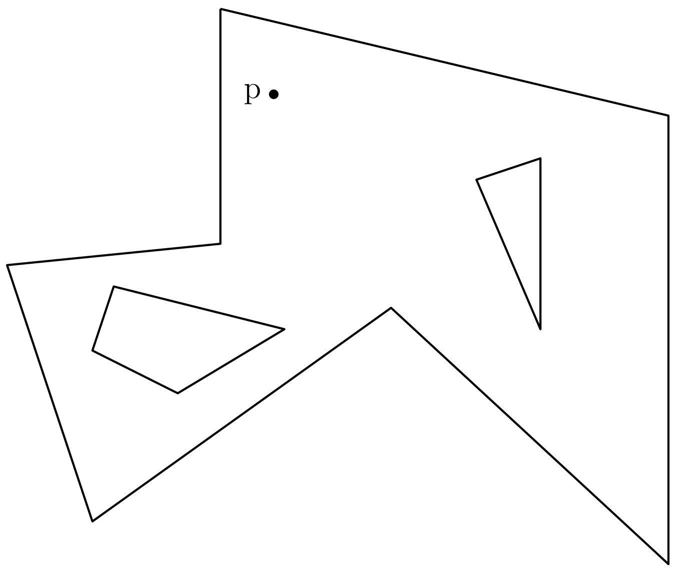


Abbildung 5

b) Betrachte das Polygon in Abbildung 6. Zeichne dort ein kleinstmögliches Point-Guard Set (als Punkte) ein. Füge zusätzlich ein unabhängiges Witness Set (als Kreuze) ein, um zu zeigen, dass dein Guard Set tatsächlich minimal ist.

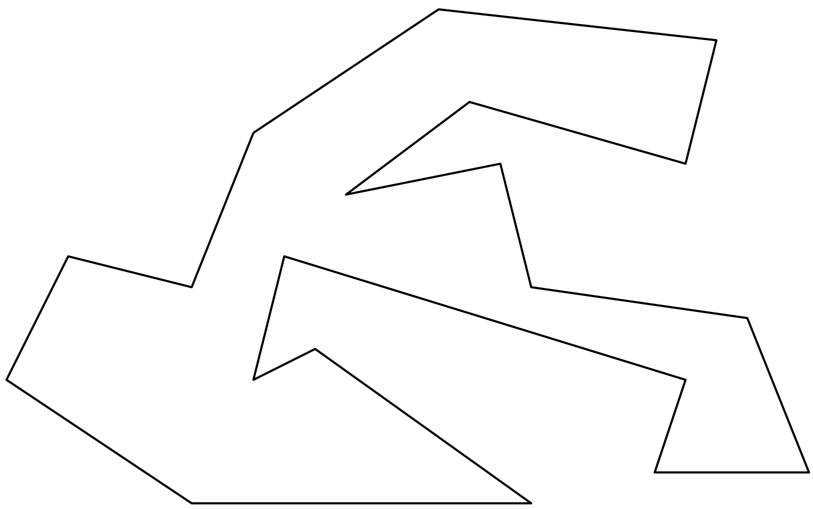


Abbildung 6

c) Für ein gegebenes Polygon P bezeichne $\mathrm{g_p}(P)$ (bzw. $\mathrm{g_v}(P)$) die kleinste Anzahl von Point-Guards (bzw. Vertex-Guards), mit der P vollständig überwacht werden kann.

Zeige: Für jedes $c \in \mathbb{N}$ gibt es ein Polygon P, sodass $c \leq \frac{\mathrm{g_v}(P)}{\mathrm{g_p}(P)}$ gilt, d.h. manchmal sind deutlich mehr Vertex-Guards als Point-Guards erforderlich.

d) Ein *Polyomino* der Größe n ist eine zusammenhängende Fläche, die aus n kantenweise verbundenen Einheitsquadraten besteht (siehe Abbildung 7). Wir referenzieren die Einheitsquadrate anhand ihres jeweiligen Mittelpunktes (x, y). Ohne Beschränkung der Allgemeinheit sei $(x, y) \in \mathbb{N}^2$. Ein auf (x, y) platzierter *Turm-Guard* sieht intuitiv alle Quadrate in vertikaler und horizontaler Richtung, bis er an die Grenze des Polyominos gerät. Genauer gesagt sind das die Quadrate in

$$\begin{aligned}\mathrm{Vis}(x, y) := \{(x + c, y) \mid 0 \leq c \leq c_1\} &\cup \{(x - c, y) \mid 0 \leq c \leq c_2\} \\ \cup \{(x, y + c) \mid 0 \leq c \leq c_3\} &\cup \{(x, y - c) \mid 0 \leq c \leq c_4\},\end{aligned}$$

wobei c_1, c_2, c_3, c_4 jeweils größtmöglich gewählt sind, sodass jedes Quadrat in $\mathrm{Vis}(x, y)$ im Polyomino enthalten ist. In Abbildung 7 ist ein beispielhafter Turm-Guard mit Sichtbarkeitsbereich dargestellt. Ein Polyomino ist *vollständig überwacht*, wenn jedes Quadrat von mindestens einem Turm-Guard gesehen wird.

Zeige für beliebiges $n \in \mathbb{N}$: Jedes Polyomino der Größe n kann mit $\lfloor \frac{n}{2} \rfloor$ Turm-Guards vollständig überwacht werden. Zusätzlich gibt es ein Polyomino der Größe n, in dem mindestens $\lfloor \frac{n}{2} \rfloor$ Turm-Guards erforderlich sind.

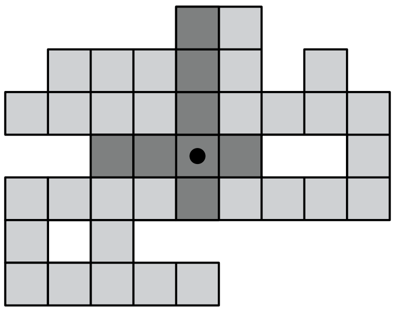


Abbildung 7: Ein Polyomino mit zwei Löchern. Der Sichtbarkeitsbereich des Turm-Guards, dargestellt als schwarzer Punkt, ist dunkel-grau markiert.

Aufgabe 7: Sweep Line **(10 Punkte)**

Betrachte das folgende Skyline-Problem:

Gegeben: n Rechtecke der Form $R_i = (x_\ell, x_r, h)$ mit Start x_ℓ, Ende x_r und Höhe h

Gesucht: Eine Menge von Tupeln (X_j, H_j), an welchen Punkten X_j sich die Höhe des höchsten Rechtecks auf $H_j \geq 0$ ändert.

Als Beispiel betrachte die Instanz aus Abbildung 8 mit den Rechtecken $R_1 = (0, 2, 7)$, $R_2 = (1, 5, 5)$, $R_3 = (4, 10, 4)$, $R_4 = (6, 9, 7)$ und $R_5 = (11, 13, 7)$. Die Skyline lässt sich dann beschreiben durch die Tupel $(0, 7), (2, 5), (5, 4), (6, 7), (9, 4), (10, 0), (11, 7), (13, 0)$.

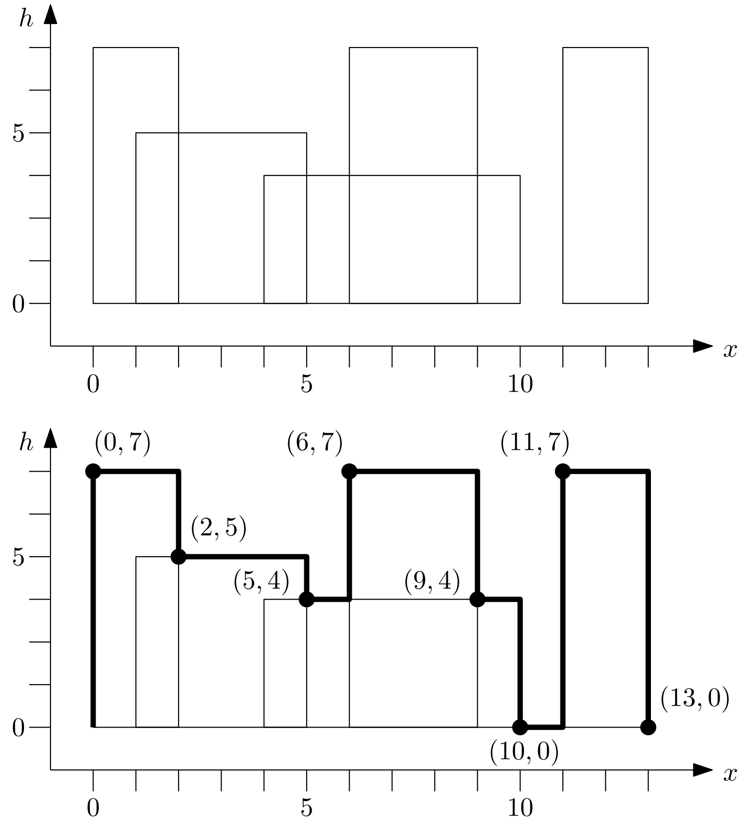


Abbildung 8: Oben: Menge von Rechtecken. Unten: Skyline (fette Linie) mit den gesuchten Punkten.

Gib einen Sweep-Line-Algorithmus an, der dieses Problem in $O(n \log n)$ Zeit löst. (Hinweis: Achte darauf, dass zwei Rechtecke gemeinsame Start- bzw. Endpunkte besitzen können.)

Viel Erfolg ☺