

Algorithmische Geometrie

Zusammenfassung



Grundsätzliche Hinweise

Ziel der Prüfung

- wir müssen euch bescheinigen, dass
 - ihr den Stoff der Vorlesung kennt
 - die Inhalte verstanden habt
 - die gelernten Techniken zu benutzen wisst
- mündliche Prüfung: besonders gut geeignet für die ersten beiden Punkte
- der Fokus wird auf dem zweiten Punkt liegen

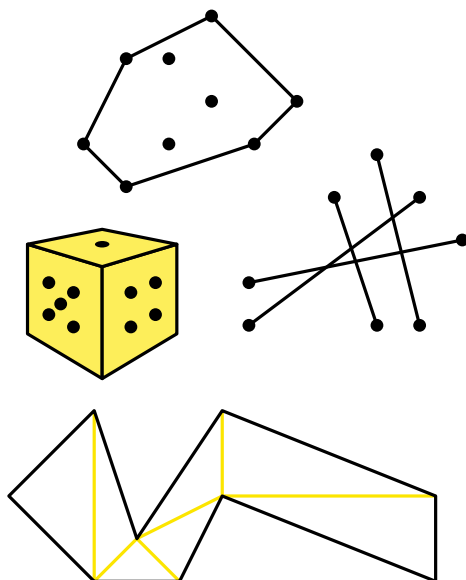
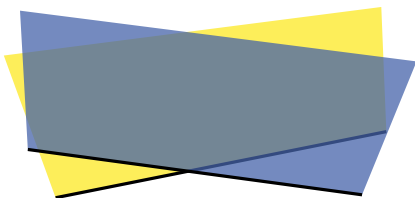
Vorbereitung

- Schritt 1: rekapituliert und versteht die Inhalte
- Schritt 2: erklärt jemandem die Inhalte
- **Achtung:** Schritt 2 ist genau das, was in der Prüfung passiert. Das solltet ihr definitiv vorher üben!
- die Fragen auf den folgenden Folien können da als Startpunkt dienen

Überblick

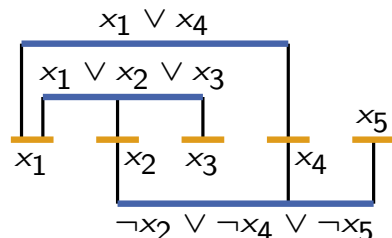
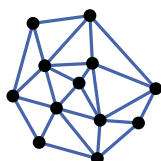
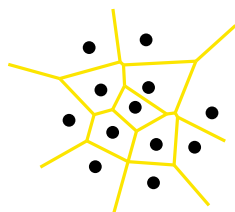
Basic Toolbox

- konvexe Hülle
- Linienschnitt
- Triangulierung
- Ebenenschnitt



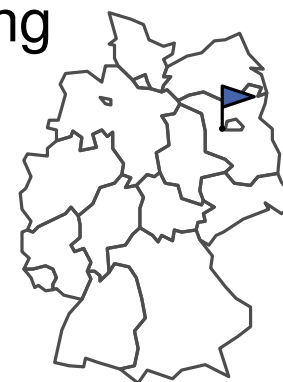
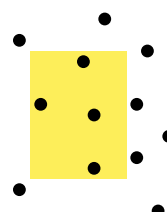
Erweiterte Toolbox

- Voronoi-Diagramme
- Delaunay-Triangulierung
- untere Schranken



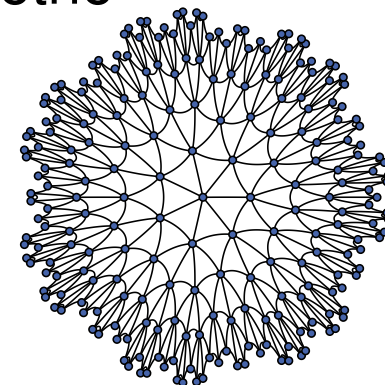
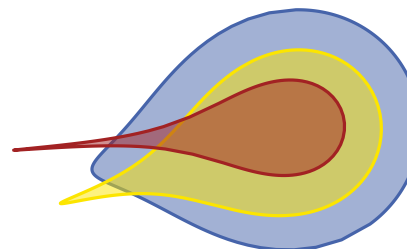
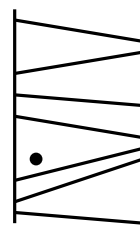
Geometrische Datenstrukturen

- orthogonal range searching
- persistente Datenstrukturen
- Punkt-Lokalisierung



Verwandte Themen

- Berechnungsmodelle
- hyperbolische Geometrie
- Origami



Konvexe Hülle

Problemstellung

- Wann ist eine Menge Konvex?
- Welche verschiedenen Sichtweisen gibt es auf Konvexität?
- Welches Problem haben wir betrachtet?

Algorithmen: Graham Scan/Gift Wrapping

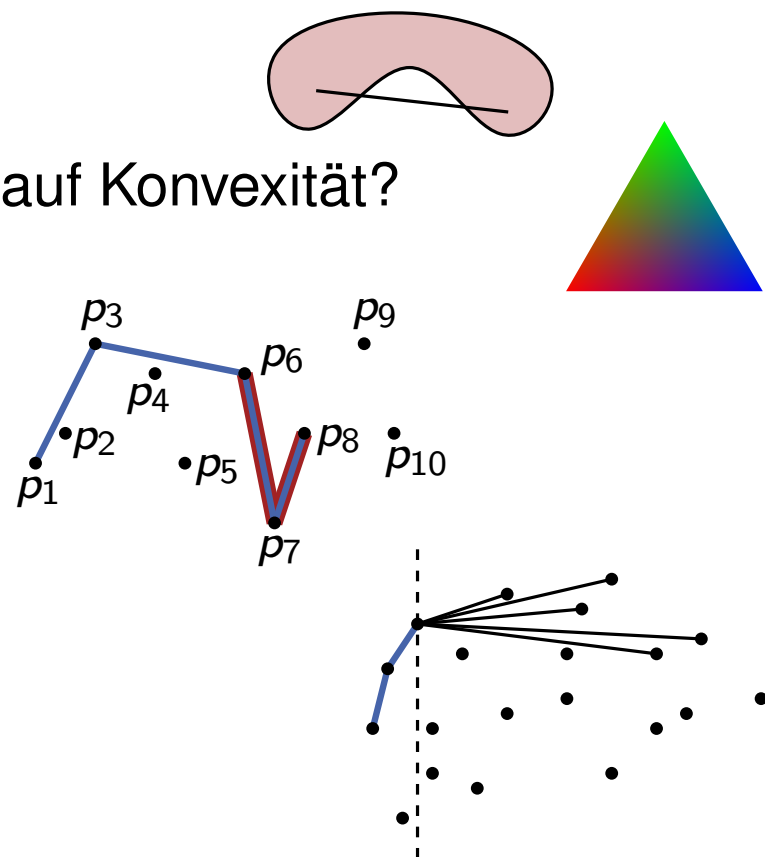
- Wie funktioniert der Algorithmus?
- Warum ist er korrekt?
- Welche Laufzeit hat er? Warum?
- Ist er robust?

Untere Schranke

- Geht es schneller? Warum nicht?

Konvexe Hülle eines einfachen Polygons

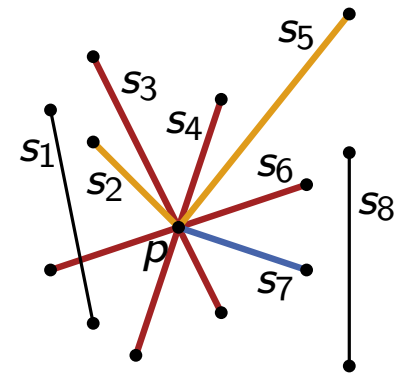
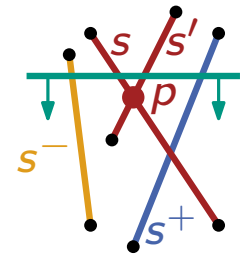
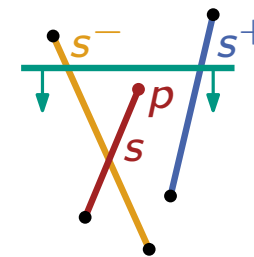
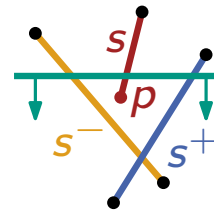
- Geht es schneller?
- Warum? Wie nutzt man das gegebene Polygon aus?



Streckenschnitt & Sweep-Line

Schnittpunkte zwischen Strecken

- Was ist das grundsätzliche Vorgehen?
- Welche Sonderfälle gibt es?
- Wie kann man mit diesen umgehen?
- insbesondere: Wie kann man mehrere Ereignisse bei einem einzelnen Punkt behandeln?



Laufzeit

- Welche Laufzeit erhält man?
- Wie zeigt man diese Laufzeit?

Speicherverbrauch

- Genügt linear viel Speicher? Wie?
- Braucht man ggf. $\Omega(n^2)$ Speicher, wenn man nicht aufpasst?

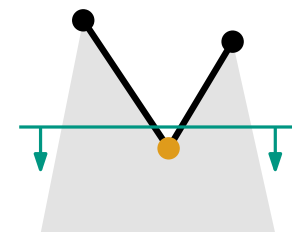
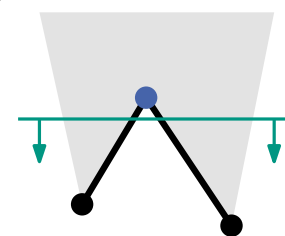
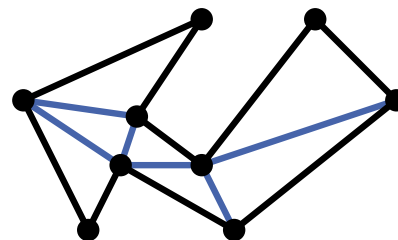
Doppelt-verkettete Kantenliste

- Was ist das?
- Wozu ist das da?
- Was hat das mit Linienschnitt zu tun?

Triangulierung von Polygonen

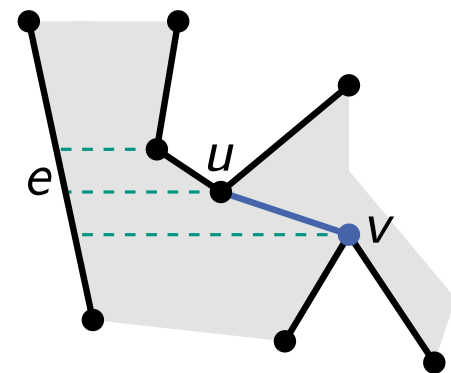
Grundlagen

- Was ist die grundlegende Idee?
- Was bedeutet y -monoton?
- Was sind Split- und Merge-Knoten und was haben die mit y -Monotonie zu tun?



Eliminierung von Split-Knoten

- Wie wird man einen Split-Knoten los?
- Was hat es mit dem Helfer auf sich?
- Warum sind die eingefügten Kanten kreuzungsfrei?
- Wie führt das zu einem Algorithmus?
- Welche Event-Typen gibt es? Wie behandelt man sie?
- Welche Laufzeit erhalten wir?
- Überträgt sich das auf Merge-Knoten?



Triangulierung

- Was ist die Idee bei der Triangulierung y -monotoner Polygone?

Lineare Programme

Gussformen

- Was ist das Gussform-Problem?
- Warum lässt sich das mittels 2D-LP lösen?

Lineare Programme

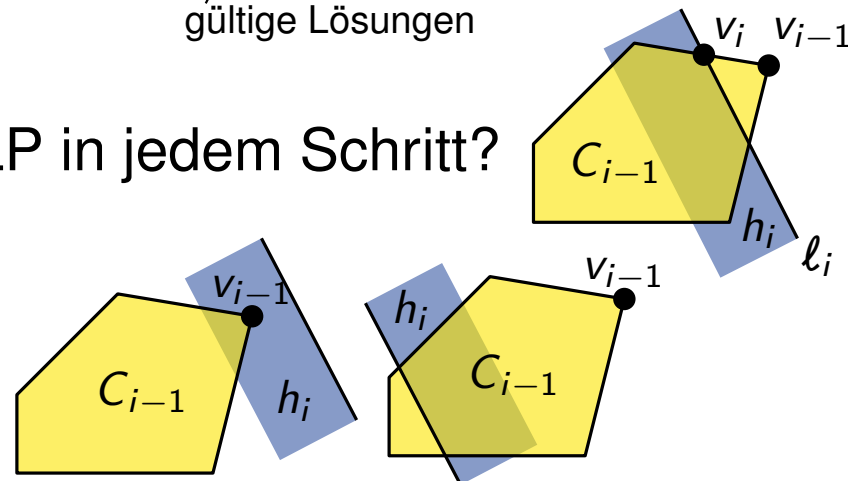
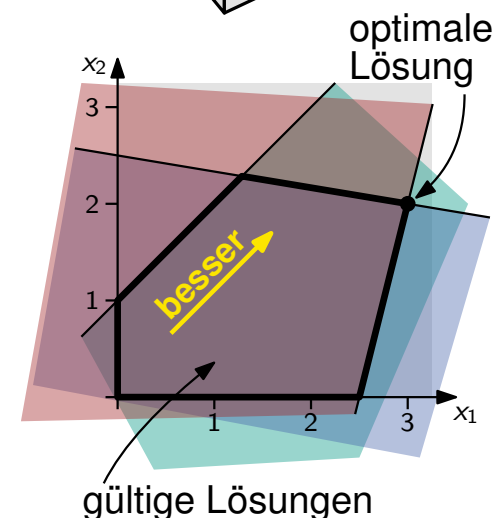
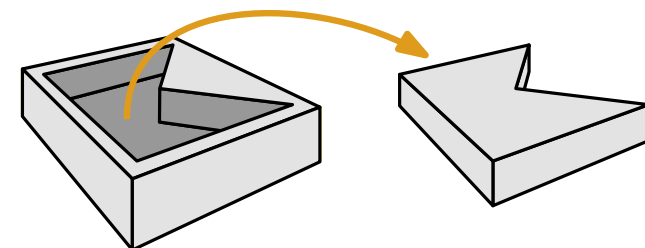
- Was ist ein LP?
- Was hat das mit Geometrie zu tun?

Algorithmus 1: Teile und Herrsche

- Was ist die Idee?
- Welche Laufzeit bekommen wir?

Algorithmus 2: Inkrementeller Ansatz

- Was ist die Idee?
- Inwiefern reduziert sich das auf ein 1D-LP in jedem Schritt?
- Warum hilft eine zufällige Ordnung?
- Welche Laufzeit bekommen wir?
- Warum?
- Inwiefern hilft eine *rückwärts*-Analyse?



Orthogonale Bereichsanfragen

Grundlagen

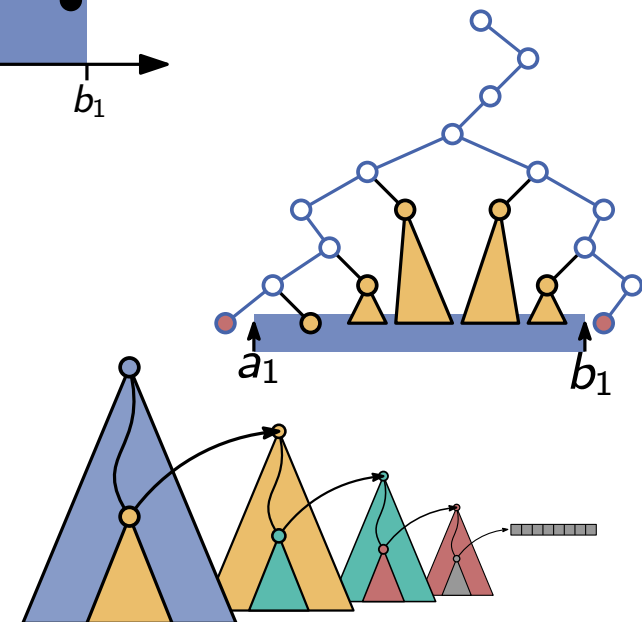
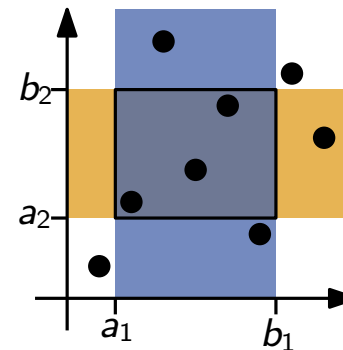
- Was ist die Problemstellung?
- Warum will man das überhaupt lösen?

Range Trees

- Was ist ein 2D-Range Tree?
- Wie sucht man darin und wie lange dauert das?
- Wie baut man ihn auf und wie lange dauert das?
- Wie viel Speicher brauchen wir?
- Verallgemeinerung auf mehrere Dimensionen?

Verbesserte Anfrage

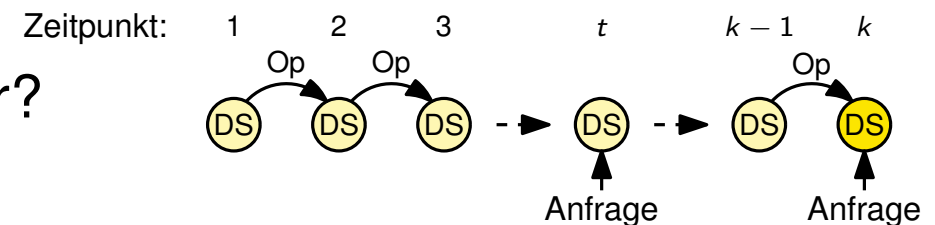
- Wie kann man $\log n$ viele Suchen sparen?
- Wie verallgemeinert sich das zu fractional cascading?
- Wie bekommen wir schnellere Anfragen der Form $[-\infty, b_2] \times [-\infty, b_3]$?
- Anfragen der Form $[a_1, b_1] \times [-\infty, b_2] \times [-\infty, b_3]$?
- Wie werden wir das ∞ los? Was macht das mit Laufzeit/Speicher?



Zeitreisen und Point-Location

Persistenz

- Was ist eine persistente Datenstruktur?
- Welche Arten von Persistenz gibt es?
- Was ist eine verlinkte Datenstruktur?
- Was gilt für verlinkte Datenstrukturen mit beschränkten Eingangsgrad?
- Wie funktioniert das?
 - Was sind atomare Operationen?
 - Wie machen wir diese persistent?
 - Warum sind die rekursiven Aufrufe nicht zu teuer?
 - Wie funktioniert die Potentialmethode (amortisierte Analyse)?



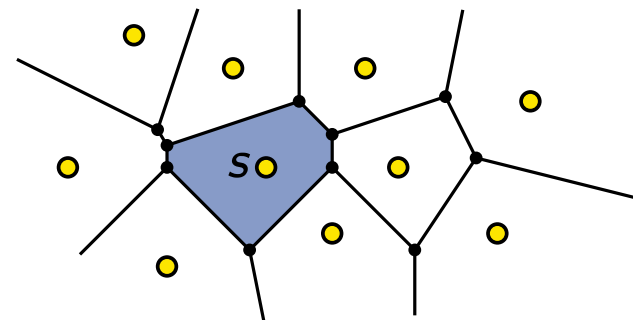
Point-Location

- Was ist das Problem?
- Wie kann man das mit Persistenz lösen?
- Welche Laufzeit/Speicherverbrauch bekommen wir?
- Kann man das auch anders lösen? Wie hilft da fractional cascading?

Voronoi-Diagramme

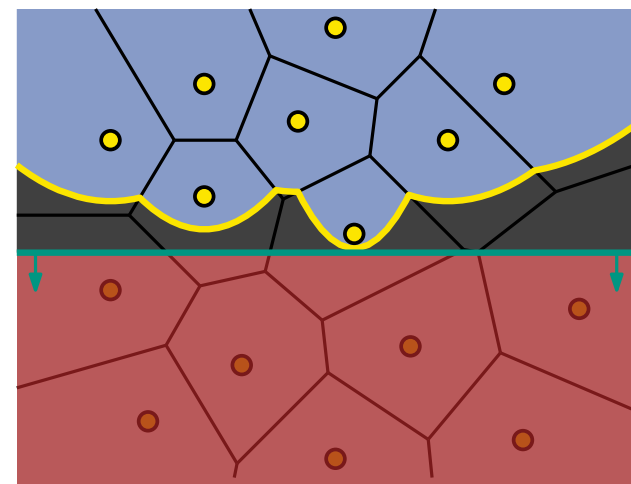
Grundlagen

- Was ist das Voronoi-Diagramm?
- Welche Punkte sind Knoten des Diagramms?
(leere Kreise)



Sweep-Line Algo

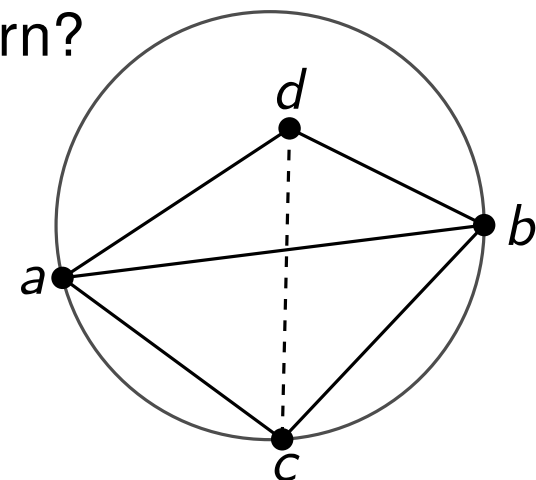
- Was ist bei dem Sweep-Line Algorithmus anders als bei den bisherigen?
- Was ist die Beach-Line?
- Welche Form hat sie und warum?
- Inwiefern ändert sich die Beach-Line nur an wenigen diskreten Stellen?
- Welche sind das? Also was sind unsere Events?
- Wie behandelt man die Events?
- Wie bekommt man am Ende das Voronoi-Diagramm?
- Welche Sonderfälle gibt es? Wie kann man mit ihnen umgehen?



Delaunay-Triangulierung

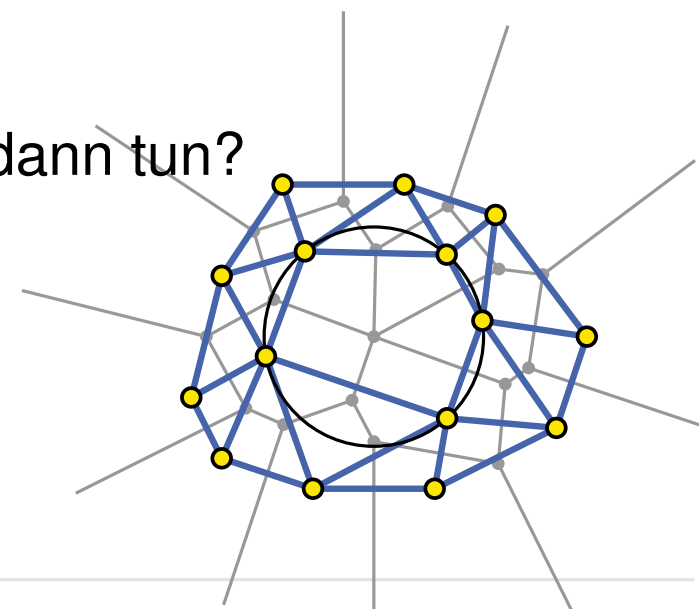
Verbesserung von Triangulierungen

- Was ist der Winkelvektor einer Triangulierung?
- Wie können wir eine Triangulierung iterativ verbessern?
- Was sind verbotene Kanten?
- Welche Kanten sind verboten und was hat das mit Kreisen zu tun?
- Wie hilft da der (allgemeine) Satz des Thales?



Optimale Triangulierung

- Was ist die Delaunay-Triangulierung?
- Ist sie eindeutig?
- Wann ist sie nicht eindeutig und was muss man dann tun?
- Was hat sie mit Voronoi-Diagrammen zu tun?
- Warum gibt es diesen Zusammenhang?



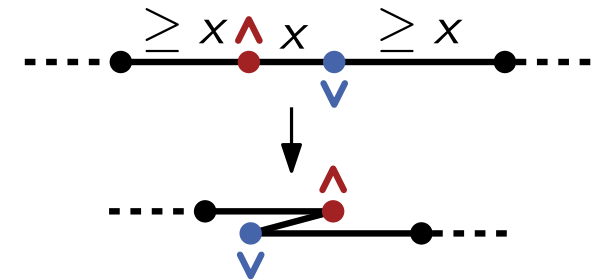
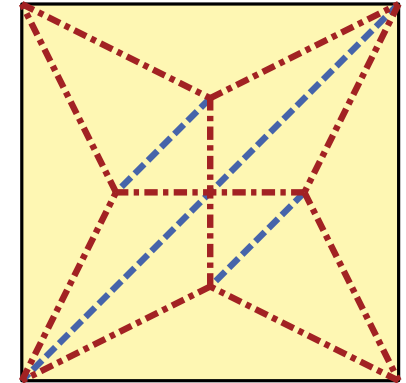
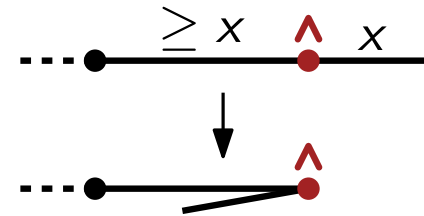
Origami

Grundlagen

- Was ist ein Faltmuster? Was ein Berg-Tal Muster?
- Was bedeutet flache Faltbarkeit?

1D-Fall

- Was ist ein Crimp bzw. ein End-Fold?
- Sind diese Reduktionsregeln sicher? Warum?
- Sind sie immer anwendbar? Warum?
- Wie bekommen wir daraus einen Algorithmus?

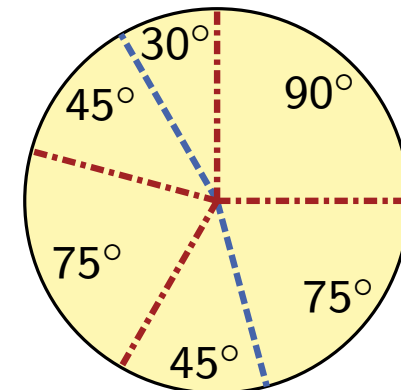


2D-Fall: ein Knoten

- Was hat das mit dem 1D-Fall zu tun?
- Wann ist ein Faltmuster flach faltbar?
- Wie erkennen wir flach faltbare Berg-Tal Muster?

Mehrere Knoten

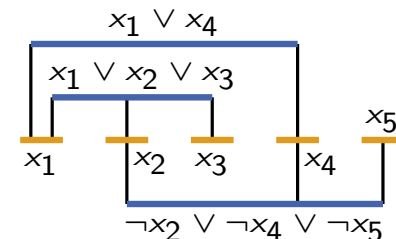
- Was bedeutet lokale Falbarkeit?
- Welche Probleme gibt es in dem Zusammenhang noch und wie schwer sind sie?



Schwere Probleme

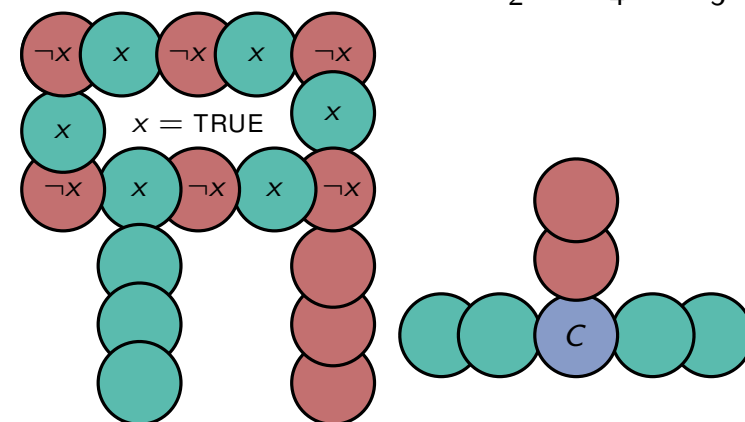
Reduktionen von schweren SAT-Varianten

- Welche SAT-Varianten sind für geometrische Probleme nützlich?
- Was muss man bei einer Reduktion alles bauen?
- Was muss man dann noch beweisen?
- Dürfen die Transportkanäle fehlerhaft sein?



Beispiel 1: Proportional Symbol Maps

- Was ist das? Welche Probleme erhält man?
- Wie beweisen wir Schwere?
- Warum muss man den Idealfall von $1/4$ bzw. $1/8$ Überlappung etwas relaxieren?



Beispiel 2: Unit Disk Graphs

- Was ist das? Welches Problem betrachten wir?
- Wie beweisen wir Schwere?
- Wozu brauchen wir das Splitter-Gadget?
- Warum kann der Graph nicht komplett anders gezeichnet werden?
- Was hat es mit der Komplexitätsklasse $\exists \mathbb{R}$ auf sich?

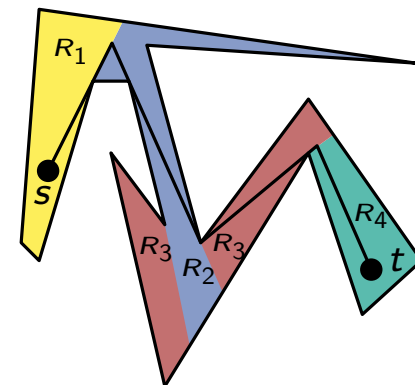
Suchen in sublogarithmischer Zeit

Berechnungsmodelle

- Was sind RAM, real RAM und word RAM?
- Welche Operationen gehen in konstanter Zeit?

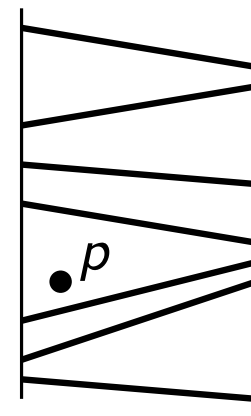
Real RAM

- Warum nützlich für die algorithmische Geometrie?
- Warum muss man damit aber auch vorsichtig sein?



Word RAM: Suchen

- Welche zwei Ideen benutzen wir um die Suche zu beschleunigen?
- Warum ist jede für sich nicht ausreichend?
- Wie verzweigen wir in jedem Suchschritt? Welche Tiefe hat der Baum?
- Wie entscheiden wir uns beim Verzweigen für ein Kind?
- Wie geht das in konstanter Zeit?



Word RAM: 2D-Suchen

- Was sind die Schwierigkeiten im Vergleich zur 1D-Suche?
- Wie lösen wir das trotzdem?

Geometrie

Axiomatischer Ansatz

- Was ist das?
- Wie definiert man Punkte, Geraden, Strecken, etc. ohne Anschauung?
- Wie füllen die Axiome diese Definitionen mit Leben?

Absolute Ebene

- Was ist das?
- Welche Aussagen gelten hier?

Euklidische Ebene

- Was ist das?
- Welche Aussagen gelten hier?

Hyperbolische Ebene

- Was ist das?
- Welche Aussagen gelten hier?
- Wie darf ich mir das vorstellen?
- Was ist das Poincaré Modell?
- Was ist das native Modell?
- Was sind Vor- und Nachteile der Modelle?

Axiomgruppe I: Inzidenzaxiome

zwei Punkte definieren eine Gerade; jede Gerade enthält zwei Punkte; es gibt drei nicht kollineare Punkte

Axiomgruppe II: Abstandsaxiome

Distanz ist eine Metrik; Gleichheit bei Dreiecksungleichung genau dann, wenn kollinear

Axiomgruppe III: Anordnungsaxiome

in jede Richtung mit jedem Abstand gibt es einen Punkt; Geraden zerlegen Ebene in zwei Halbebenen

Axiomgruppe IV: Bewegungsaxiome

es gibt zwei Bewegungen die gleich lange Strecken aufeinander abbilden (orientierungserhaltend)

Axiomgruppe V: Euklidisches Parallelenaxiom

Gerade g und Punkt $P \notin g \Rightarrow$ höchstens eine Gerade parallel zu g durch P

Viel Erfolg!