Vorlesung Algorithmische Geometrie

Spanner und weitere Anwendungen der WSPD

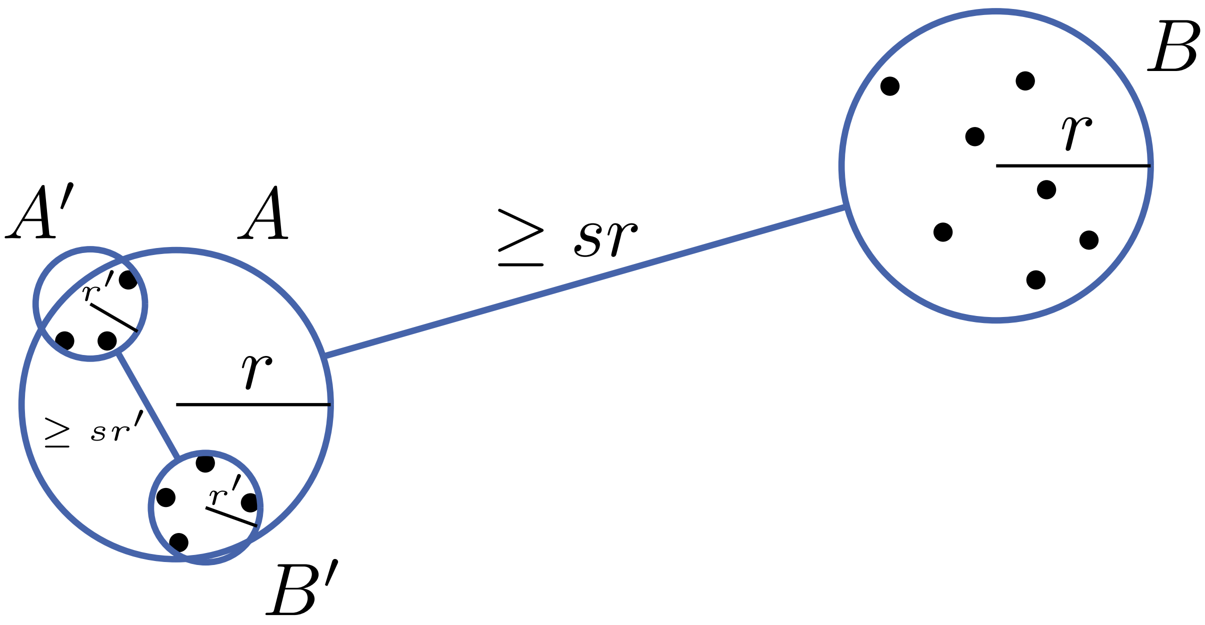
LEHRSTUHL FÜR ALGORITHMIK I · INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Martin Nöllenburg

05.07.2011

Wdh.: Well-Separated Pair Decomposition

Def.: Ein Paar disjunkter Punktmengen A und B im $\mathbb{R}^d$ heißt s-**well separated** für ein $s > 0$, falls A und B jeweils von einer Kugel mit Radius r überdeckt werden und der Abstand der beiden Kugeln mindestens sr ist.



Wdh.: Well-Separated Pair Decomposition

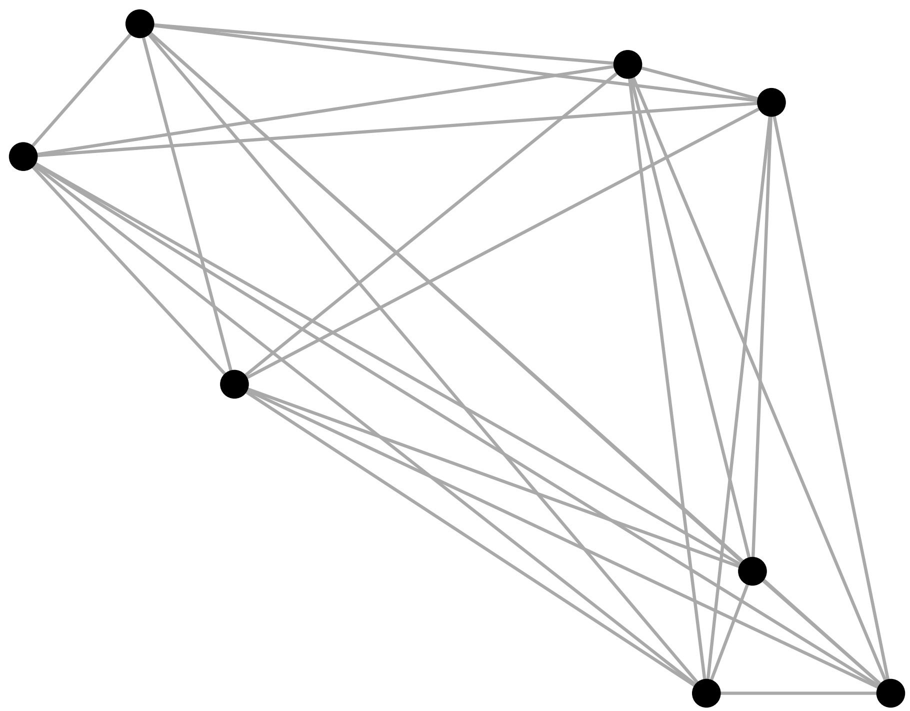
Karlsruhe Institute of Technology

Def.: Ein Paar disjunkter Punktmengen A und B im $\mathbb{R}^d$ heißt s-**well separated** für ein $s > 0$, falls A und B jeweils von einer Kugel mit Radius r überdeckt werden und der Abstand der beiden Kugeln mindestens sr ist.

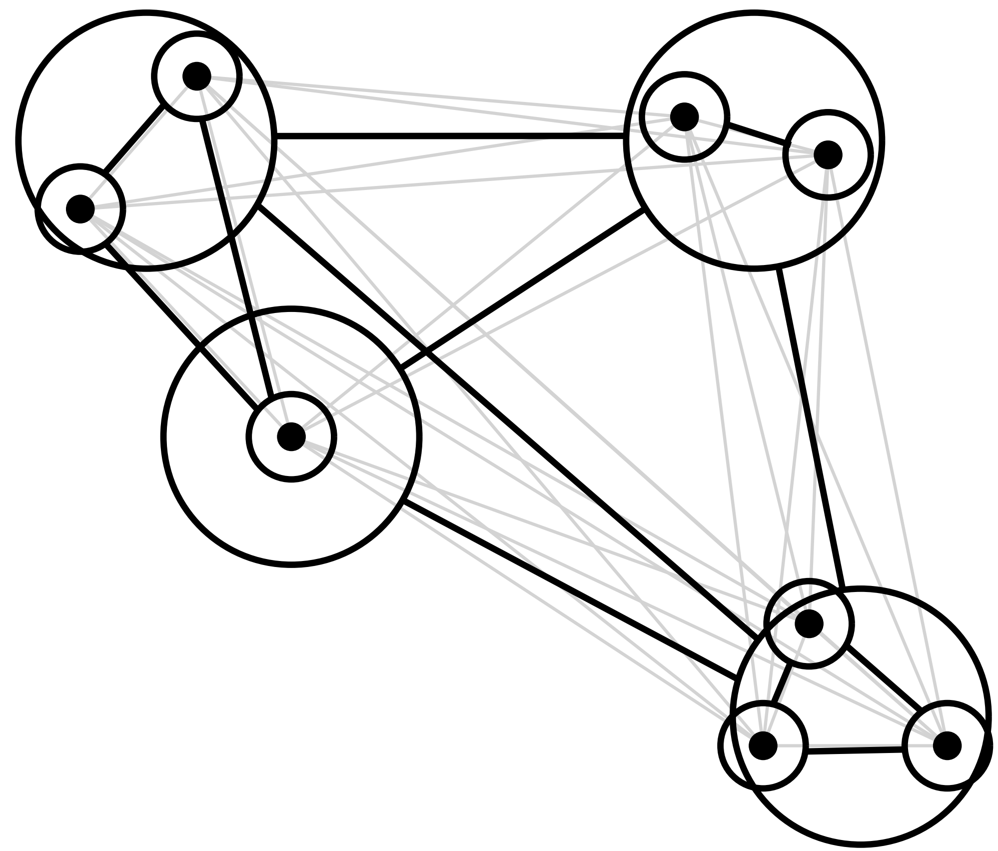
Def.: Für eine Punktmenge P und ein $s > 0$ ist eine s-**well separated pair decomposition** (s-WSPD) eine Menge von Paaren $\{\{A_1, B_1\}, \ldots, \{A_m, B_m\}\}$ mit

- $A_i, B_i \subset P$ für alle i
- $A_i \cap B_i = \emptyset$ für alle i
- $\bigcup_{i=1}^{m} A_i \otimes B_i = P \otimes P$
- $\{A_i, B_i\}$ s-well separated für alle i

Beispiel



28 Punktpaare



12 s-well separated pairs

Analyse WSPD-Konstruktion

Satz 1: Gegeben eine Punktmenge P im $\mathbb{R}^d$ und $s \geq 1$ lässt sich eine s-WSPD mit $O(s^d n)$ Paaren in Zeit $O(n \log n + s^d n)$ konstruieren.

Analyse WSPD-Konstruktion

Satz 1: Gegeben eine Punktmenge P im $\mathbb{R}^d$ und $s \geq 1$ lässt sich eine s-WSPD mit $O(s^d n)$ Paaren in Zeit $O(n \log n + s^d n)$ konstruieren.

Anmerkung zum Beweis:

Behauptung im Beweis war:

$$2(2 + 6s\sqrt{d})^d \in O(s^d)$$

wobei d **feste** Dimension und s **variabler** Separationsfaktor.

Analyse WSPD-Konstruktion

Satz 1: Gegeben eine Punktmenge P im $\mathbb{R}^d$ und $s \geq 1$ lässt sich eine s-WSPD mit $O(s^d n)$ Paaren in Zeit $O(n \log n + s^d n)$ konstruieren.

Anmerkung zum Beweis:

Behauptung im Beweis war:

$$2(2 + 6s\sqrt{d})^d \in O(s^d)$$

wobei d **feste** Dimension und s **variabler** Separationsfaktor.

$$\boxed{O(g(n)) = \{f(n) \mid \exists c, n_0 : 0 \leq f(n) \leq cg(n) \forall n \geq n_0\}}$$

Analyse WSPD-Konstruktion

Satz 1: Gegeben eine Punktmenge P im $\mathbb{R}^d$ und $s \geq 1$ lässt sich eine s-WSPD mit $O(s^d n)$ Paaren in Zeit $O(n \log n + s^d n)$ konstruieren.

Anmerkung zum Beweis:

Behauptung im Beweis war:

$$2(2 + 6s\sqrt{d})^d \in O(s^d)$$

wobei d **feste** Dimension und s **variabler** Separationsfaktor.

$$\boxed{O(g(n)) = \{f(n) \mid \exists c, n_0 : 0 \leq f(n) \leq cg(n) \forall n \geq n_0\}}$$

Wähle nun z.B. $c = 2 \cdot (12\sqrt{d})^d$. Damit folgt die Behauptung.

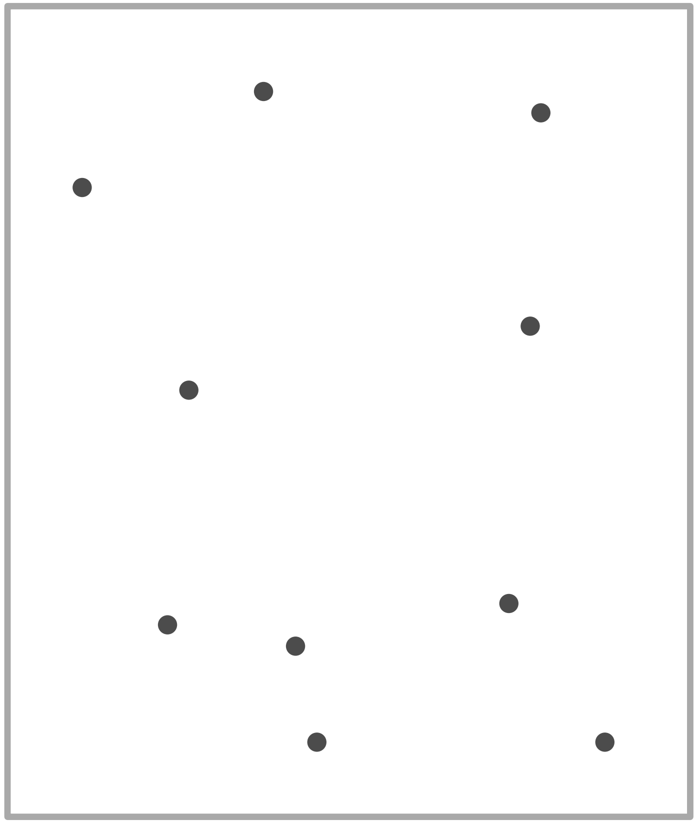
Analyse WSPD-Konstruktion

Satz 1: Gegeben eine Punktmenge P im $\mathbb{R}^d$ und $s \geq 1$ lässt sich eine s-WSPD mit $O(s^d n)$ Paaren in Zeit $O(n \log n + s^d n)$ konstruieren.

Also: s-WSPD hat nur lineare Größe für festes s und d!

Zurück zur Anwendung: t-Spanner

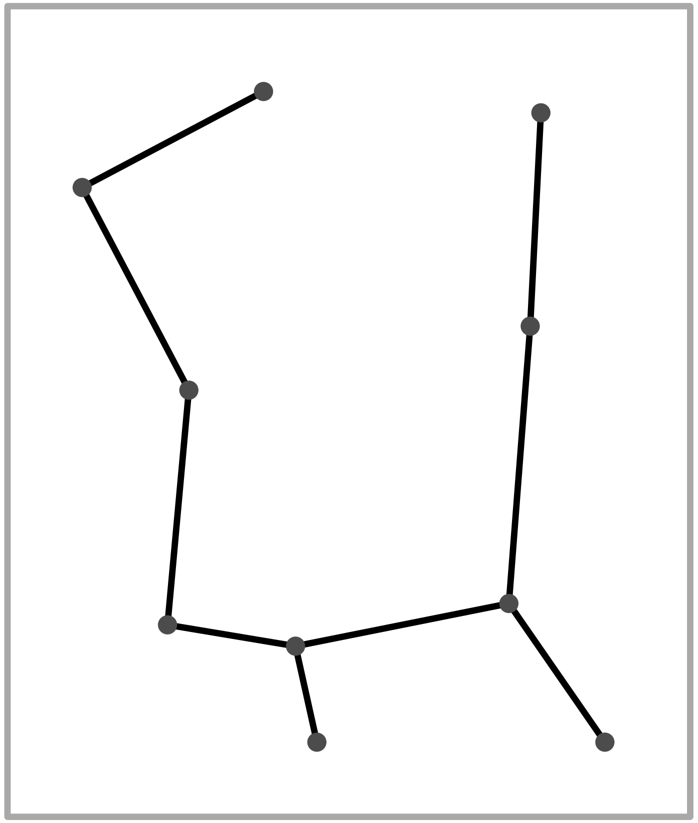
Motivation: Spanner



Aufgabe:

Eine Menge von Städten soll über ein neues Straßennetz miteinander verbunden werden.

Motivation: Spanner

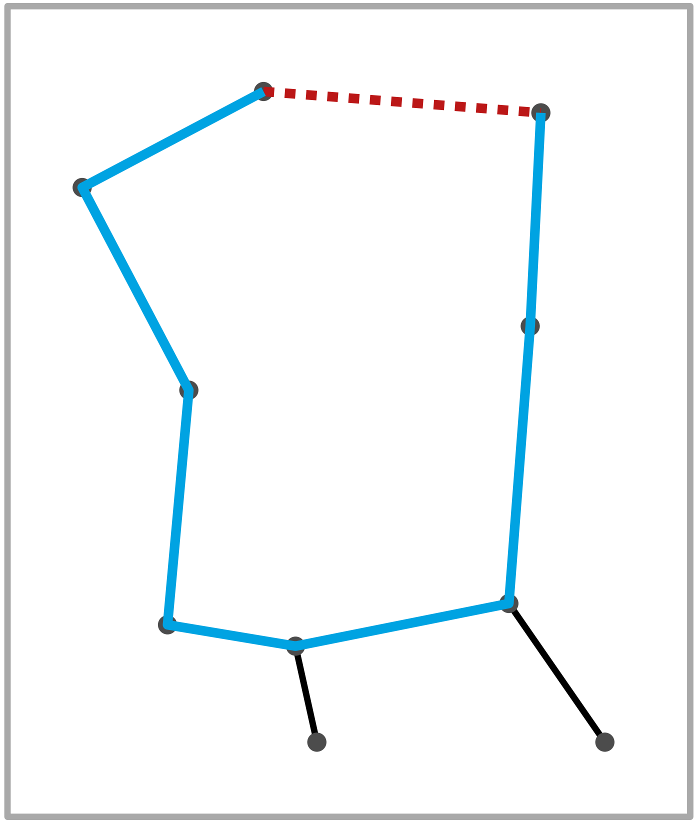


Aufgabe:

Eine Menge von Städten soll über ein neues Straßennetz miteinander verbunden werden.

1. Idee: Euklid. min. Spannbaum

Motivation: Spanner



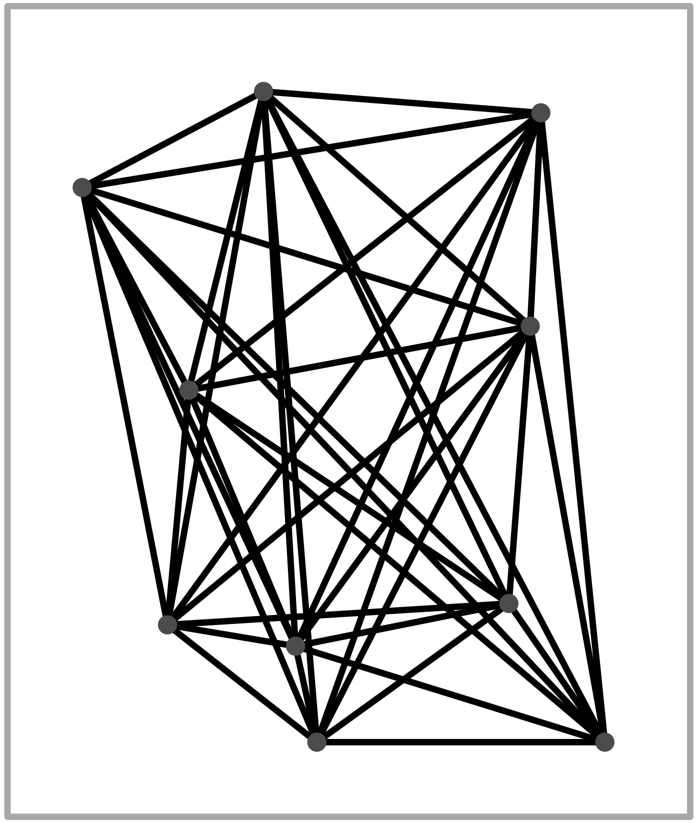
Aufgabe:

Eine Menge von Städten soll über ein neues Straßennetz miteinander verbunden werden.

Allerdings soll für kein Paar (x, y) der Weg im Straßennetz viel länger als die Distanz $||xy||$ sein.

1. Idee: Euklid. min. Spannbaum

Motivation: Spanner



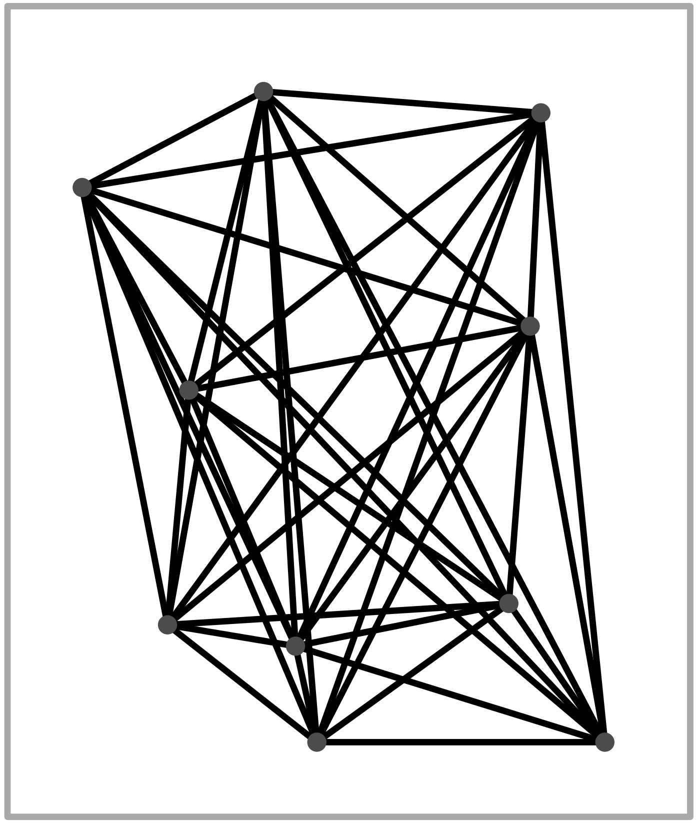
Aufgabe:

Eine Menge von Städten soll über ein neues Straßennetz miteinander verbunden werden.

Allerdings soll für kein Paar (x, y) der Weg im Straßennetz viel länger als die Distanz $||xy||$ sein.

1. Idee: Euklid. min. Spannbaum
2. Idee: vollständiger Graph

Motivation: Spanner



Aufgabe:

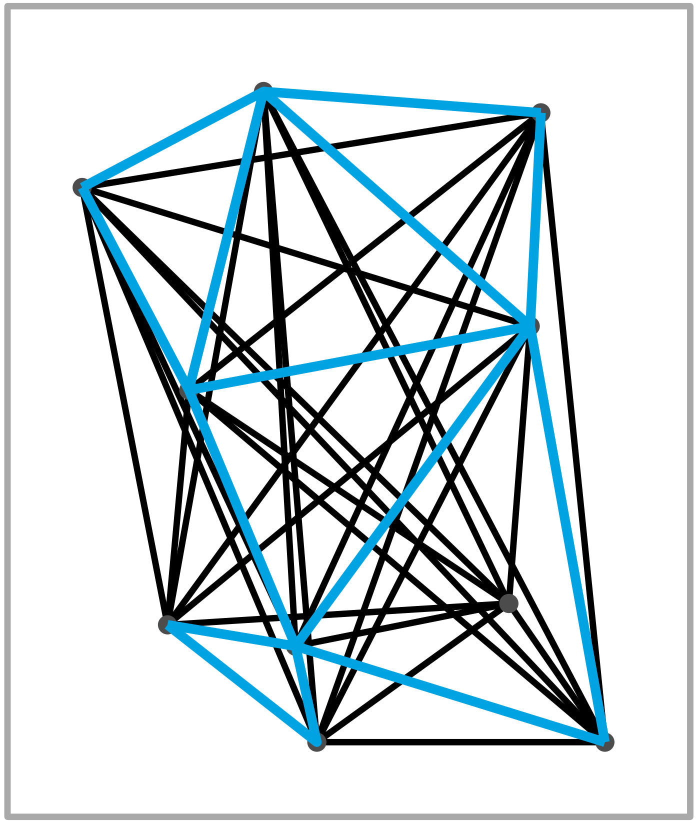
Eine Menge von Städten soll über ein neues Straßennetz miteinander verbunden werden.

Allerdings soll für kein Paar (x, y) der Weg im Straßennetz viel länger als die Distanz $||xy||$ sein.

Die Baukosten sollen im Rahmen bleiben, also z.B. nur $O(n)$ Kanten.

1. Idee: Euklid. min. Spannbaum
2. Idee: vollständiger Graph

Motivation: Spanner



Aufgabe:

Eine Menge von Städten soll über ein neues Straßennetz miteinander verbunden werden.

Allerdings soll für kein Paar (x, y) der Weg im Straßennetz viel länger als die Distanz $||xy||$ sein.

Die Baukosten sollen im Rahmen bleiben, also z.B. nur $O(n)$ Kanten.

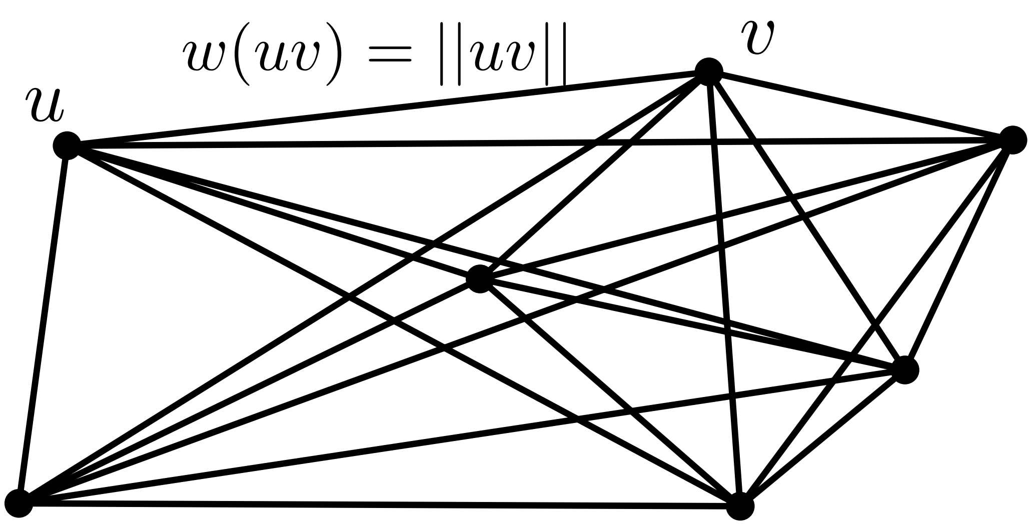
1. Idee: Euklid. min. Spannbaum
2. Idee: vollständiger Graph
3. Idee: **sparse t-Spanner**

t-Spanner

Für eine Menge P von n Punkten in $\mathbb{R}^d$ ist der **Euklidische Graph** $\mathcal{EG}(P) = (P, \binom{P}{2})$ der vollständige gewichtete Graph, dessen Kantengewichte dem Euklidischen Abstand der Kantenendpunkte entsprechen.

t-Spanner

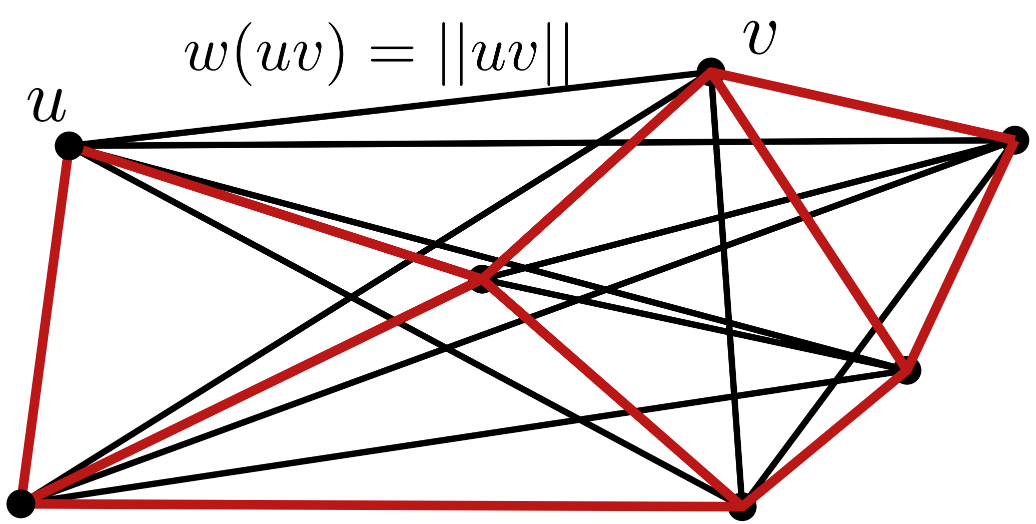
Für eine Menge P von n Punkten in $\mathbb{R}^d$ ist der **Euklidische Graph** $\mathcal{EG}(P) = (P, \binom{P}{2})$ der vollständige gewichtete Graph, dessen Kantengewichte dem Euklidischen Abstand der Kantenendpunkte entsprechen.



t-Spanner

Für eine Menge P von n Punkten in $\mathbb{R}^d$ ist der **Euklidische Graph** $\mathcal{EG}(P) = (P, \binom{P}{2})$ der vollständige gewichtete Graph, dessen Kantengewichte dem Euklidischen Abstand der Kantenendpunkte entsprechen.

Da $\mathcal{EG}(P)$ $\Theta(n^2)$ Kanten hat, wird oft ein dünner Graph mit $O(n)$ Kanten gesucht, dessen kürzeste Wege die Kantengewichte in $\mathcal{EG}(P)$ approximieren.



t-Spanner

Für eine Menge P von n Punkten in $\mathbb{R}^d$ ist der **Euklidische Graph** $\mathcal{EG}(P) = (P, \binom{P}{2})$ der vollständige gewichtete Graph, dessen Kantengewichte dem Euklidischen Abstand der Kantenendpunkte entsprechen.

Da $\mathcal{EG}(P)$ $\Theta(n^2)$ Kanten hat, wird oft ein dünner Graph mit $O(n)$ Kanten gesucht, dessen kürzeste Wege die Kantengewichte in $\mathcal{EG}(P)$ approximieren.

Def.: Ein gewichteter Graph G mit Knotenmenge P heißt t-**Spanner** für P und einen Dehnungsfaktor $t \geq 1$, falls für alle Paare $x, y \in P$ gilt

$$||xy|| \leq \delta_G(x, y) \leq t \cdot ||xy||,$$

wobei $\delta_G(x, y) =$ Länge kürzester x-y-Weg in G.

WSPD und t-Spanner

Def.: Für n Punkte P in $\mathbb{R}^d$ und eine WSPD W von P definiere den Graphen $G = (P, E)$ mit
$E = \{\{x, y\} \mid \{u, v\} \in W$ und $\mathrm{rep}(u) = x, \mathrm{rep}(v) = y\}$.

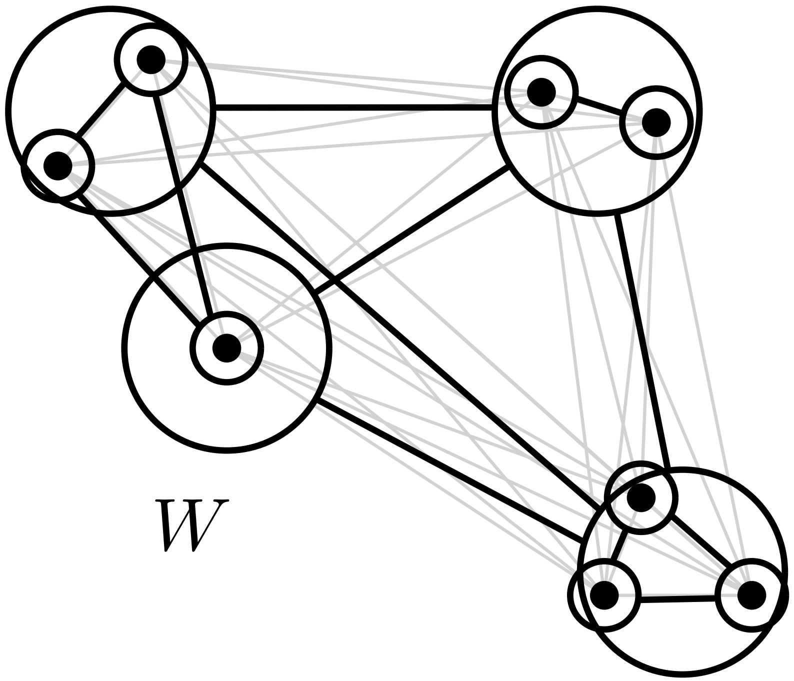
Wdh.: Jedes Paar $\{u, v\} \in W$ entspricht zwei Quadtreeknoten u und v. Aus jedem Quadtreeknoten wird wie folgt ein Repräsentant gewählt. Für Blatt u definiere den Repräsentanten

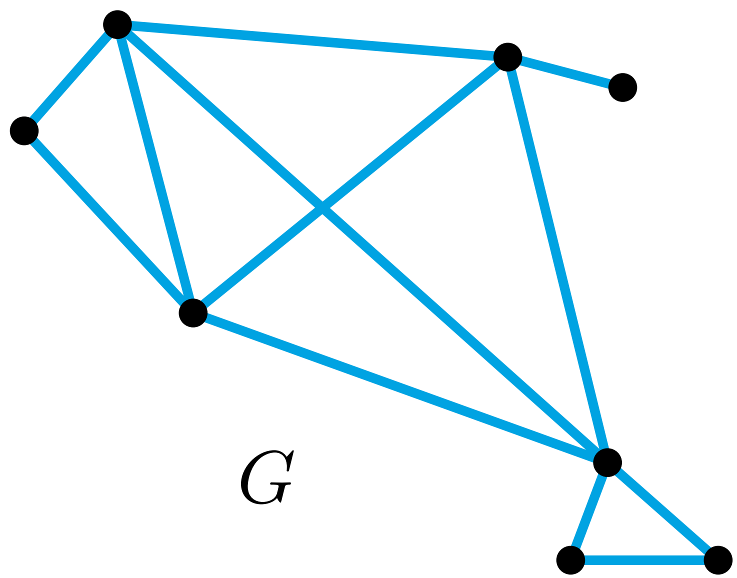
$$\mathrm{rep}(u) = \begin{cases} p & \text{falls } P_u = \{p\} \text{ (} u \text{ ist Blatt)} \\ \emptyset & \text{sonst.} \end{cases}$$

Für einen inneren Knoten v setze $\mathrm{rep}(v) = \mathrm{rep}(u)$ für ein nichtleeres Kind u von v.

WSPD und t-Spanner

Def.: Für n Punkte P in $\mathbb{R}^d$ und eine WSPD W von P definiere den Graphen $G = (P, E)$ mit
$E = \{\{x, y\} \mid \{u, v\} \in W$ und $\mathrm{rep}(u) = x, \mathrm{rep}(v) = y\}$.





WSPD und t-Spanner

Def.: Für n Punkte P in $\mathbb{R}^d$ und eine WSPD W von P definiere den Graphen $G = (P, E)$ mit $E = \{\{x, y\} \mid \{u, v\} \in W$ und $\text{rep}(u) = x, \text{rep}(v) = y\}$.

Lemma 1: Ist W eine s-WSPD für ein geeignetes $s = s(t)$, so ist G ein t-Spanner für P mit $O(s^d n)$ Kanten.

WSPD und t-Spanner

Def.: Für n Punkte P in $\mathbb{R}^d$ und eine WSPD W von P definiere den Graphen $G = (P, E)$ mit
$E = \{\{x, y\} \mid \{u, v\} \in W$ und $\text{rep}(u) = x, \text{rep}(v) = y\}$.

Lemma 1: Ist W eine s-WSPD für ein geeignetes $s = s(t)$, so ist G ein t-Spanner für P mit $O(s^d n)$ Kanten.

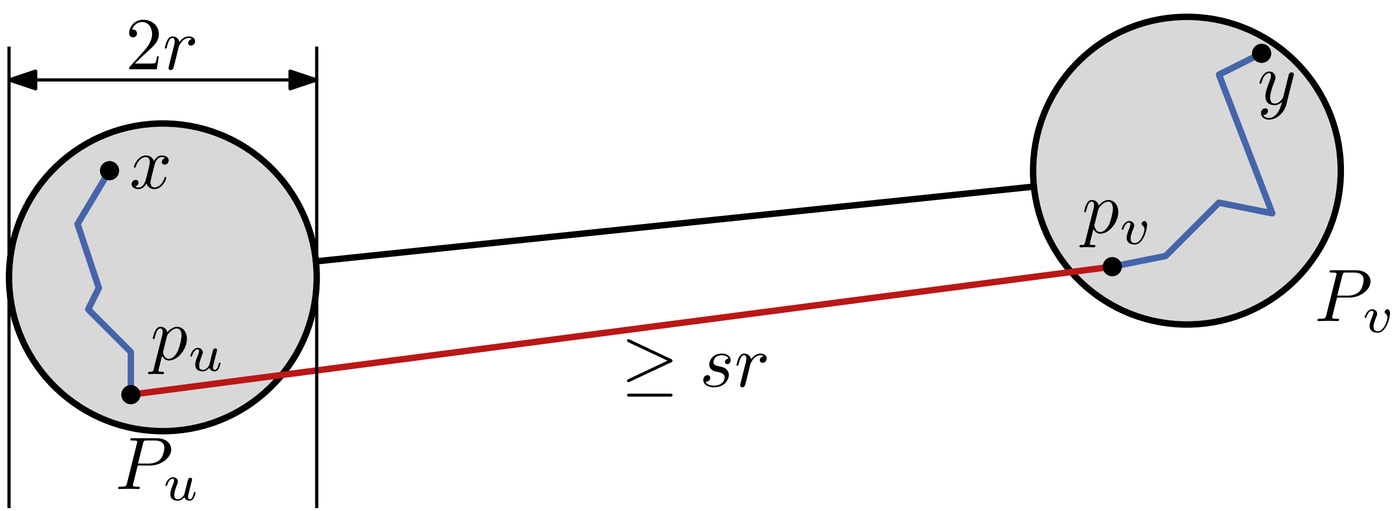
Beweis: (Tafel)

WSPD und t-Spanner

Def.: Für n Punkte P in $\mathbb{R}^d$ und eine WSPD W von P definiere den Graphen $G = (P, E)$ mit
$E = \{\{x, y\} \mid \{u, v\} \in W$ und $\mathrm{rep}(u) = x, \mathrm{rep}(v) = y\}$.

Lemma 1: Ist W eine s-WSPD für ein geeignetes $s = s(t)$, so ist G ein t-Spanner für P mit $O(s^d n)$ Kanten.

Beweis: (Tafel)



Zusammenfassung

Satz 2: Für eine Menge P von n Punkten in $\mathbb{R}^d$ und ein $\varepsilon \in (0,1]$ kann ein $(1+\varepsilon)$-Spanner für P mit $O(n/\varepsilon^d)$ Kanten in $O(n \log n + n/\varepsilon^d)$ Zeit berechnet werden.

Zusammenfassung

Satz 2: Für eine Menge P von n Punkten in $\mathbb{R}^d$ und ein $\varepsilon \in (0,1]$ kann ein $(1+\varepsilon)$-Spanner für P mit $O(n/\varepsilon^d)$ Kanten in $O(n \log n + n/\varepsilon^d)$ Zeit berechnet werden.

Beweis: Für $t = (1+\varepsilon)$ gilt mit $s = 4 \cdot \frac{t+1}{t-1}$

$$O(s^d n) = O\left(\left(4 \cdot \frac{2+\varepsilon}{\varepsilon}\right)^d n\right) \subseteq O\left(\left(\frac{12}{\varepsilon}\right)^d n\right) = O\left(\frac{n}{\varepsilon^d}\right)$$

Zusammenfassung

Satz 2: Für eine Menge P von n Punkten in $\mathbb{R}^d$ und ein $\varepsilon \in (0,1]$ kann ein $(1+\varepsilon)$-Spanner für P mit $O(n/\varepsilon^d)$ Kanten in $O(n \log n + n/\varepsilon^d)$ Zeit berechnet werden.

Beweis: Für $t = (1+\varepsilon)$ gilt mit $s = 4 \cdot \frac{t+1}{t-1}$

$$O(s^d n) = O\left(\left(4 \cdot \frac{2+\varepsilon}{\varepsilon}\right)^d n\right) \subseteq O\left(\left(\frac{12}{\varepsilon}\right)^d n\right) = O\left(\frac{n}{\varepsilon^d}\right)$$

P

$\downarrow$ $\quad O(n \log n)$

komprimierter Quadtree

$\downarrow$ $\quad O(n/\varepsilon^d)$

WSPD

$\downarrow$ $\quad O(n/\varepsilon^d)$

$(1+\varepsilon)$-Spanner

$\square$

Weitere Anwendungen der WSPD

Euklidischer MST

Problem: Finde für eine Punktmenge P einen minimalen Spannbaum (MST) im Euklidischen Graphen $\mathcal{EG}(P)$.

Euklidischer MST

Problem: Finde für eine Punktmenge P einen minimalen Spannbaum (MST) im Euklidischen Graphen $\mathcal{EG}(P)$.

Prim: MST in einem Graph $G = (V, E)$ kann in $O(|E| + |V| \log |V|)$ Zeit berechnet werden.

Euklidischer MST

Problem: Finde für eine Punktmenge P einen minimalen Spannbaum (MST) im Euklidischen Graphen $\mathcal{EG}(P)$.

Prim: MST in einem Graph $G = (V, E)$ kann in $O(|E| + |V| \log |V|)$ Zeit berechnet werden.

- $\mathcal{EG}(P)$ hat $\Theta(n^2)$ Kanten $\Rightarrow$ Laufzeit $O(n^2)$:-(
- $(1 + \varepsilon)$-Spanner für P hat $O(n/\varepsilon^d)$ Kanten $\Rightarrow$ Laufzeit $O(n \log n + n/\varepsilon^d)$:-)

Euklidischer MST

Problem: Finde für eine Punktmenge P einen minimalen Spannbaum (MST) im Euklidischen Graphen $\mathcal{EG}(P)$.

Prim: MST in einem Graph $G = (V, E)$ kann in $O(|E| + |V| \log |V|)$ Zeit berechnet werden.

- $\mathcal{EG}(P)$ hat $\Theta(n^2)$ Kanten $\Rightarrow$ Laufzeit $O(n^2)$:-(
- $(1 + \varepsilon)$-Spanner für P hat $O(n/\varepsilon^d)$ Kanten
  $\Rightarrow$ Laufzeit $O(n \log n + n/\varepsilon^d)$:-)

Wie gut ist der MST eines $(1 + \varepsilon)$-Spanners?

Euklidischer MST

Problem: Finde für eine Punktmenge P einen minimalen Spannbaum (MST) im Euklidischen Graphen $\mathcal{EG}(P)$.

Prim: MST in einem Graph $G = (V, E)$ kann in $O(|E| + |V| \log |V|)$ Zeit berechnet werden.

- $\mathcal{EG}(P)$ hat $\Theta(n^2)$ Kanten $\Rightarrow$ Laufzeit $O(n^2)$:-(
- $(1 + \varepsilon)$-Spanner für P hat $O(n/\varepsilon^d)$ Kanten
  $\Rightarrow$ Laufzeit $O(n \log n + n/\varepsilon^d)$:-)

Wie gut ist der MST eines $(1 + \varepsilon)$-Spanners?

Satz 3: Der aus einem $(1 + \varepsilon)$-Spanner von P gewonnene MST ist eine $(1 + \varepsilon)$-Approximation des EMST von P.

Durchmesser von P

Problem: Finde den Durchmesser einer Punktmenge P, d.h. das Paar $\{x, y\} \subset P$ mit größtem Abstand.

Durchmesser von P

Problem: Finde den Durchmesser einer Punktmenge P, d.h. das Paar $\{x, y\} \subset P$ mit größtem Abstand.

- brute-force alle Punktpaare testen $\Rightarrow$ Laufzeit $O(n^2)$:-(
- teste Abstände $||\text{rep}(u)\ \text{rep}(v)||$ aller ws-Paare $\{P_u, P_v\}$ $\Rightarrow$ Laufzeit $O(n \log n + s^d n)$:-)

Durchmesser von P

Problem: Finde den Durchmesser einer Punktmenge P, d.h. das Paar $\{x, y\} \subset P$ mit größtem Abstand.

- brute-force alle Punktpaare testen $\Rightarrow$ Laufzeit $O(n^2)$:-(
- teste Abstände $||$rep(u) rep$(v)||$ aller ws-Paare $\{P_u, P_v\}$ $\Rightarrow$ Laufzeit $O(n \log n + s^d n)$:-)

Wie gut ist der berechnete Durchmesser?

Durchmesser von P

Problem: Finde den Durchmesser einer Punktmenge P, d.h. das Paar $\{x, y\} \subset P$ mit größtem Abstand.

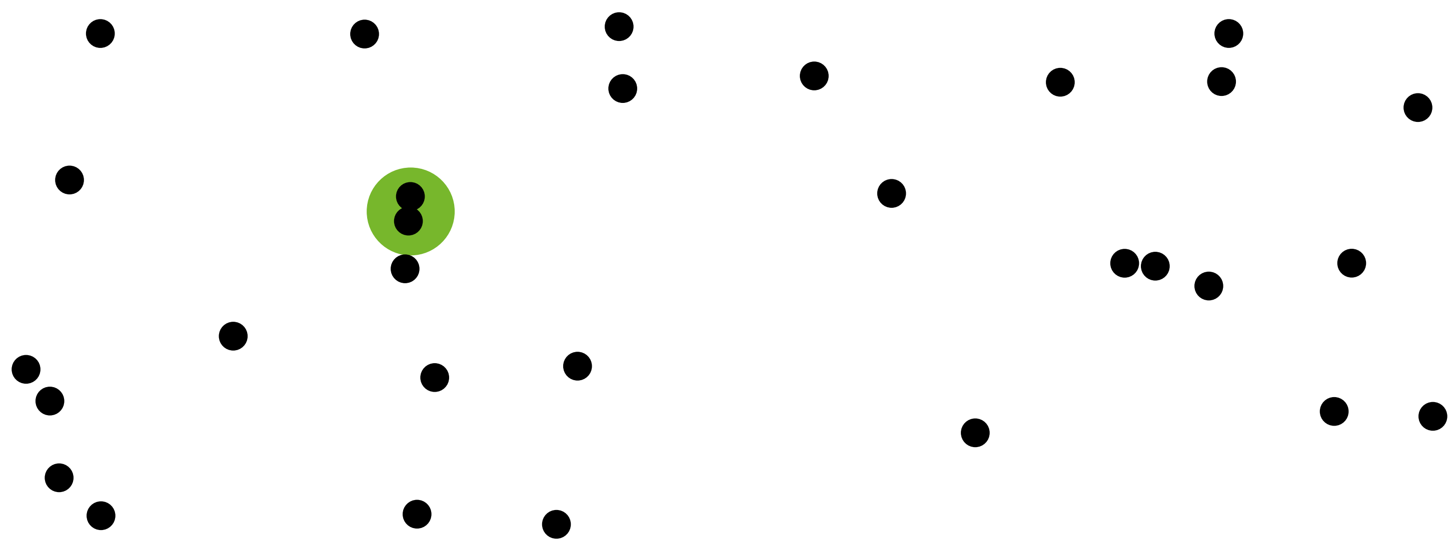
- brute-force alle Punktpaare testen $\Rightarrow$ Laufzeit $O(n^2)$:-(
- teste Abstände $||$rep(u) rep$(v)||$ aller ws-Paare $\{P_u, P_v\}$ $\Rightarrow$ Laufzeit $O(n \log n + s^d n)$:-)

Wie gut ist der berechnete Durchmesser?

Satz 4: Der aus einer s-WSPD von P berechnete Durchmesser ist eine $(1 + \varepsilon)$-Approximation des Durchmessers von P.

Nähestes Punktepaar

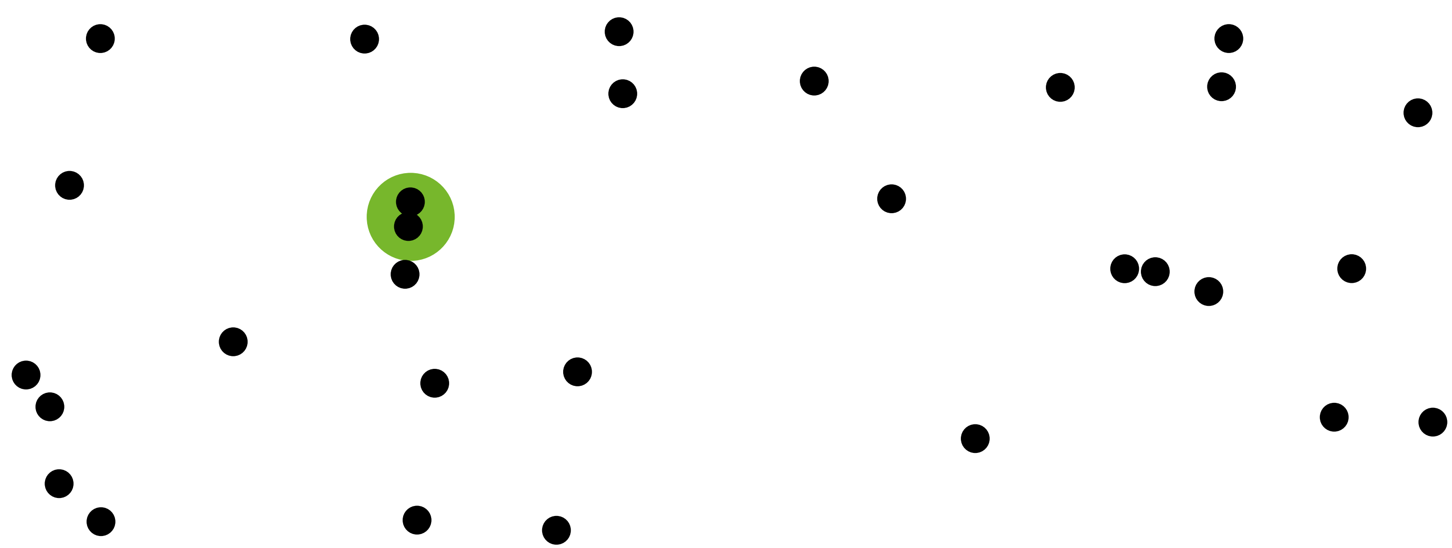
Problem: Finde das Paar $\{x, y\} \subset P$ mit geringstem Abstand.



Nähestes Punktepaar

Problem: Finde das Paar $\{x, y\} \subset P$ mit geringstem Abstand.

- brute-force alle Punktpaare testen $\Rightarrow$ Laufzeit $O(n^2)$:-(
- teste Abstände $||\text{rep}(u)\ \text{rep}(v)||$ aller ws-Paare $\{P_u, P_v\}$
  $\Rightarrow$ Laufzeit $O(n \log n + s^d n)$:-)

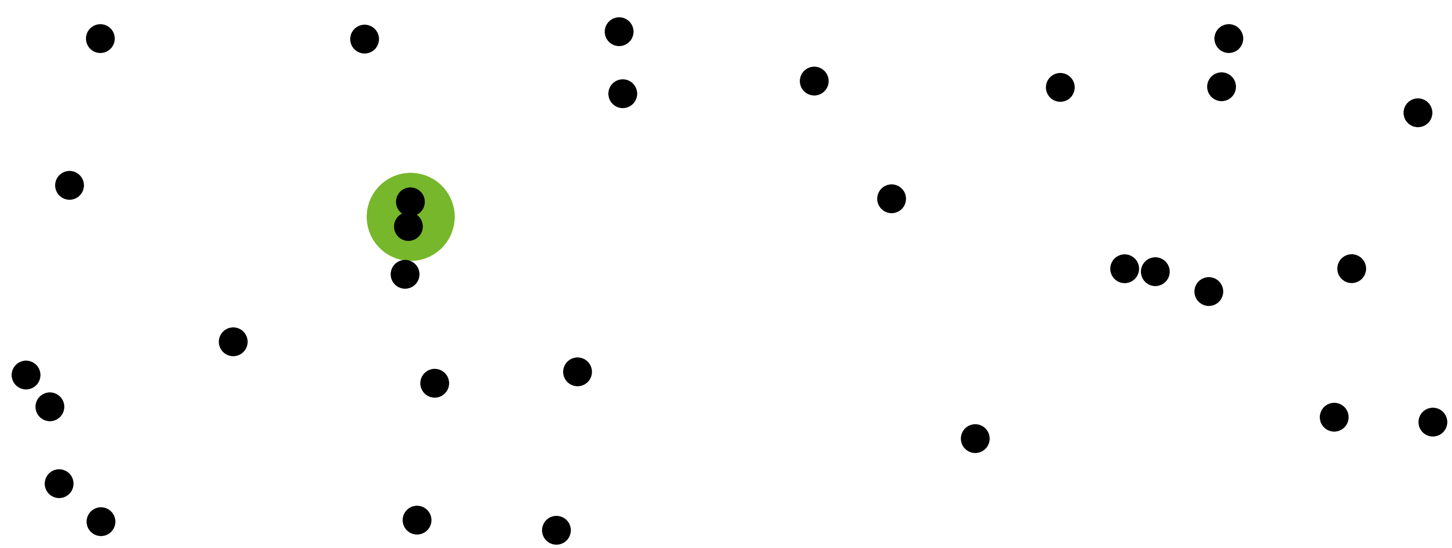


Nähestes Punktepaar

Problem: Finde das Paar $\{x, y\} \subset P$ mit geringstem Abstand.

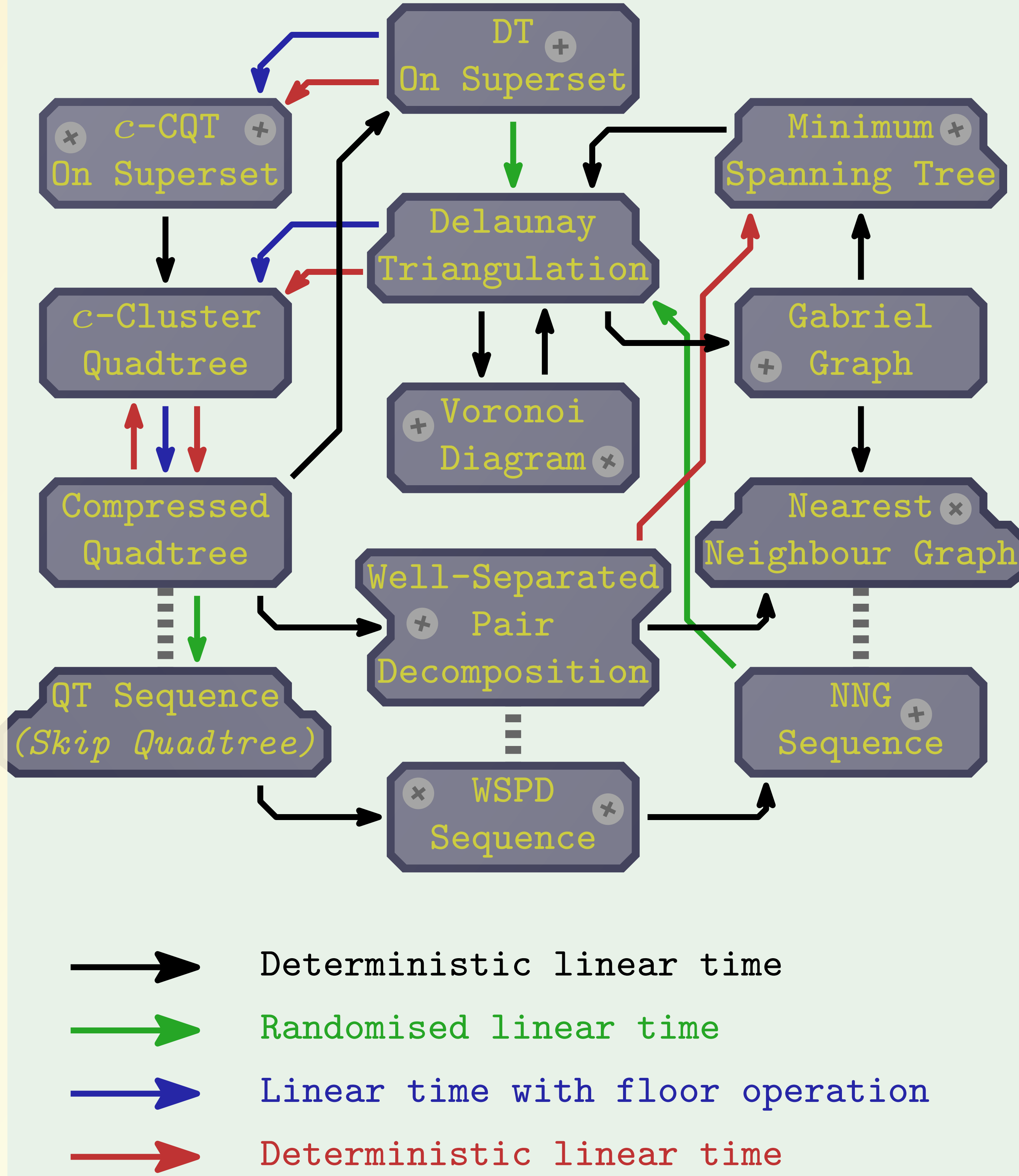
- brute-force alle Punktpaare testen $\Rightarrow$ Laufzeit $O(n^2)$:-(
- teste Abstände $||\text{rep}(u)\ \text{rep}(v)||$ aller ws-Paare $\{P_u, P_v\}$ $\Rightarrow$ Laufzeit $O(n \log n + s^d n)$:-)

Übungsblatt: Für $s > 2$ liefert dieser Ansatz sogar exakt das näheste Paar.



Proximity Structures

[Dirichlet, 1850]
[Delaunay, 1934]
[Gabriel, 1969]
[Clarkson, 1983]
[Preparata & Shamos, 1985]
[Bern, Eppstein & Gilbert, 1990]
[Chazelle, 1991]
[Matsui, 1995]
[Callahan & Kosaraju, 1995]
[Chin & Wang, 1998]
[Krznaric & Levcopoulos, 1998]
[Chazelle, Devillers, Hurtado, Mora, Sacristán & Teillaud, 2001]
[Eppstein, Goodrich & Sun, 2005]
[Buchin & Mulzer, 2009]
[Löffler & Mulzer, 2011]



Diskussion

Welche weiteren Anwendungen hat die WSPD?

Diskussion

Welche weiteren Anwendungen hat die WSPD?

WSPD bietet sich immer dann an, wenn man auf die $\Theta(n^2)$ exakten Distanzen in einer Punktmenge verzichten kann und diese stattdessen approximiert. Ein Beispiel sind kräftebasierte Layoutalgorithmen im Graphenzeichnen, wo paarweise Abstoßungskräfte von n Objekten berechnet werden müssen.

Diskussion

Welche weiteren Anwendungen hat die WSPD?

WSPD bietet sich immer dann an, wenn man auf die $\Theta(n^2)$ exakten Distanzen in einer Punktmenge verzichten kann und diese stattdessen approximiert. Ein Beispiel sind kräftebasierte Layoutalgorithmen im Graphenzeichnen, wo paarweise Abstoßungskräfte von n Objekten berechnet werden müssen.

Wozu geometrisch approximieren?

Diskussion

Welche weiteren Anwendungen hat die WSPD?

WSPD bietet sich immer dann an, wenn man auf die $\Theta(n^2)$ exakten Distanzen in einer Punktmenge verzichten kann und diese stattdessen approximiert. Ein Beispiel sind kräftebasierte Layoutalgorithmen im Graphenzeichnen, wo paarweise Abstoßungskräfte von n Objekten berechnet werden müssen.

Wozu geometrisch approximieren?

Einerseits ersetzt man dadurch langsame Berechnungen durch schnellere (aber ungenauere), andererseits sind oft auch die Eingabedaten schon mit einer gewissen Ungenauigkeit behaftet, so dass approximative Lösungen je nach Anwendung ausreichend sind.

Diskussion

Welche weiteren Anwendungen hat die WSPD?

WSPD bietet sich immer dann an, wenn man auf die $\Theta(n^2)$ exakten Distanzen in einer Punktmenge verzichten kann und diese stattdessen approximiert. Ein Beispiel sind kräftebasierte Layoutalgorithmen im Graphenzeichnen, wo paarweise Abstoßungskräfte von n Objekten berechnet werden müssen.

Wozu geometrisch approximieren?

Einerseits ersetzt man dadurch langsame Berechnungen durch schnellere (aber ungenauere), andererseits sind oft auch die Eingabedaten schon mit einer gewissen Ungenauigkeit behaftet, so dass approximative Lösungen je nach Anwendung ausreichend sind.

Geht es nicht auch genauso schnell exakt?

Diskussion

Welche weiteren Anwendungen hat die WSPD?

WSPD bietet sich immer dann an, wenn man auf die $\Theta(n^2)$ exakten Distanzen in einer Punktmenge verzichten kann und diese stattdessen approximiert. Ein Beispiel sind kräftebasierte Layoutalgorithmen im Graphenzeichnen, wo paarweise Abstoßungskräfte von n Objekten berechnet werden müssen.

Wozu geometrisch approximieren?

Einerseits ersetzt man dadurch langsame Berechnungen durch schnellere (aber ungenauere), andererseits sind oft auch die Eingabedaten schon mit einer gewissen Ungenauigkeit behaftet, so dass approximative Lösungen je nach Anwendung ausreichend sind.

Geht es nicht auch genauso schnell exakt?

Oft im $\mathbb{R}^2$ ja, aber nicht mehr im $\mathbb{R}^d$ für $d > 2$. (EMST, Durchmesser)

Was haben wir alles behandelt?

Was haben wir alles behandelt?

t-Spanner

Ebenenunterteilung

konvexe Hülle (2d und 3d)

Quadtrees

randomisierte Algorithmen

kd-Trees

Voronoi-Diagramme

Dualität

Polygontriangulierung

Streckenschnitte

WSPD

Schnitt von Halbebenen

Delaunay-Triangulierung

Sweep-Line Algorithmen

Trapezzerlegung

Range-Trees

Geradenarrangements

Divide & Conquer

Was kommt in der Prüfung dran?

- grundlegende Problemdefinitionen

Was kommt in der Prüfung dran?

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen

Was kommt in der Prüfung dran?

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen

Was kommt in der Prüfung dran?

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen

Was kommt in der Prüfung dran?

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale Beweisideen, nicht die technischen Details

Was kommt in der Prüfung dran?

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale Beweisideen, nicht die technischen Details
- Einsatz der Algorithmen und Datenstrukturen in Anwendungen

Was kommt in der Prüfung dran?

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale Beweisideen, nicht die technischen Details
- Einsatz der Algorithmen und Datenstrukturen in Anwendungen
- bei 3SWS bzw. im Master: Stoff der Übungen

Was kommt in der Prüfung dran?

- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale Beweisideen, nicht die technischen Details
- Einsatz der Algorithmen und Datenstrukturen in Anwendungen
- bei 3SWS bzw. im Master: Stoff der Übungen

Wie vorbereiten?

- Vorlesungsfolien **und** Tafelbeweise aus angegebener Literatur (v.a. [BCKO08])
- Übungsaufgaben
- nicht behandeltes Material aus der Literatur nur ergänzend

Was kommt in der Prüfung dran?

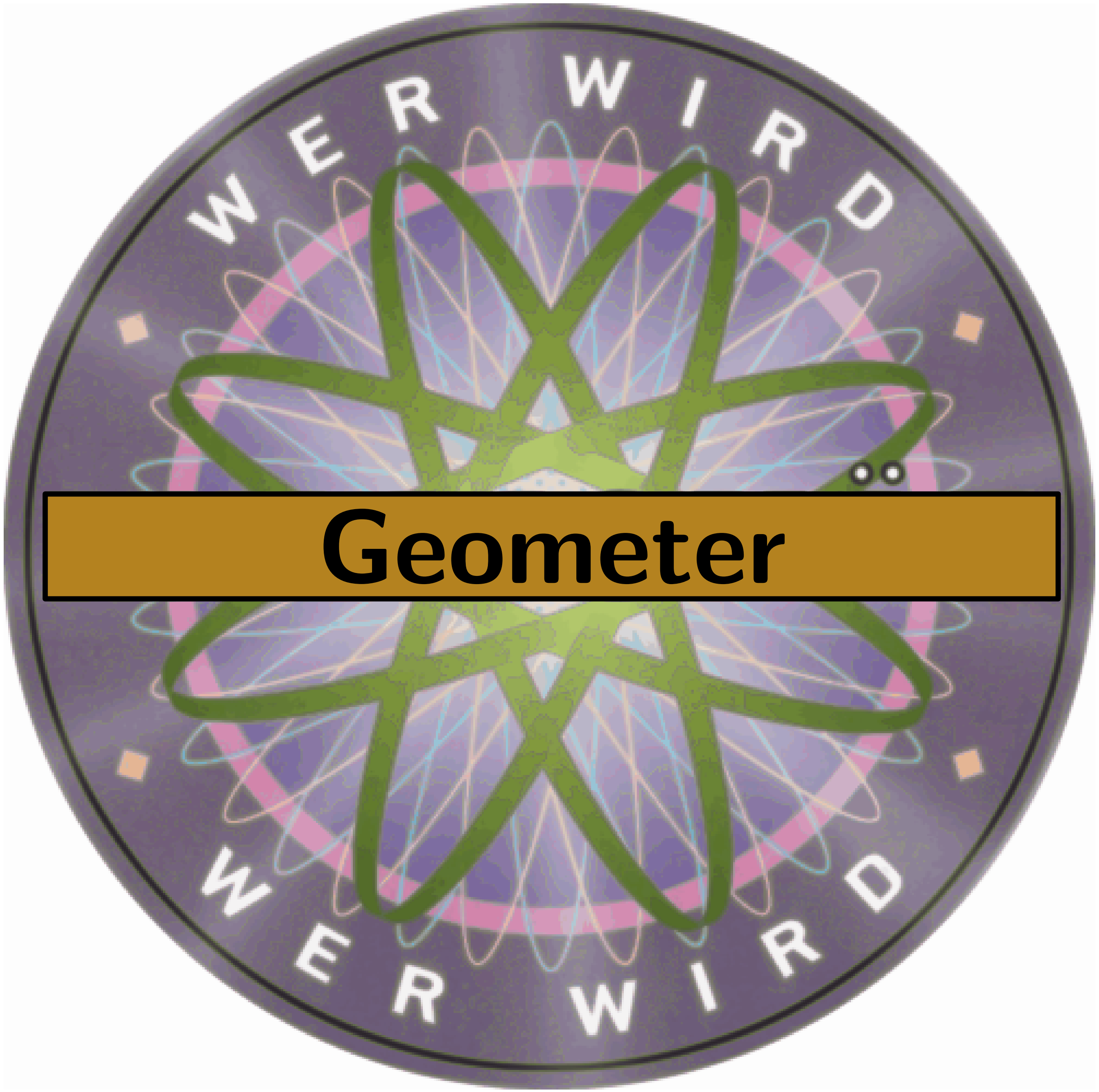
- grundlegende Problemdefinitionen
- Aufbau und Queries von Datenstrukturen
- Zusammenhänge zwischen verschiedenen Datenstrukturen
- Eigenschaften und Ablauf der Algorithmen
- zentrale ils
- Einsatz
  Anwend
- bei 3SWS bzw. im Master: Stoff der Übungen

Weitere Fragen von euch?

Wie vorbereiten?

- Vorlesungsfolien **und** Tafelbeweise aus angegebener Literatur (v.a. [BCKO08])
- Übungsaufgaben
- nicht behandeltes Material aus der Literatur nur ergänzend


WER WIRD
MILLIONÄR
WER WIRD


WER WIRD
Geometer
WER WIRD