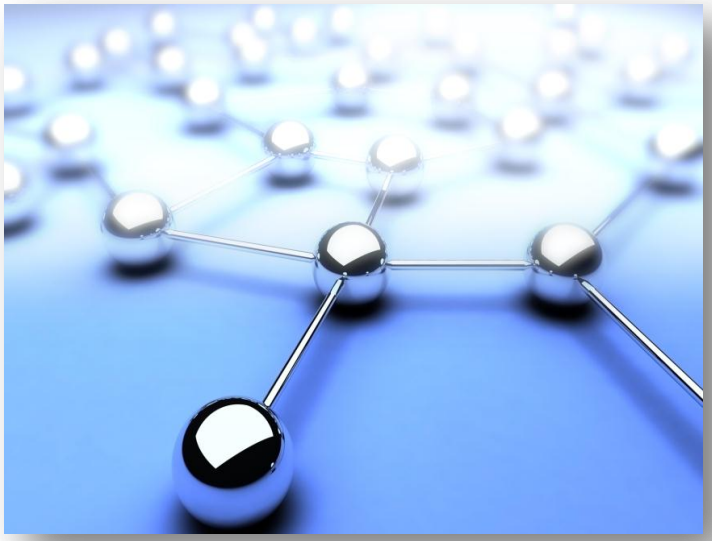
HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Vorlesung Algorithmen für hochkomplexe Virtuelle Szenen

Sommersemester 2012



Matthias Fischer
mafi@upb.de

Vorlesung 4
24.4.2012

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

kd-Baum = k-dimensional tree

2-dimensionale kd-Bäume

- Motivation
- Problem
- Baumstruktur
- Konstruktion
- Bereichsanfrage

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

kd-Bäume

Computational Geometry - Algorithms and Applications;
Mark de Berg, Otfried Cheong, Marc de Kreveld, Mark Overmars;
Springer Verlag, 2008.

Kapitel: Orthogonal Range Searching


HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Vorgestellt von Bentley 1975

Ziele

Sortierung von Objekten / Punkten im d-dimensionalen Raum

Anwendungen

- (orthogonale) Bereichsanfragen
- Verdeckungsberechnung
- Weitere …

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

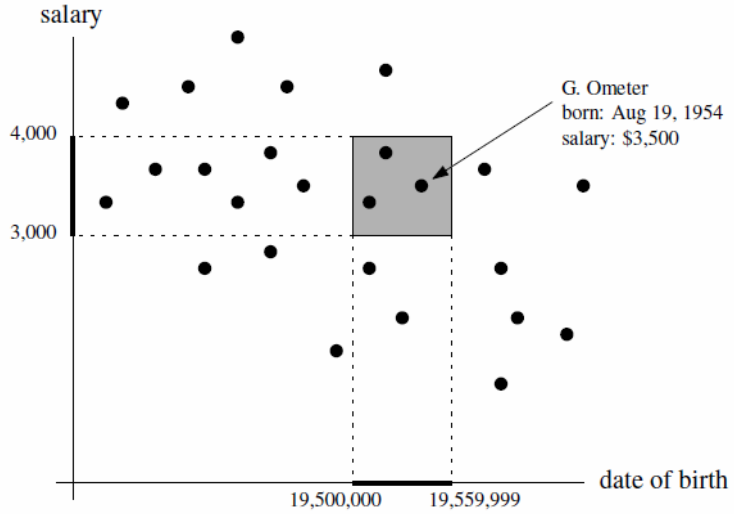
Orthogonale Bereichsanfrage

Gegeben:

eine Menge von Punkten und ein Rechteck oder Würfel

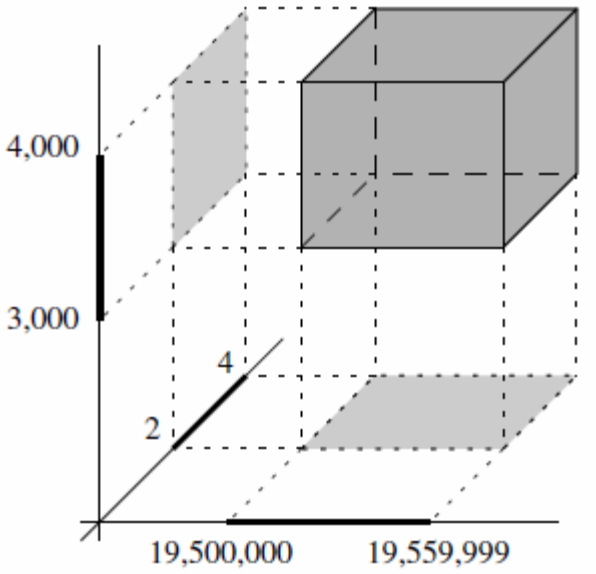
Gesucht:

alle Punkte, die in dem Rechteck oder Würfel liegen



Rechteck: 2-dimensionale Bereichsanfrage

Würfel: 3-dimensionale Bereichsanfrage



Ähnlichkeit zu Datenbanken

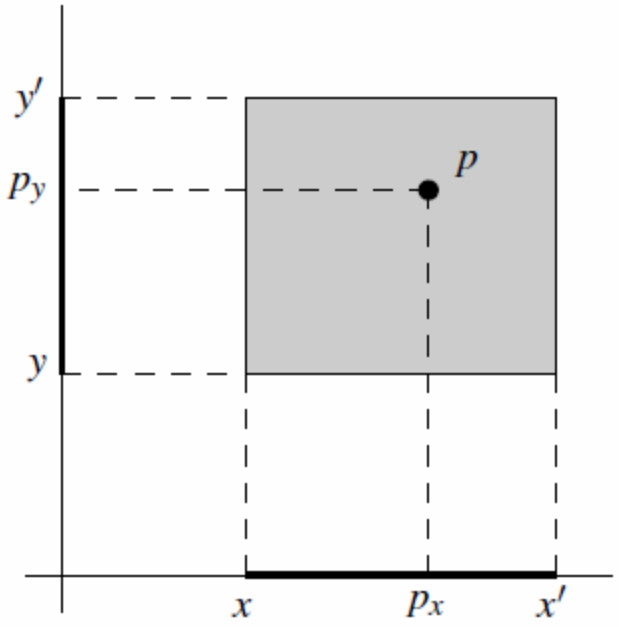
- Bereichsanfragen können geometrisch interpretiert werden
- höher dimensionale Anfragen (> 3) interessant

kd-Baum
2D-Problem

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Ein Punkt $p = (p_x, p_y)$ liegt in einem Rechteck [x:x‘] x [y:y‘] genau dann, wenn: p_x in [x:x‘] und p_y in [y:y‘] liegt

2D-Problem
Bestimme alle Punkte, die im Rechteck liegen



Einfache Lösung benötigt lineare Zeit:
„prüfe für jeden Punkt, ob er in dem Rechteck liegt“.

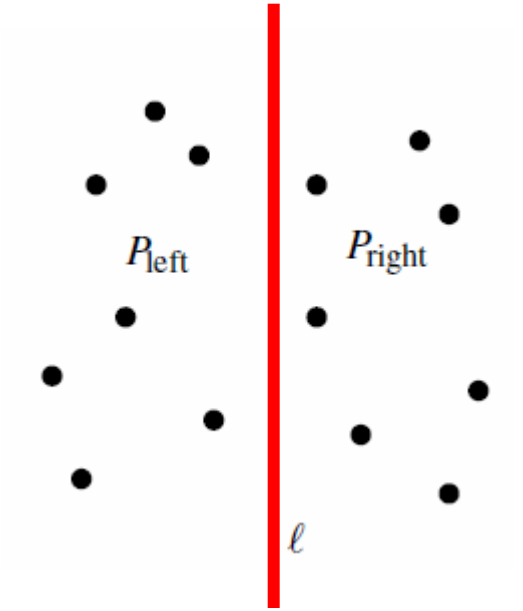
Mit einem kd-Baum können wir die Anfrage schneller beantworten.

kd-Baum
2D-Baumstruktur

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

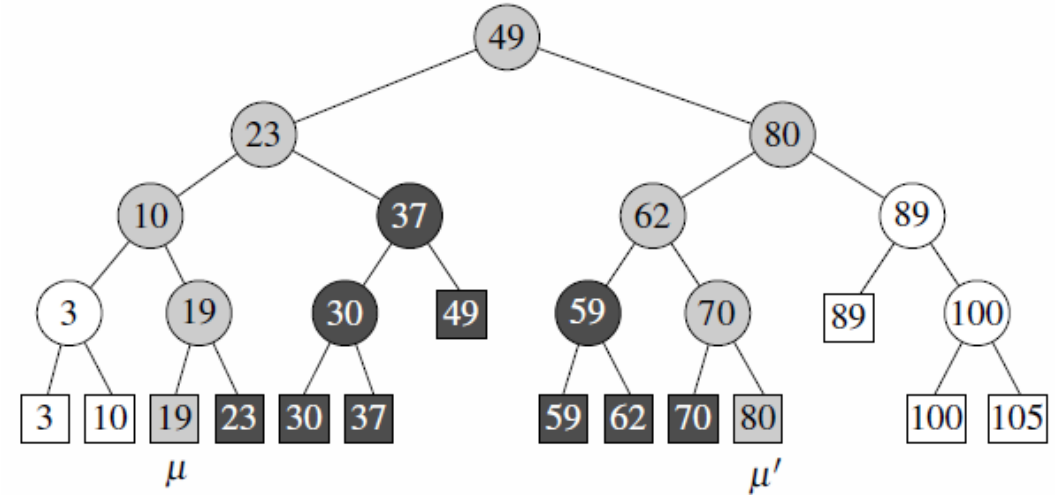
kd-Baum:

- an jedem Knoten teilen wir mit einer Splittinggeraden die Punkte in zwei gleich große Mengen, abwechselnd in x/y-Richtung
- die Splittinggerade wird im Knoten des kd-Baums gespeichert
- die beiden Mengen links/rechts (oben/unten) der Splittinggeraden speichern wir rekursiv in den Kindern



Analogie zu einem binären Suchbaum

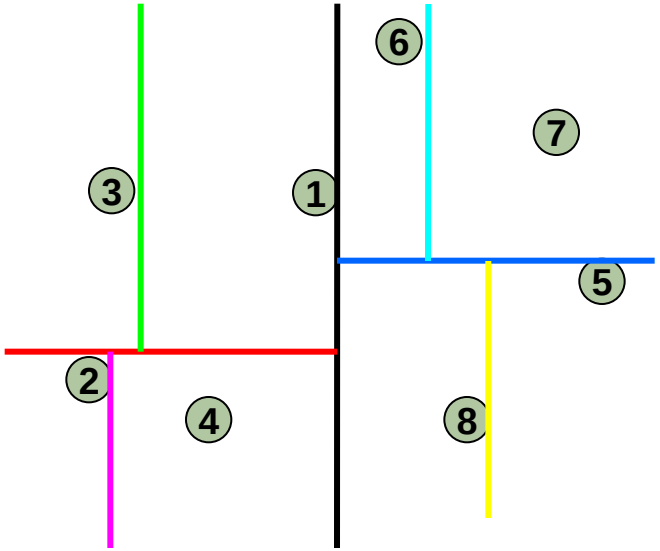
- an jedem Knoten teilen wir die Punkte in gleich große Mengen
- den Splitt-Wert speichern wir in der Wurzel
- die beiden Mengen speichern wir rekursiv in den Kindern

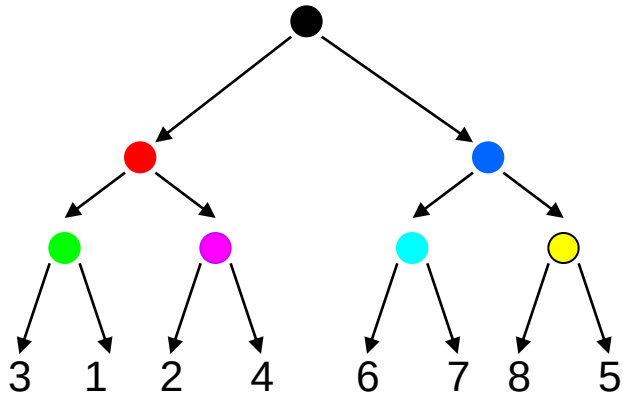


kd-Baum

2D-Baumstruktur

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide





Unterteilung bzgl. x

Unterteilung bzgl. y

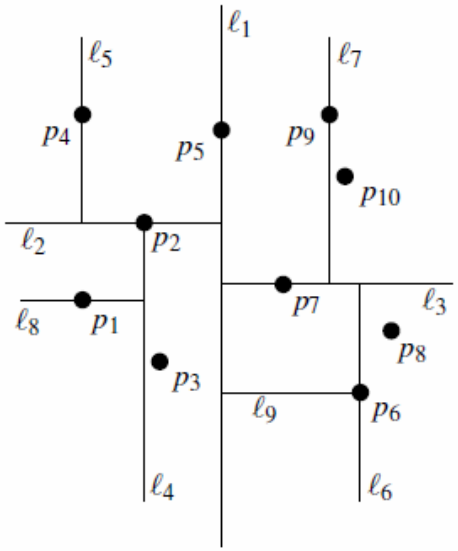
Unterteilung bzgl. x

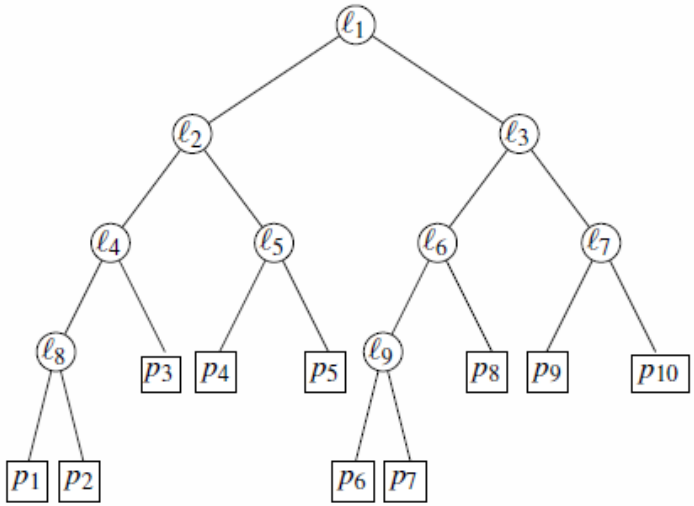
....

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Eigenschaften eines kd-Baums

- Binärbaum
- Splitting-Geraden sind achsenparallel (x/y-Achse)
- Teilmengen sollen etwa gleich groß sein
- von Level zu Level wechselt die Unterteilungsachse (x/y-Richtung)
  - auf gerader Baumtiefe: vertikal
  - auf ungerader Baumtiefe: horizontal
- Punkte werden ausschließlich in den Blättern gespeichert





kd-Baum
2D-Konstruktion

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Eingabe: 2-dimensionale Punktemenge P

Ausgabe: Wurzel eines kd-Baums

Algo BuildKDTree(Set of Points P, depth)

if P enthält nur einen Punkt p

    return Blattknoten der p speichert

else

    if depth gerade

        then Teile P mit vertikaler Linie l durch den Median bzgl. x-Koordinate in Mengen P1, P2

        else Teile P mit horizontaler Linie l durch den Median bzgl. y-Koordinate in Mengen P1,P2

        vleft = *BuildKDTree(P1, depth+1)*

        vright = *BuildKDTree(P2, depth+1)*

        Erzeuge Knoten v mit linkem Nachfolger vleft, rechtem Nachfolger vright und Splittinggerade l

return v

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Analyse

- Welche Zeit und welchen Platz benötigen wir zum Aufbau eines kd-Baums von n Punkten?

Aufwendiger Schritt: Bestimmung der Splittinggeraden

- erfordert die Bestimmung des Median von n Elementen (jeweils x/y Koordinate)
- Medianbestimmung in $O(n)$ Zeit möglich, aber Verfahren sind kompliziert

In der praktischen Implementierung

- sortiere einmal alle Punkte vor der Berechnung des Baumes in zwei Listen, jeweils für x- und y-Koordinate
- für jede Teilung bilde die entsprechenden sortierten Teillisten, Durchlauf in linearer Zeit möglich, Sortierung dabei aufrecht erhalten!
- Medianbestimmung geht dann in $O(1)$ und Erzeugung der Teillisten in $O(n)$

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Aufbau des kd-Baumes lässt sich durch eine Rekurrenzgleichung für die Anzahl der Aufrufe $T(n)$ beschreiben:

$$T(n) = \begin{cases} O(1) & ,n = 1 \\ O(n) + 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) & ,n > 1 \end{cases}$$

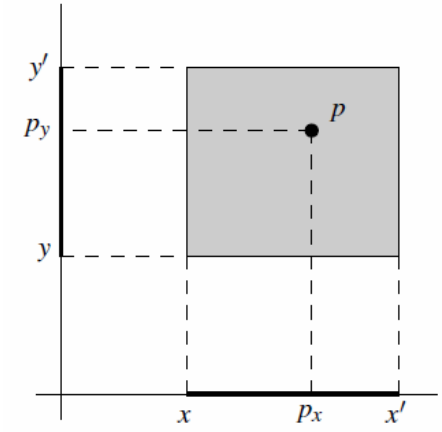
$$\rightarrow T(n) = O(n \log n)$$

Ergebnis: Ein kd-Baum für n Punkte benötigt

- $O(n)$ Platz und kann in
- $O(n \log n)$ Zeit konstruiert werden

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Wie führen wir eine Bereichssuche für ein Rechteck mit dem kd-Baum durch?



Idee:

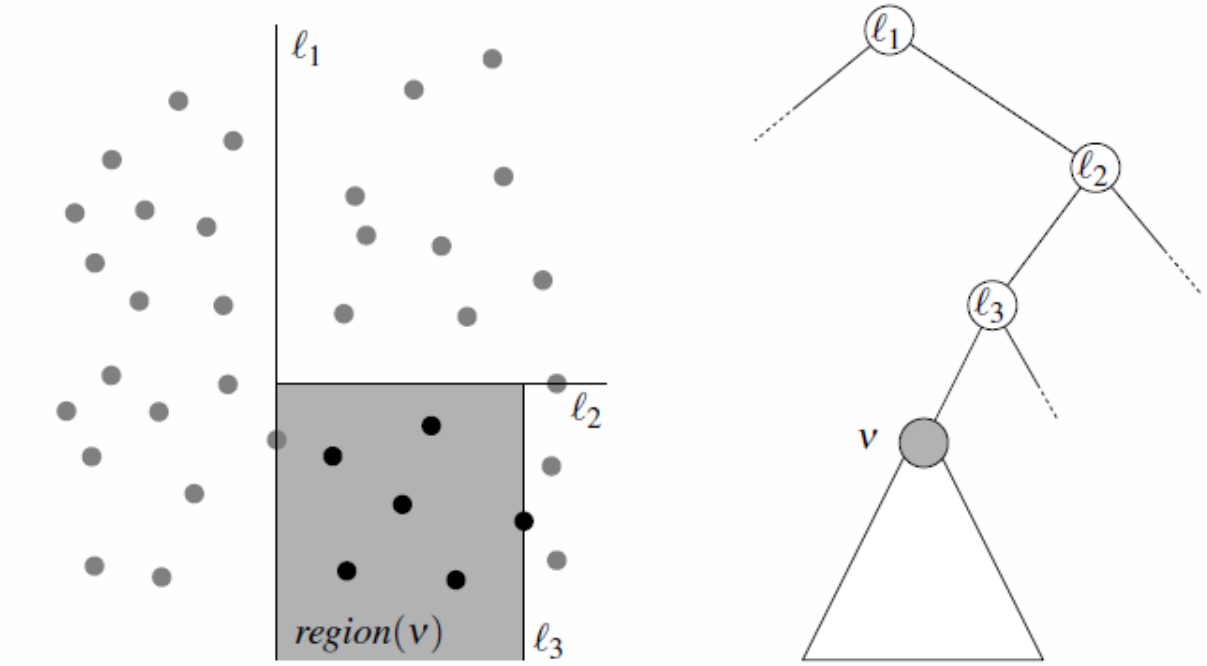
Durchlaufe den Baum und überprüfe,
welche Regionen sich mit dem Rechteck überschneiden oder im Rechteck enthalten sind.

Was sind Regionen?

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Was sind Regionen?

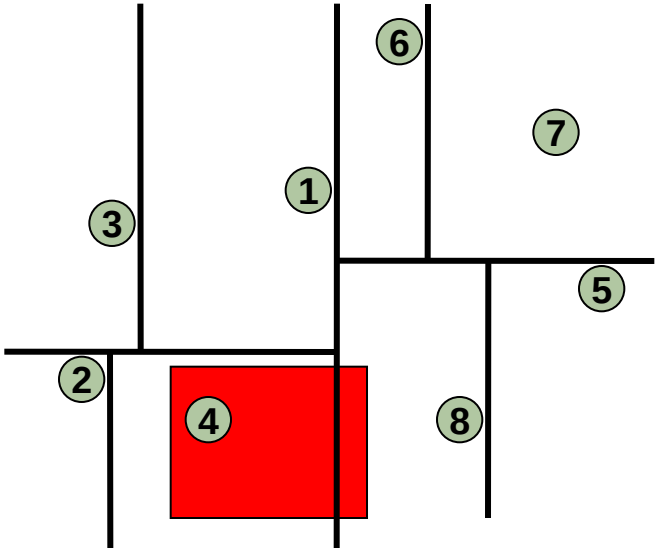
- linker Teilbaum entspricht linker(unterer) „Halbebene“ und rechter Teilbaum entspricht rechter(oberer) „Halbebene“
- es entstehen geschlossene oder an mehreren Seiten offene Regionen (Rechteck)
- die Begrenzung einer Region $region(v)$ erfolgt durch Splittinggeraden, die Vorfahren des Knotens v sind

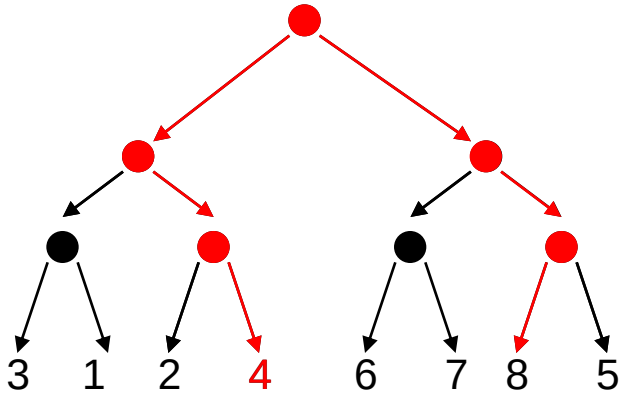


HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Berechnung der Bereichsanfrage

Wir starten an der Wurzel und prüfen,
ob sich Regionen mit dem roten Rechteck überschneiden





Ausgabe: 4

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

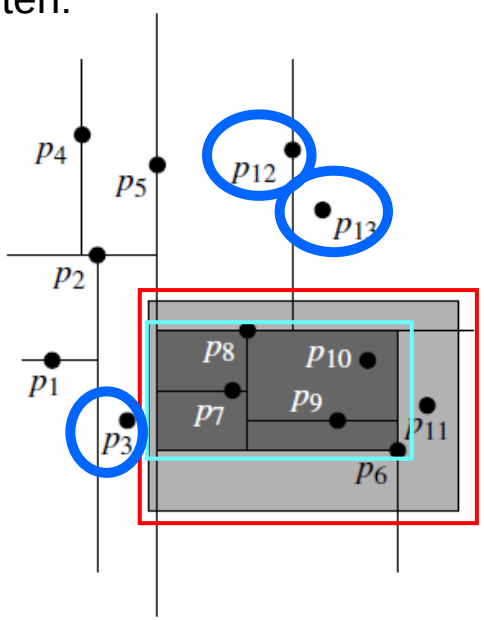
Weiteres Beispiel

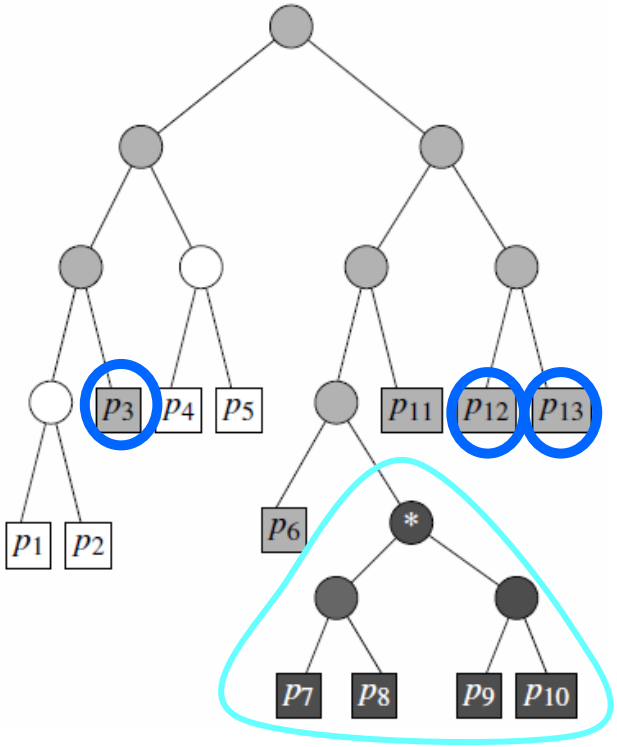
Schwarz: Region des Teilbaums ist vollständig im Rechteck

- alle Punkte ausgeben: p_7, p_8, p_9, p_{10}

Grau: Region überschneidet sich mit Rechteck

- Knoten im Blatt mit der Region testen:
  p_6, p_{11} im Rechteck,
  p_3, p_{12}, p_{13} nicht im Rechteck

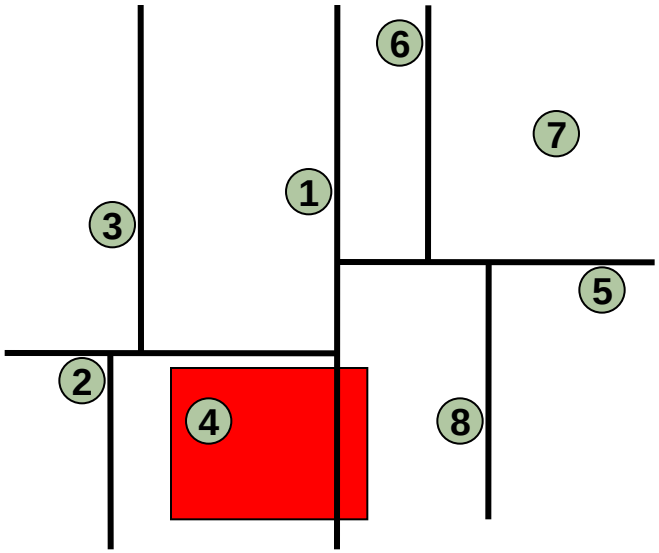




HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Algo SearchKDTree(Knoten v, Bereich R)

if v ist Blattknoten
  then
    wenn p(v) in R gib p(v) aus
  else
    if region(lc(v)) vollständig in R
      then $ReportSubTree(lc(v))$
      else **if** region(lc(v)) schneidet R
        then SearchKDTree(lc(v), R)
    if region(rc(v)) vollständig in R
      then $ReportSubTree(rc(v))$
      else **if** region(rc(v)) schneidet R
        then SearchKDTree(rc(v), R)

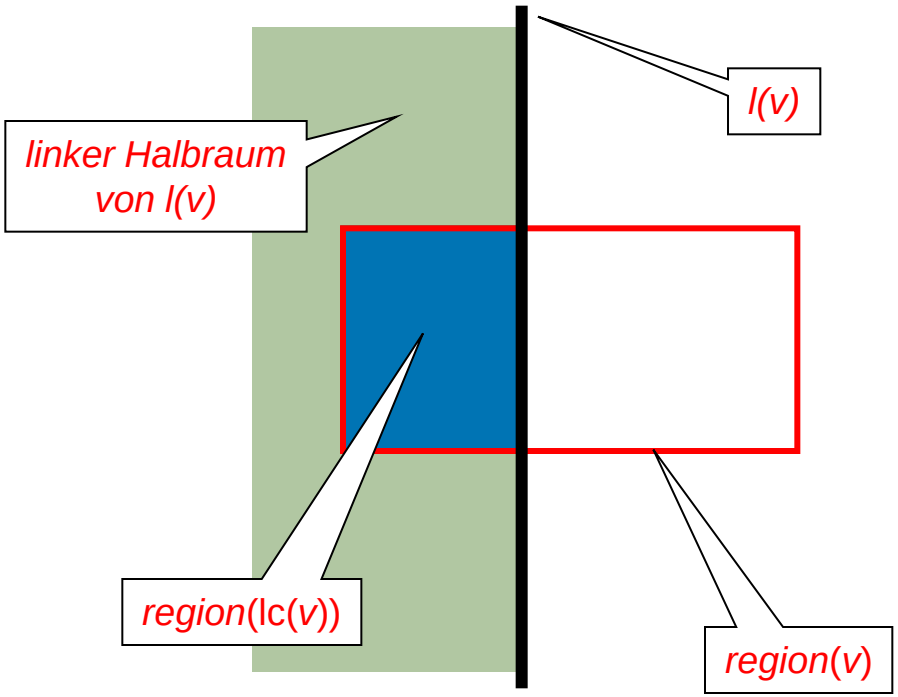


- ***ReportSubtree(v)*** := Gib alle Punkte des Teilbaums aus (vollständige Teilbaumtraversierung)
- **lc(v), rc(v)** := linkes, rechtes Kind von v
- **p(v)** := der im Knoten v gespeicherte Punkt.

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Berechnung der Regionen:

- entweder in Preprocessing für alle Knoten des kd-Baums,
- oder Berechnung von $region(lc(v))$ während der Traversierung mit Hilfe der im Knoten gespeicherten Splittinggeraden $l(v)$.
- $region(lc(v)) := region(v) \cap$ "linker Halbraum der Splittinggerade"



HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Laufzeitanalyse

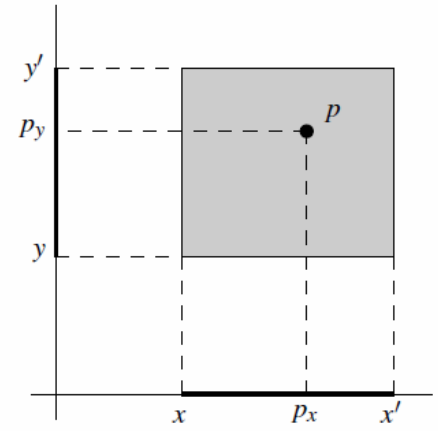
Sei k die Anzahl alle Punkte im Rechteck $[x\colon x'] \; x \; [y\colon y']$

Wie lange benötigt **SearchKDTree(..)** zur Berechnung der k Punkte, die im Rechteck liegen?

→ $O(n)$

Richtig, aber das Ergebnis ist nicht aufschlussreich:

auch eine lineare Suche ergibt Laufzeit $O(n)$



Definition

Ein Algorithmus ist output-sensitiv,
wenn seine Laufzeit von der Größe der Ausgabe abhängt.

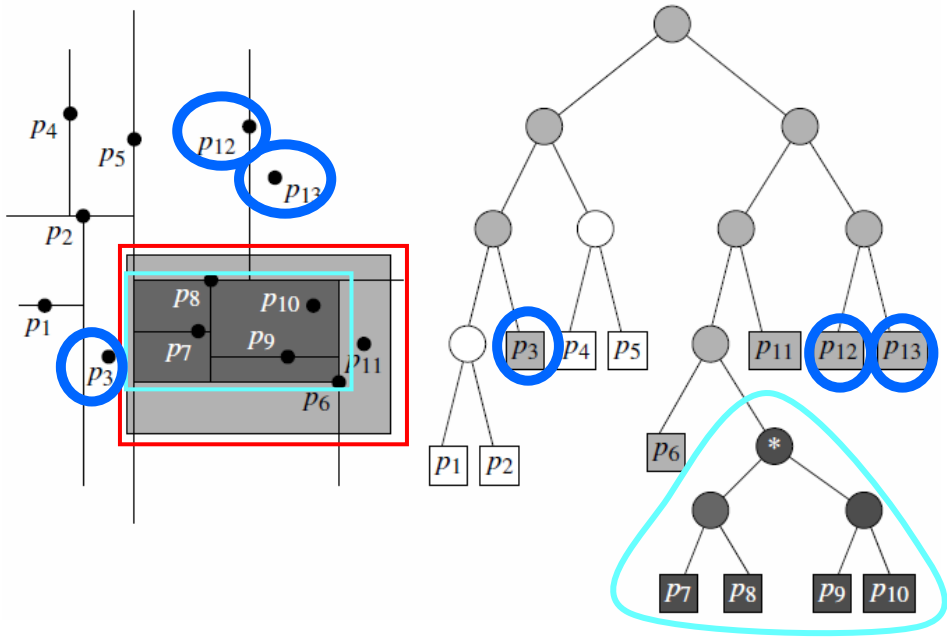
HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Laufzeit

Eine achsenparallele Bereichsanfrage (Rechteck) wird mit einen kd-Baum für n Punkte in Zeit $O(\sqrt{n}+k)$ beantwortet;
k ist die Anzahl der zurück gegebenen Punkte.

Analyse

- alle **ReportSubTree()** Aufrufe ergeben in der Summe einen Teil der Ausgabe $O(k)$ (schwarze Knoten im Beispiel)
- welche weiteren Traversierungsschritte müssen wir noch zählen?
- → **SearchKDTree(...)** - Aufrufe ! (graue Knoten im Beispiel)



HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Wie viele weitere kd-Baum-Knoten besucht die Traversierung?

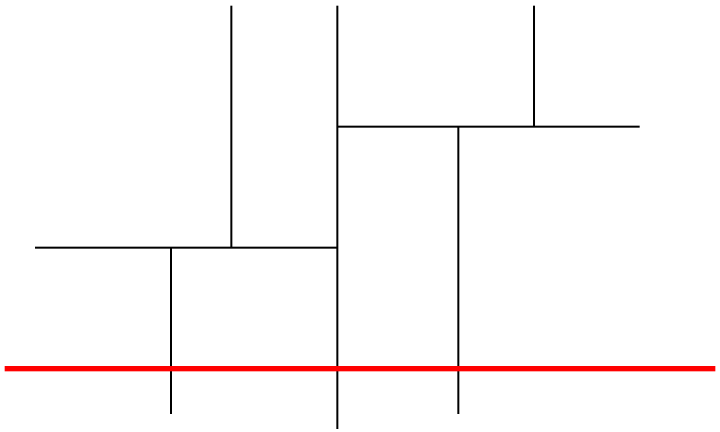
- Anzahl der SearchKDTree(...) – Aufrufe
- für jeden dieser Knoten überlappen sich das Rechteck und region(v),keine vollständige Überdeckung wie bei ReportSubTree(..)
- wir müssen diese Regionen zählen

Problem

- Rechteckgröße und Lage ist frei wählbar, Regionen zählen wird schwierig

Stattdessen

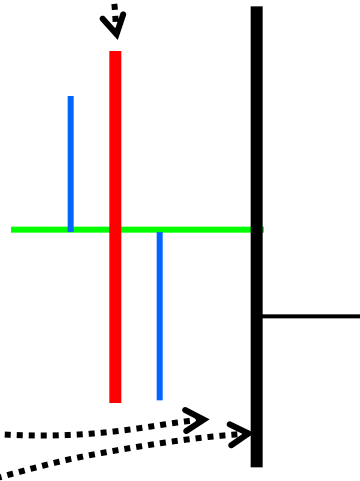
- wir zählen, wie viele Regionen eine unendlich lange Gerade schneidet
- stellvertretend für eine Seite des Anfragerechtecks
- das führen wir für jede der 4 Seiten des Anfragerechtecks aus und erhalten den gesamten Aufwand



HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

Frage: Wie viele Regionen schneidet eine vertikale Gerade?

Antwort:
Sie schneidet entweder die linken oder die rechten Regionen einer vertikalen Splittinggeraden, aber nie beide



Anzahl geschnittener Regionen

$T(n) :=$ Anzahl geschnittener Regionen eines kd-Baums, dessen Wurzel eine **vertikale** Splittinggerade enthält

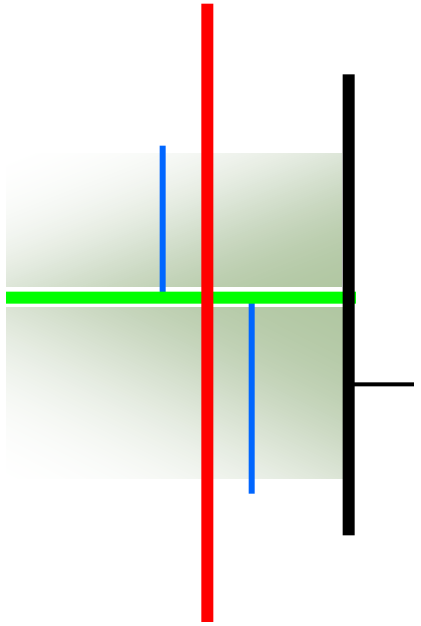
Problem:

- eine Rekursionsstufe tiefer finden wir eine horizontale Splittinggerade
- die Ausgangssituation (= vertikale Splittinggerade) trifft nicht mehr auf $T(n)$ zu

kd-Baum
2D-Bereichsanfrage

HEINZ NIXDORF INSTITUT
Universität Paderborn
Algorithmen und Komplexität
Prof. Dr. Friedhelm Meyer auf der Heide

$T(n) :=$ Anzahl geschnittener Regionen eines kd-Baums, dessen Wurzel eine vertikale Splittinggerade enthält

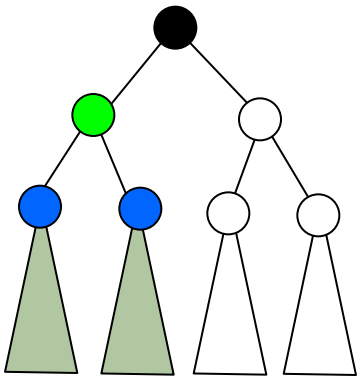


Lösung

- im Baum zwei Ebenen tiefer gehen
- es kommt ein Knoten mit vertikaler Splittinggeraden

Was ergibt sich für $T(n)$?

- $2T(\frac{n}{4})$:
  steht für die 2 Regionen links von der Splitting-geraden, die die vertikale Gerade schneidet;
  jede Region enthält $\frac{n}{4}$ Punkte
- 2:
  Wurzelregion und Region eines Kindes



- $T(n) = \begin{cases} O(1), & n = 1 \\ 2 + 2T\left(\left\lfloor \frac{n}{4} \right\rfloor\right), & n > 1 \end{cases} \rightarrow T(n) = O(\sqrt{n})$