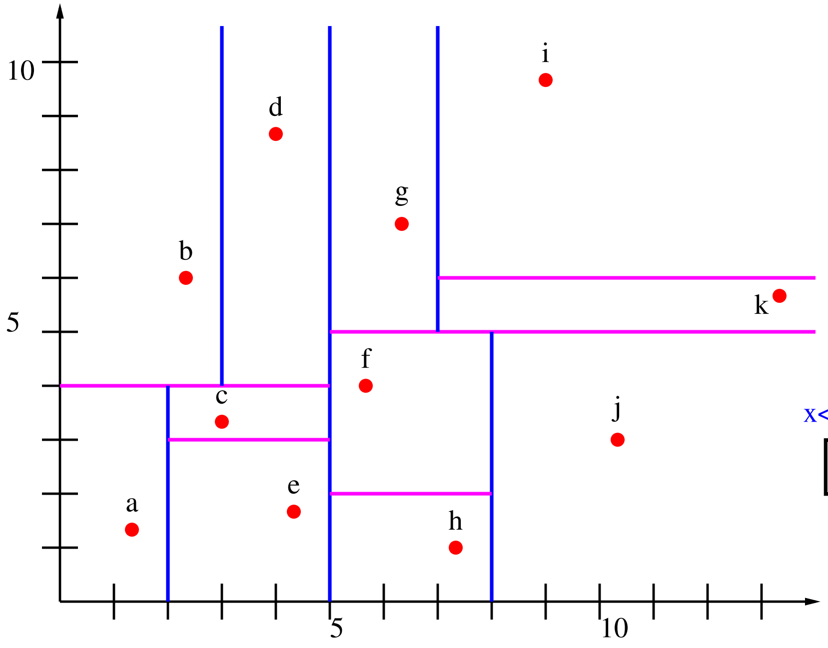
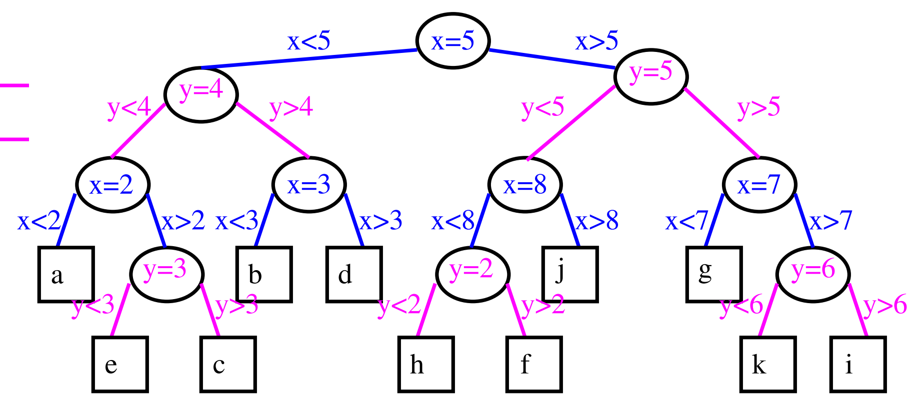
Zusammenfassung: Geom. Datenstrukturen kd-Baum

Elmar Langetepe
University of Bonn

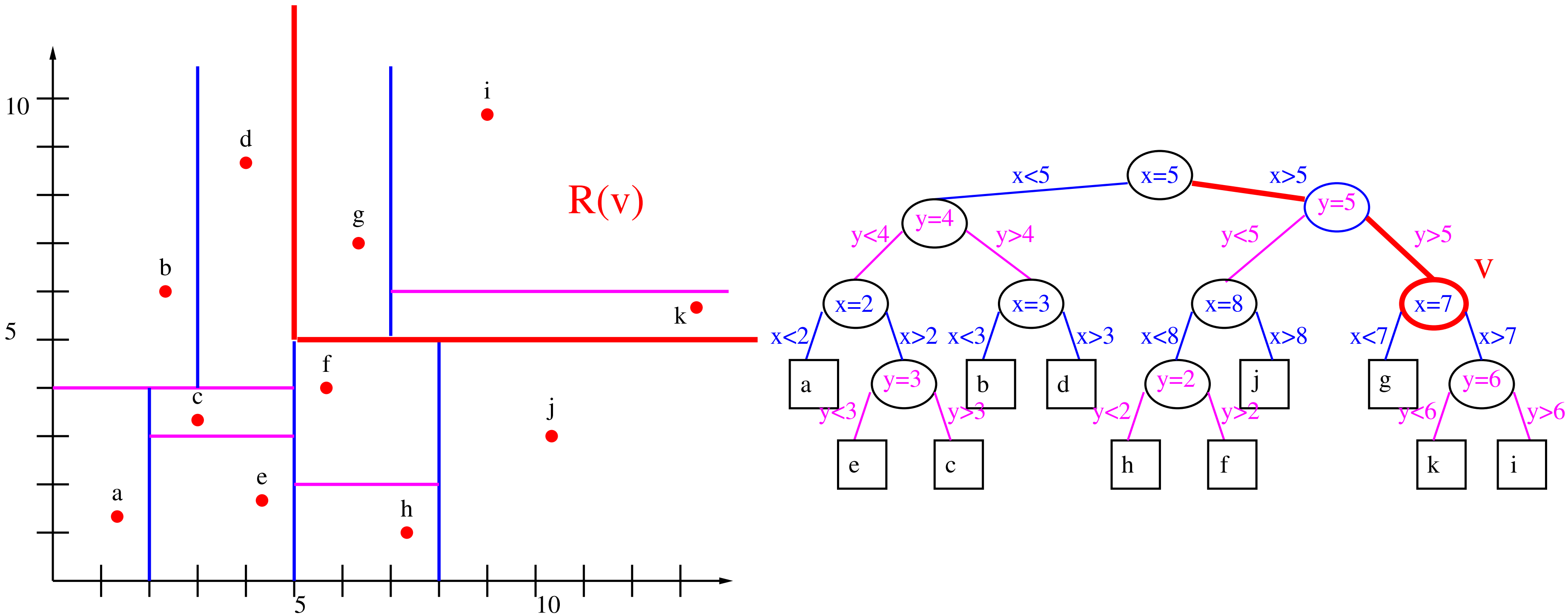
kd-Baum Aufbau

- Splitgeraden $X = s$, $Y = s$ im Wechsel!
- Binärer Baum mit entsprechenden Teilmengen



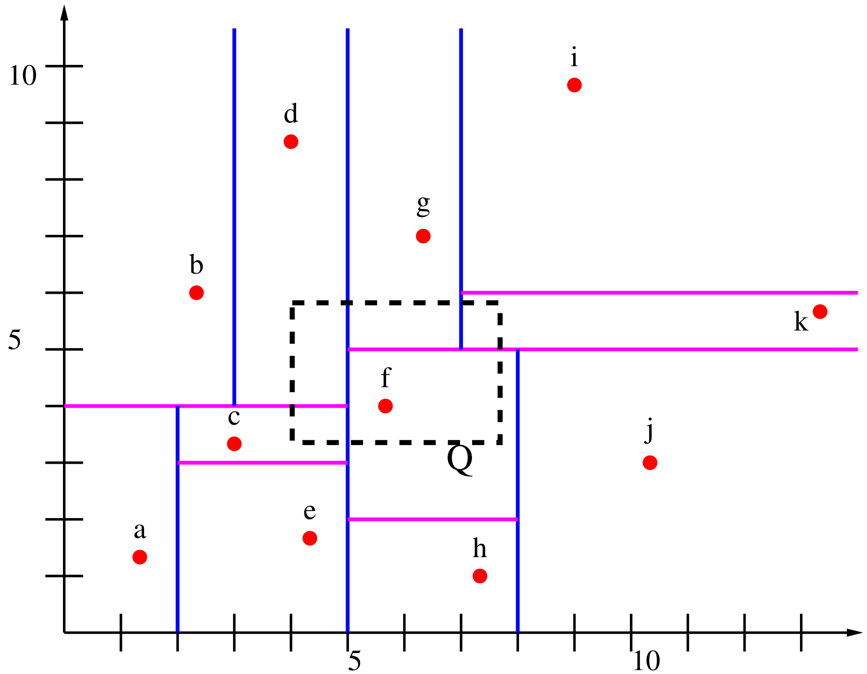


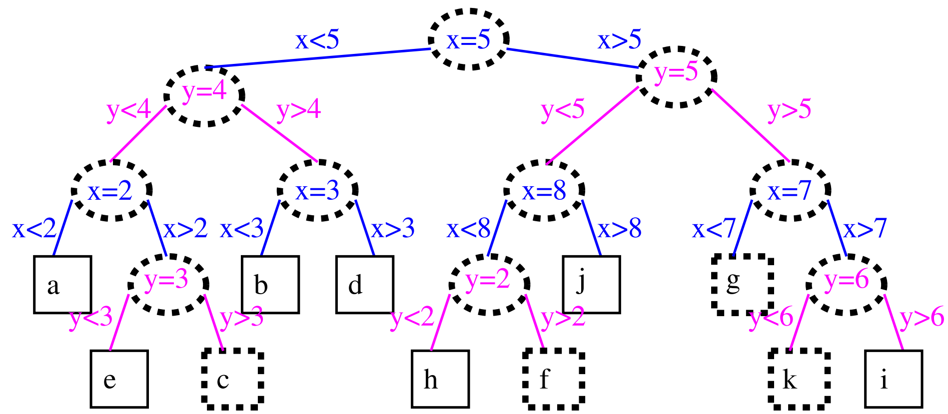
Knoten entspricht Rechteck $R(v)$



Jeder Knoten v entspricht einem Rechteck $R(v)$ als Schnitt von Halbebenen ($X = s$ und $Y = s$).

Anfrage-Query mit Rechteck q





- Bestimme alle Knoten v mit $R(v) \cap q \neq \emptyset$
- Falls v Blatt, teste $v \in q$

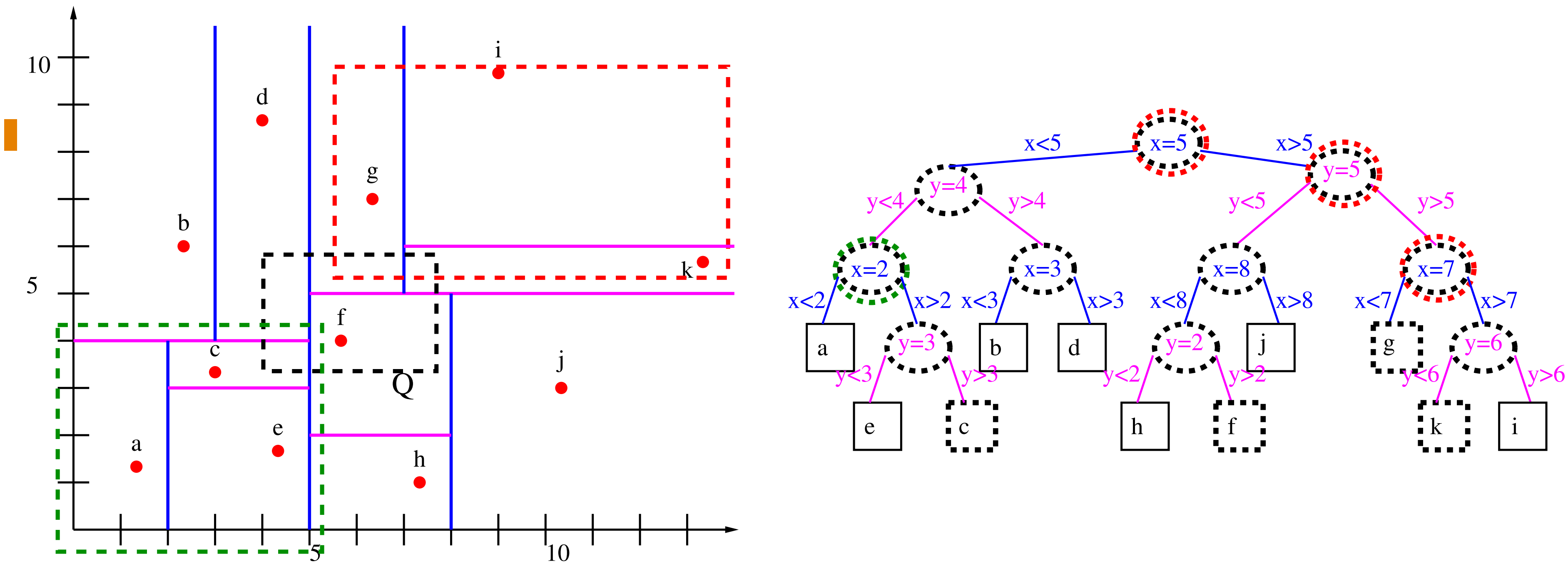
Query Laufzeit

Lemma 3.6: Sei T ein 2d-Baum der Höhe h mit n Punkten. Eine Bereichsanfrage mit achsenparallelen Rechteck Q läßt sich in Zeit $O\left(2^{\frac{h}{2}} + a\right)$ beantworten, wobei a die Größe der Antwort ist.

Beweis:

- Zähle alle Knoten mit $R(v) \cap q \neq \emptyset$
- 1) Knoten v mit $R(v) \subseteq q$: Teilbaum ausgeben $a(v)$
- 2) Knoten v mit $R(v) \not\subseteq q$ (aber $R(v) \cap q \neq \emptyset$)
  $q \subseteq R(v)$: Höhe h viele
  Seite von q schneidet $R(v)$: $2^{\frac{k}{2}}$ viele für Tiefe k (Beweis)
- Aufsummieren über alle Tiefen k: $O\left(2^{\frac{h}{2}} + a\right)$

Visuell: Laufzeit $O\left(2^{\frac{h}{2}} + a\right)$



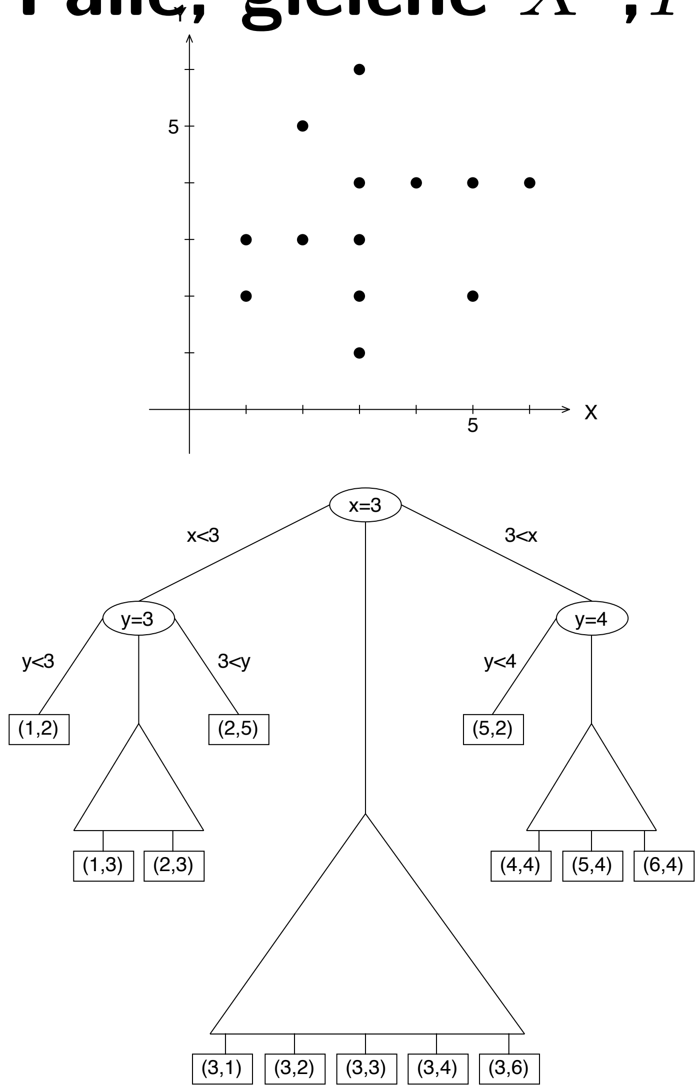
1. Knoten v mit $R(v) \subseteq q$, Teilbaum ausgeben
2. Knoten v mit $R(v) \not\subseteq q$ (aber $R(v) \cap q \neq \emptyset$), zwei Typen

Ergebnis

Theorem 3.7: Ein ausgeglichener 2d-Baum für n Punkte in der Ebene läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz . Eine Bereichsanfrage mit achsenparallelen Rechteck q kann in Zeit $O\left(\sqrt{n} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

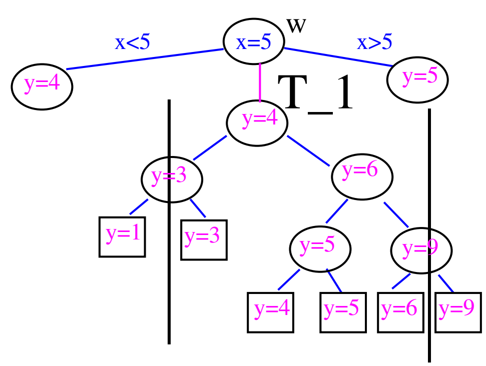
Sortieren nach X und Y.
Rekursiv in gleichgroße Teilmengen aufteilen, lineares Aufteilen.
Höhe ist $O(\log n)$.

Degenerierte Fälle, gleiche X-,Y-Koordinaten



Query Laufzeitabschätzung

- Zusätzlicher Aufwand für dritte Struktur: $R(w) \cap q \neq \emptyset$
- $R(v)$ in Sohn T_1 ist Intervall: $I(v)$
- 1) $I(v) \subseteq q$: Teilbaum ausgeben $a(v)$
  2) q schneidet $I(v)$: Höhe von T_1 viele
- Anzahl Blätter T_1 (für Knoten w Tiefe k): $\leq \frac{n}{2^{k-1}}$
- $n = 2^h$: $\sum_{k=1}^{h} 2^{\frac{k}{2}} \log 2^{h-k+1} \in O(2^{\frac{h}{2}})$



q mit Y=7 und Y=2

Ergebnis entartete 2d-Bäume

Theorem 3.7 (entartet): Ein ausgeglichener 2d-Baum für n Punkte in der Ebene läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz. Eine Bereichsanfrage mit achsenparallelen Rechteck q kann in Zeit $O\left(\sqrt{n} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

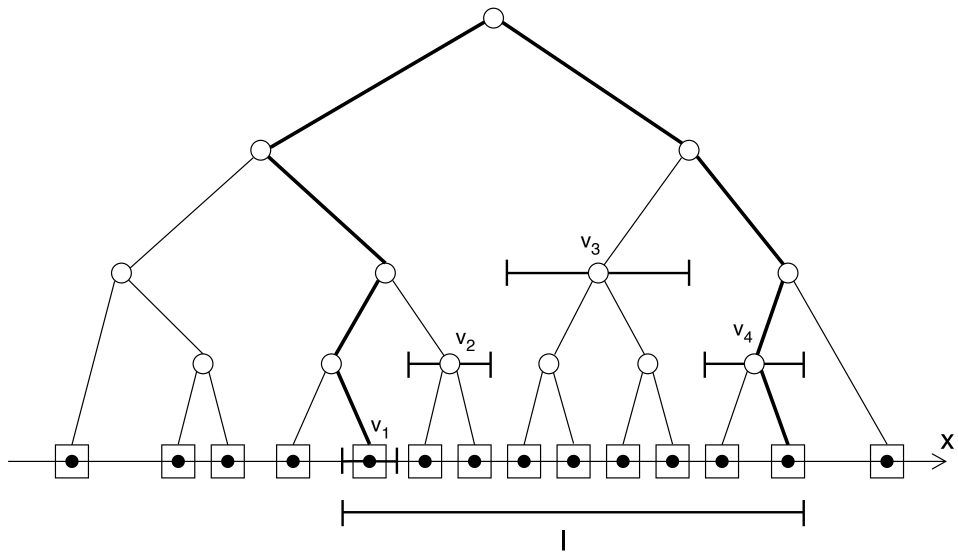
Verallgemeinerung kd-Baum

Theorem 3.10 (entartet): Ein ausgeglichener kd-Baum für n Punkte im $\mathbb{R}^k$ läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz. Eine orthogonale Bereichsanfrage kann in Zeit $O\left(n^{1-\frac{1}{k}} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

k zyklische Splitebenen, ternäre Struktur, Datenobjekte in Split(hyper)ebene in $(k-1)d$-Baum ablegen

kd-Baum: Platzoptimal, Laufzeitverbesserungen Query: $O(a + \log^2 n)$

Range Tree (Bereichsbaum)



Eindimensionaler Bereichsbaum: Intervalle die ein Anfrageintervall ausschöpfen

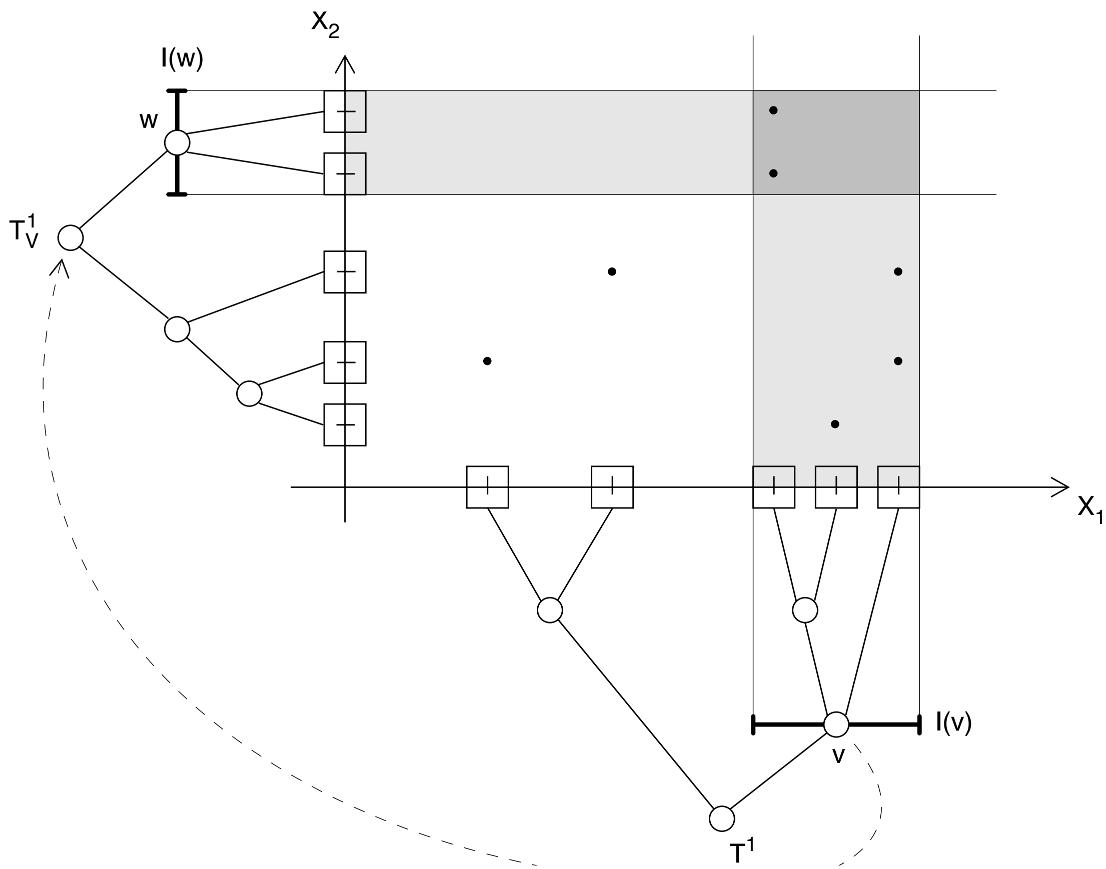
Jeder Knoten beschreibt ein Intervall

Knoten min. Höhe, Intervall disjunkt überdecken: $O(\log n)$ viele

Rekursiv k-dimensionaler Range Tree T^k

- Rekursiv Bereichsbaum T^k mit Dim. k
- T^1 für $D^1 = \{x_1 | (x_1, x_2, \ldots, x_k) \in D\}$
- Für jeden Knoten v in T^1: Konstr. $(k-1)$-dim Bereichsbaum T_v^{k-1} der Menge $D_v^{k-1} = \{(x_2, x_3, \ldots, x_k) | x_1 \in I(v)\}$
- Zeiger bei v zeigt auf T_v^{k-1}
- Letzter Baum enthält im Blatt alle Knoten aus D

Beispiel 2-dimensionaler Range Tree



Query: k-dimensionaler Range Tree

- Hyperrechteck $q \in \mathbb{R}^k$: $q = (I_1, I_2, \ldots, I_k)$
- $k = 1$: Beantworte die Frage im eindim. Suchbaum
- Sonst: Bestimme Knoten $v_1, \ldots, v_l$ in T^1:
  Intervalle $I(v_i)$ schöpfen zusammen I_1 aus
  $v_1, \ldots, v_l$ haben minimale Höhe mit dieser Eigenschaft
- Für $1 \leq i \leq l$ beantworte die Anfrage $q = (I_2, \ldots, I_k)$ für $T_{v_i}^{k-1}$
- Korrekt: Zuerst unter X_1-Koordinaten suchen
  Von diesen unter X_2-Koordinaten suchen
  Uswusf.

Buch Kapitel

Kapitel 3.3.1 Seite 126 oben – S. 134 unten

Kapitel 3.3.2 Seite 135 oben – S. 138 oben