AlgGeo 9.1

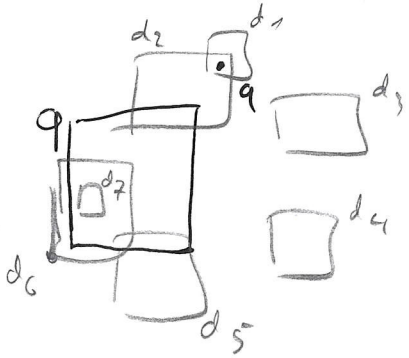
Geometrische Datenstrukturen

	Liste	AVL-Baum
Einfügen	$O(1)$	$O(\log n)$
Entfernen	$\Omega(n)$	$O(\log n)$
Suchen	$\Omega(n)$	$O(\log n)$

geometrische Objekte: 2

n Objekte

Bereichsanfragen



2 Typen

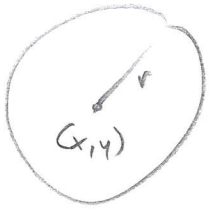
- Inklusionsanfrage → d_7
- Schnittanfrage → d_2, d_5, d_6, d_7

Objekte sind Punkte ⇒ Inklusion & Schnitt redundant

⇒ alles als Punkte darstellen

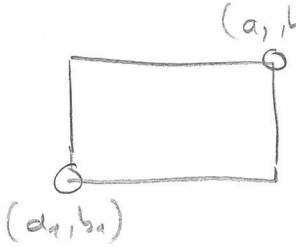
Kreis

z.B.



→ (x, y, r)

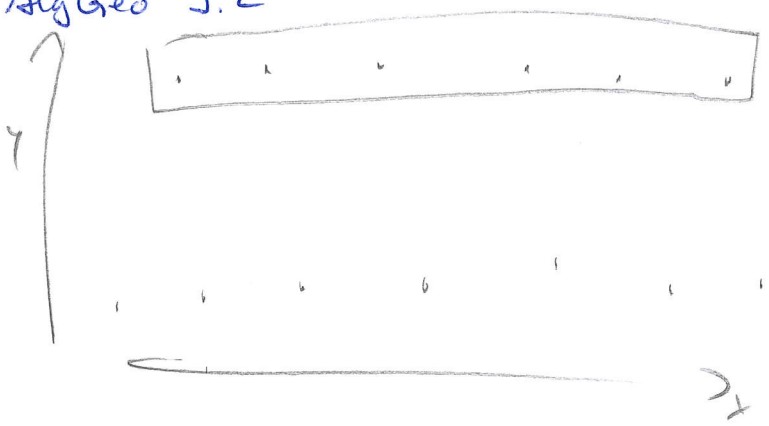
$\mathbb{R}^2$ $\mathbb{R}^3$



→ (a_1, b_1, a_2, b_2)

$\mathbb{R}^2$ $\mathbb{R}^4$

AlgGeo 9.2



→ AVL-Baum nach x-Achse

→ Fast alle Knoten müssen besucht werden

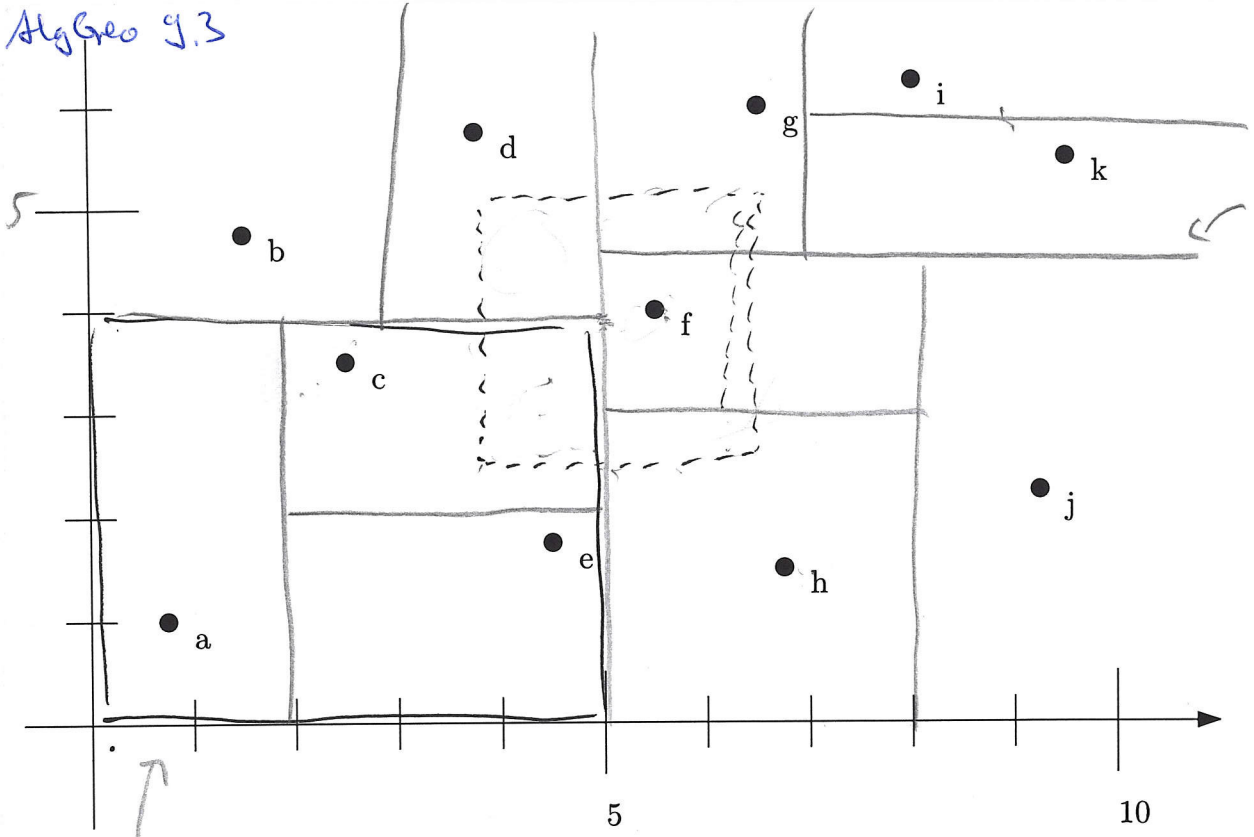
Grundsätzlich gibt es Kompromisse zwischen statischen & dynamischen DS

↗ 1x aufbauen, oft suchen

↑ suchen, einfügen, entfernen in Kombination

Kapitel 3.2

→ ∀ ~ jede statische DS kann zu einer dynamischen DS gemacht werden mit zusätzlichen Aufwand kostenfaktor log n



k-d-Baum

erstmal $\mathbb{R}^2$, allg. $\log$

Splitgerade

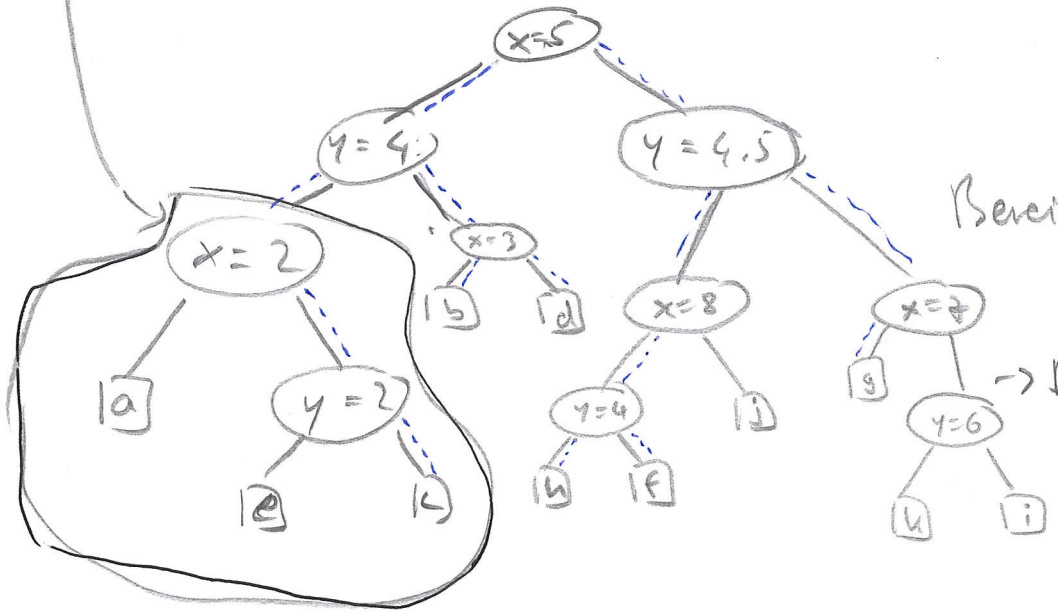
$y = 4{,}5$

↑ Splitkoordinate

innere Knoten ≙ Splitgeraden

Blätter ≙ Punkte

Teilbäume an Knoten v ≙ Rechteckbereiche $R(v)$



Punktsuche → Standardsuche im Baum

$O(\log n)$

Bereichsanfrage: achsenparalleles Rechteck

$q = [4;5{,}5] \times [2{,}5;5]$

→ Besuche alle Knoten v mit $R(v) \cap q \neq \emptyset$

→ Beispiel: fast alle inneren Knoten und 5 der Blätter

k-d-Baum Bereichsanfragen

Lemma 3.6.

Sei T ein 2-d-Baum mit Höhe h mit n Punkten.

$\Rightarrow$ achsenparallele Bereichsanfragen in Zeit $O(2^{h/2} + a)$

(a: Ausgabegröße.)

Beweis

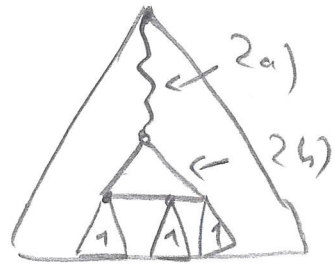
Besuchte Knoten haben einen von zwei Typen

1) v mit $R(v) \subseteq q$ ← alle Kinder, insb. Blätter sind auch Typ 1) und alle Punkte in den Blättern sind $\in q$

2) v mit $R(v) \subseteq q$ aber $R(v) \cap q \neq \emptyset$ → in $O(a)$ durchlaufen

2a) $q \subseteq R(v)$ → Aufrufspfad in T höchstens h viele

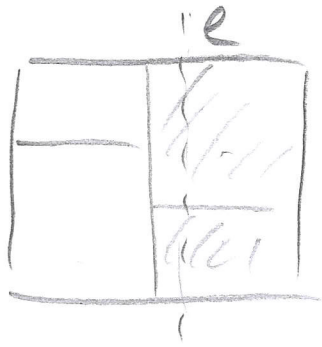
2b) min. 1 Kante ℓ von q schneidet $R(v)$

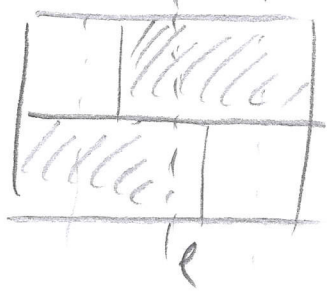


t_k := Anzahl 2b)-Knoten in Tiefe k

$t_1 \leq 1 \quad t_2 \leq 2$

$k \geq 2$ → höchstens 2 von 4 Enkeln eine 2b)-Knotens





$\Rightarrow$ Induktion $t_k \leq 2^{k/2}$

Typ 2 $O\left(h + \sum_{k=1}^{h} 2^{k/2}\right) \Rightarrow O(2^{h/2})$

□

AlgGeo 9.5

ausgeglichenem k^2-d- Baum ist $h = O(\log n)$ $\quad o(\lceil \log n \rceil)$

$\Rightarrow$ Bereichsanfragen $O(\sqrt{n} + q)$

statisch: konstruierbar in $O(n \log n)$

Speicher $O(n)$

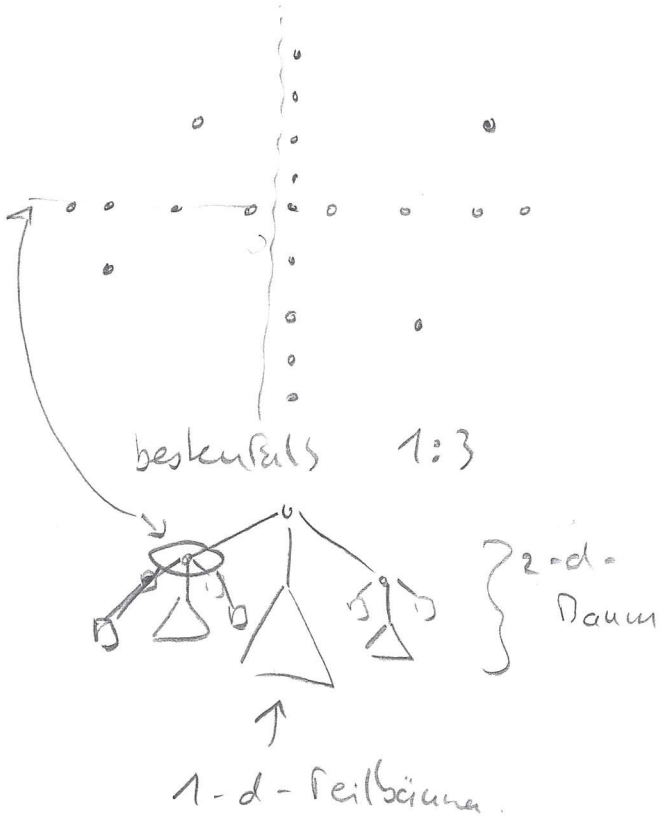
Bereichsanfragen $O(\sqrt{n} + a)$

$\Rightarrow$ dynamisiert $O(\sqrt{n} \log n + a)$

Problem: Allgemeine Lage Annahme -> Beispiel

Idee: Schnittgerade auf die Punkte, sodass links und rechts höchstens $n/2$ Punkte liegen

-> hängen zusätzlichen Kind-Baum mit niedrigerer Dimension an.

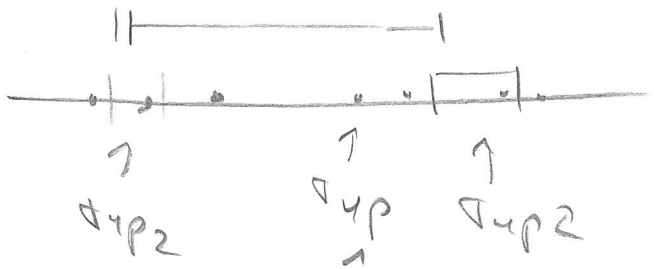


$\Rightarrow$ Semi-Baum

Knoten all $I(v)$ statt $R(v)$; jede

$\Rightarrow$ Typ 2 Knoten nur am Rand

-> nur $O(h)$



-> Besuch der mittleren Bäume auch $O(2^{n/2})$

-> Anwendung für closest-Pair $\mathbb{R}^3$ - Sweep

-> Laufzeit $O(n^{3/2} \log n)$ besser als naiv $O(n^2)$

-> übertragbar in höhere Dimensionen, aber sehr technisch

-> orthogonale Bereichsanfragen in $O(n^{1-\frac{1}{d}} + a)$

Speicher $O(n)$

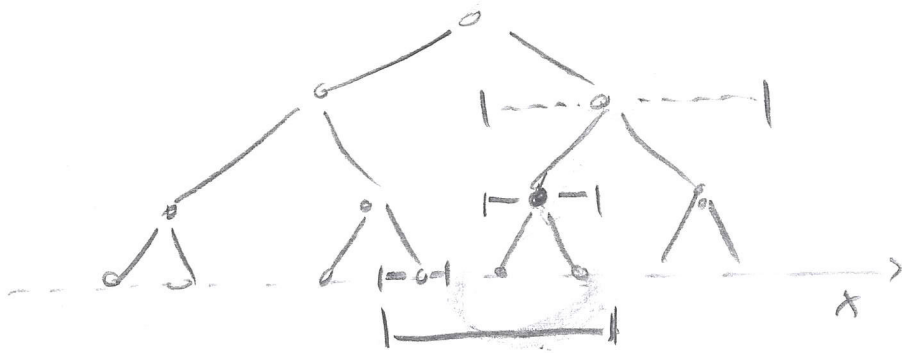
-> was, wenn wir ein bisschen mehr Speicher haben

geht das besser? => Bereichsbaum

ja, aber -> $O((\log n)^2 + a)$ aber $O(n \log n)$

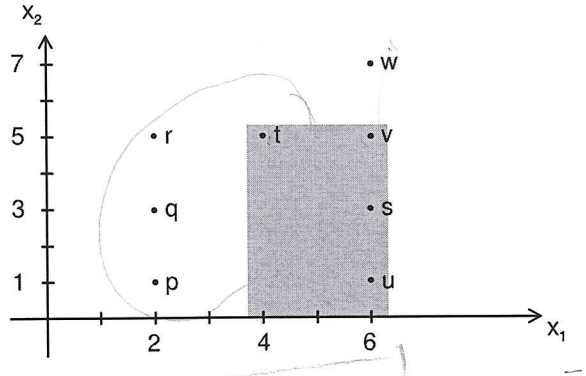
↑ Bereichsanfrage — Speicher

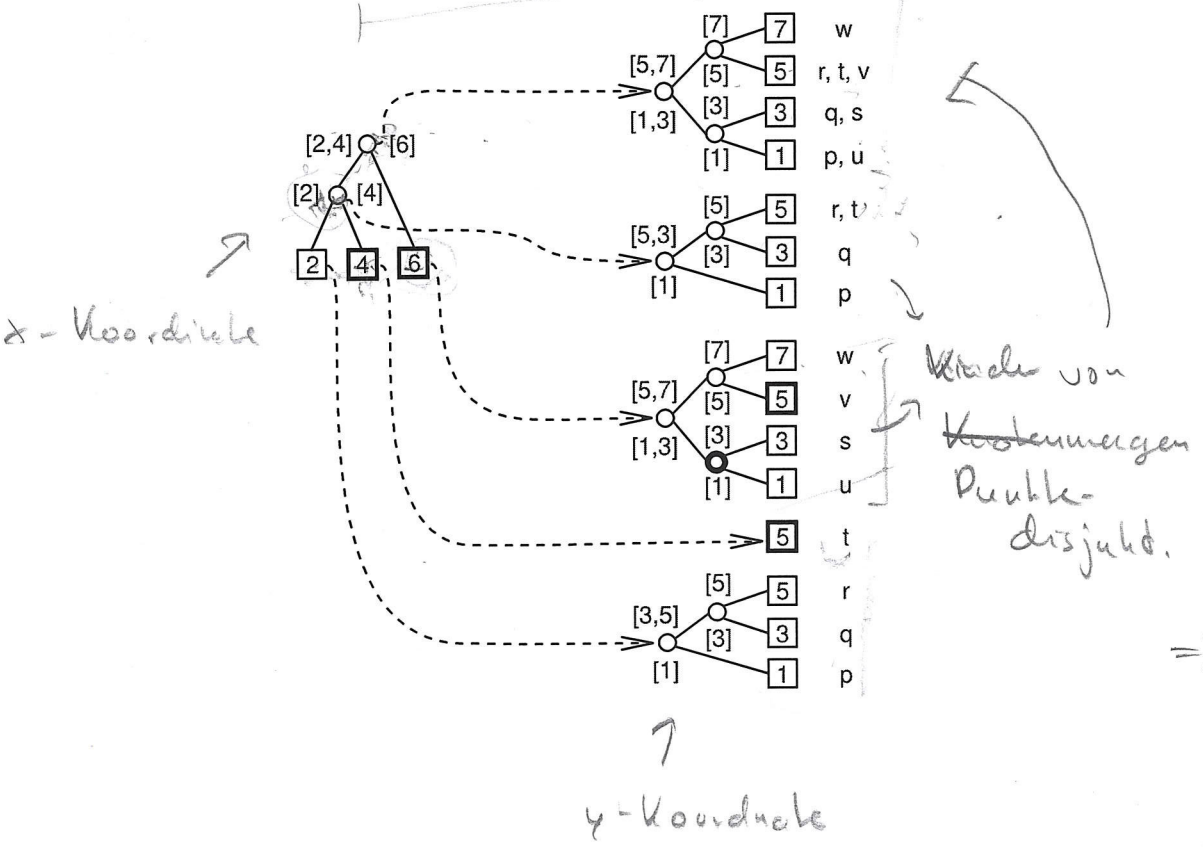
Eindimensionaler Suchbaum -> jeder Knoten v <=> $I(v)$ Intervall



Bereichsanfrage I_q besucht alle "obersten" -> höchstens $O(\log n)$
Knoten v mit $I(v) \subseteq I_q$ viele
und deren komplette Teilbäume

AlgGeo 9.7.





Kinder von Knotenmengen Punkte-disjunkt.

-> Konstruktion k-Bereichsbaum
rek-d. Baum für die
erste Koordinate
-> an jedes Blatt ein
k-1 - Bereichsbaum rekursiv
anhängen mit verbliebenen
Punkten.

Bereichsanfrage.
-> nur die "obersten"
absteigenden Knoten
und in deren rekursiven
Unterbäumen weitersuchen

$O((\log n)^{k-1} \cdot n)$ Speicher

=> jeder Punkt kommt pro Rekursions-
in höchstens $\log n$ Unterbäumen vor

$O(\log(n)^k + a)$ Bereichsanfrage

=> jede Blatt würden höchstens $\log n$
Unterbäume durchsucht.