

# Bereichsanfragen

## 2d-Baum und Quadtree

Tanja Weber, Matthias Jauernig

Fachbereich Informatik, Mathematik und Naturwissenschaften

17.12.2007

- Zeitnahme in C# über `QueryPerformanceCounter`
- Durchschnittswerte über 20 Iterationen
- keine Benutzerinteraktionen mit dem Rechner während der Analyse
- iterative Implementierungen
- Maximaler Einfüge-Wert: 100.000  
(ist gleichzeitig max. Anfrageintervall-Wert)

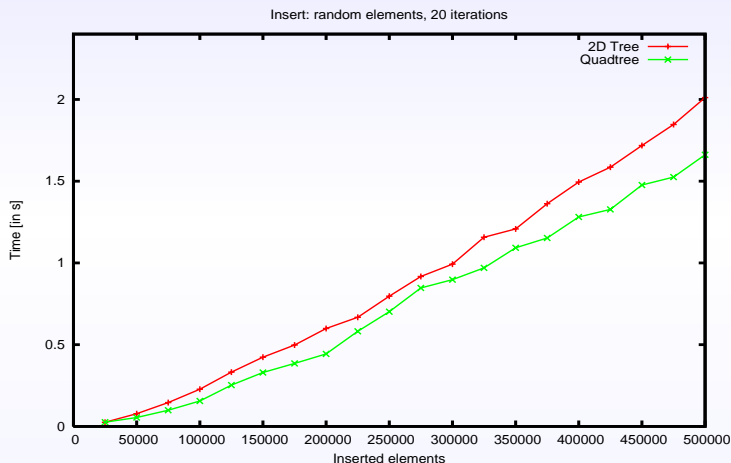
- Zeitnahme in C# über `QueryPerformanceCounter`
- Durchschnittswerte über 20 Iterationen
- keine Benutzerinteraktionen mit dem Rechner während der Analyse
- iterative Implementierungen
- Maximaler Einfüge-Wert: 100.000  
(ist gleichzeitig max. Anfrageintervall-Wert)

Mögliche Variablen:

- Anzahl eingefügter Elemente
- Einfüge-Folge: zufällig, sortiert (Wert 1, Wert 2, beide Werte), ...
- Wertebereich eingefügter Elemente (fest, dynamisch)
- Anzahl an Abfragen
- Länge des Abfrageintervalls (festgelegt, zufällig)
- Start des Abfrageintervalls (festgelegt, zufällig)

# Einfügen, zufällige Elemente

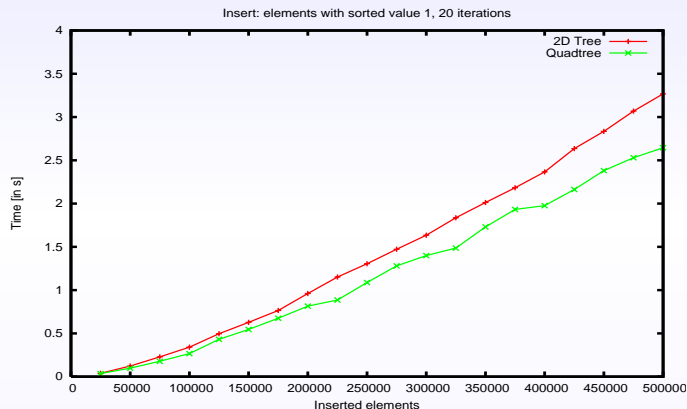
- Benötigte Zeit:



- Auswertung: Quadtree besser, da im Durchschnitt nicht so tief
- Laufzeit:  $O(n \cdot \log(n))$

# Einfügen, nach Wert1 sortierte Folge

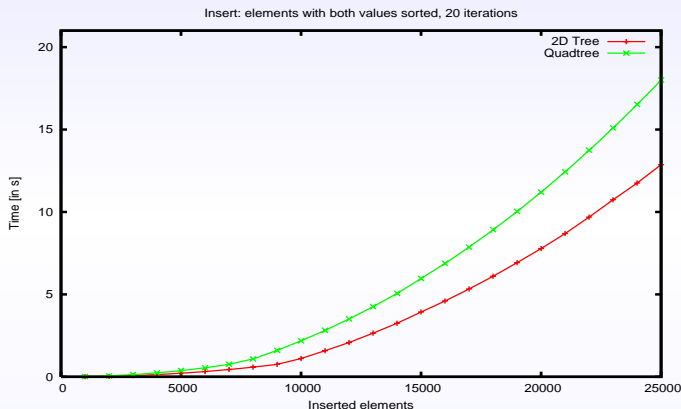
- Benötigte Zeit:



- Auswertung: Quadtree besser, da im Durchschnitt nicht so tief
- Sortierung nach Wert1 irrelevant (aber: schlechtere Zeiten)
- Laufzeit:  $O(n \cdot \log(n))$

# Einfügen, nach beiden Werten sortierte Folge

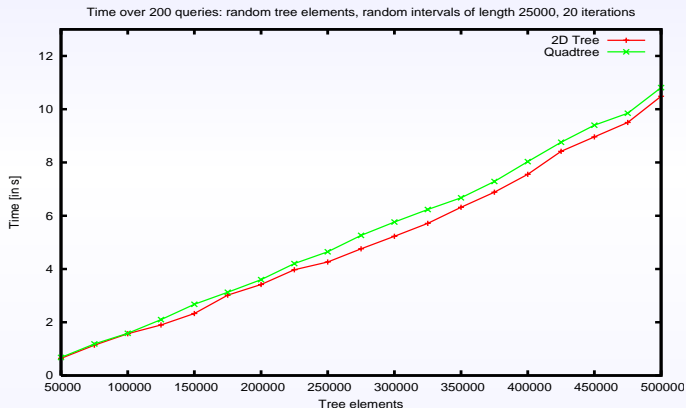
- Benötigte Zeit:



- Auswertung: Quadtree schlechter, da mehr Vergleiche nötig (für alle 4 Kinder) bis nächster Knoten gefunden wird
- Laufzeit:  $O(n*n)$

# 200 Abfragen, Intervalllänge: 25000, Intervallstarts zufällig, zufällige Elemente im Baum

- Benötigte Zeit:

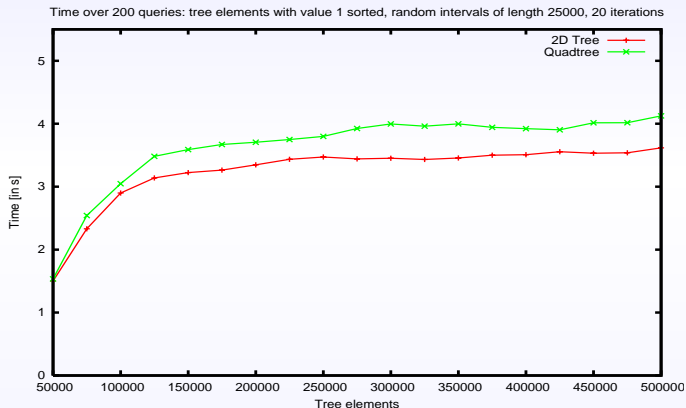


- Auswertung: 2d-Baum etwas besser
- Laufzeit: anscheinend annähernd linear (?)



# 200 Abfragen, Intervalllänge: 25000, Intervallstarts zufällig, nach Wert1 sortierte Folge im Baum

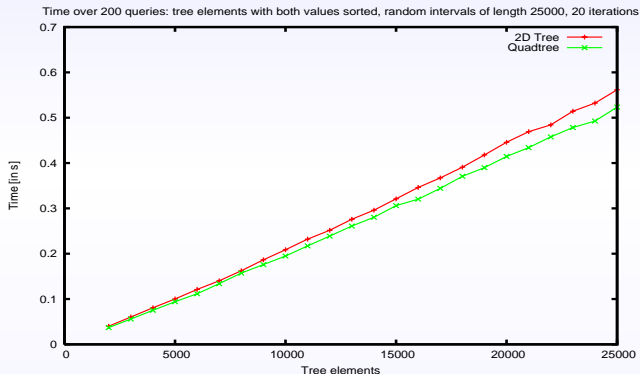
- Benötigte Zeit:



- Auswertung: 2d-Baum besser
- Laufzeit: logarithmisch (?)

# 200 Abfragen, Intervalllänge: 25000, Intervallstarts zufällig, nach beiden Werten sortierte Folge im Baum

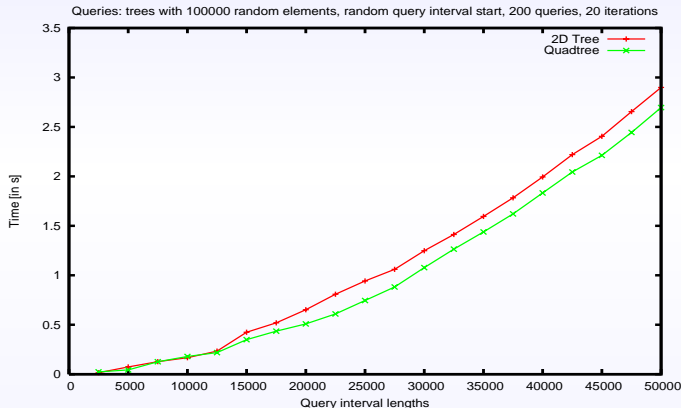
- Benötigte Zeit:



- Auswertung: Quadtree etwas besser, Vermutung: aufgrund von Ebenenverwaltung beim 2d-Baum
- Laufzeit:  $O(n)$

# 200 Abfragen, Intervallstarts zufällig, 100000 zufällige Elemente im Baum

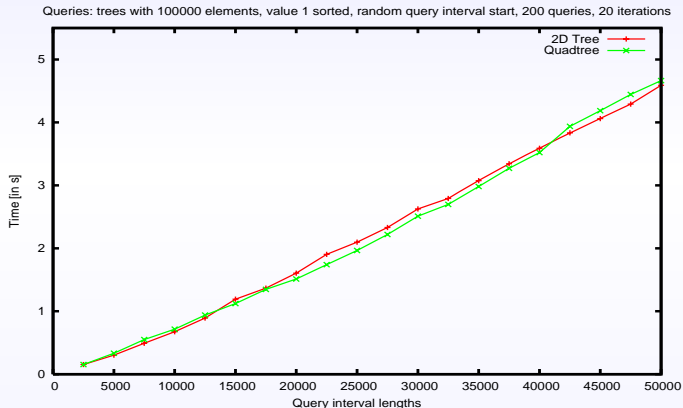
- Benötigte Zeit:



- Auswertung: Quadtree besser
- Laufzeit: überlinear

# 200 Abfragen, Intervallstarts zufällig, 100000 nach Wert1 sortierte Elemente im Baum

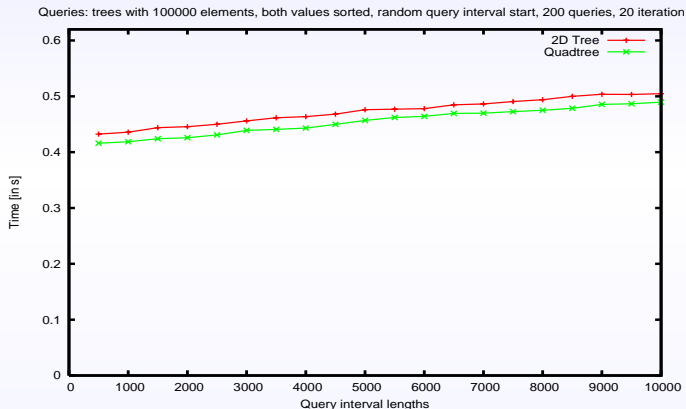
- Benötigte Zeit:



- Auswertung: Kaum Unterschiede
- Laufzeit: überlinear

# 200 Abfragen, Intervallstarts zufällig, 100000 nach beiden Werten sortierte Elemente im Baum

- Benötigte Zeit:



- Auswertung: Quadtree etwas besser
- Laufzeit:  $O(n)$

- Quadtree bei „normalen“ Einfügeoperationen schneller
- 2d-Baum bei „normalen“ Queries schneller
- häufig nur geringe Unterschiede