

Datenstrukturen und Algorithmen

Vorlesung 10: 2-3-4-Bäume (K18)

Prof. Dr. Erika Ábrahám

Theorie Hybrider Systeme
Informatik 2

<http://ths.rwth-aachen.de/teaching/ss-14/datenstrukturen-und-algorithmen/>

Diese Präsentation verwendet in Teilen Folien von Joost-Pieter Katoen.

22. Mai 2014 und 30. Mai 2014



Übersicht

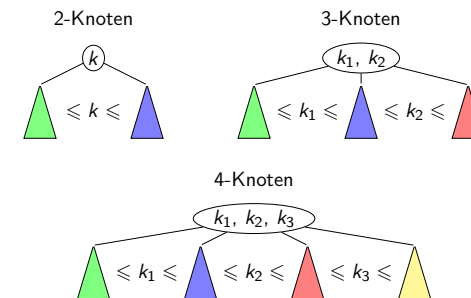
1 2-3-4-Bäume

Übersicht

1 2-3-4-Bäume

2-3-4-Bäume

Gehen die Operationen in logarithmischer Zeit nur für Binärbäume?



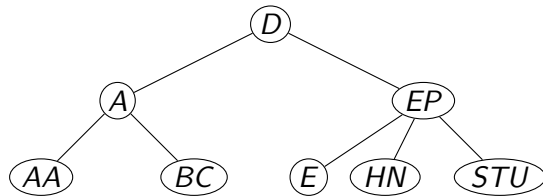
n -Knoten haben $n-1$ Schlüssel.

Innere n -Knoten haben n Kinder (\neq null), Blätter n -Knoten haben keine Kinder.

2-3-4-Baum: nur 2-, 3- oder 4-Knoten, alle Blätter haben die gleiche Tiefe (vollständiger Baum mit voller letzten Ebene).

Spezialfall von B-Bäumen. Anwendung: **Speicherverwaltung**.

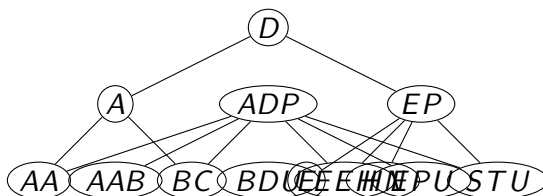
2-3-4-Bäume: Beispiel



2-3-4-Bäume: Einfügen eines Schlüssels

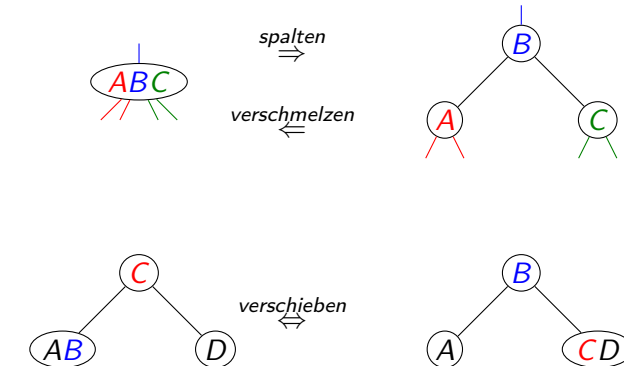
- ▶ **Steige** von der Wurzel zum Blatt, in dem eingefügt werden soll, hinunter (ähnlich wie im Binärbaum).
- ▶ **Spalte** auf dem Weg alle maximalen Knoten auf (benötigt nicht-maximalen Vaterknoten, bei der Wurzel wird der Baum nach oben erhöht).
- ▶ **Füge** den Schlüssel im Blatt ein.

Beispiel: Insert "BUDAPEST" und "AACHEN" in einen leeren 2-3-4-Baum



2-3-4-Bäume: Operationen

- ▶ Darstellung einer Multimenge durch 2-3-4-Bäume ist nicht eindeutig.
- ▶ Wir können 4-Knoten **spalten**.
- ▶ Wir können 2-Knoten **verschmelzen**.
- ▶ Wir können Schlüssel **verschieben** (rotieren).



2-3-4-Bäume: Löschen eines Schlüssels

- ▶ Bei 2-3-4-Bäumen können wir im inneren Teil des Baumes keine neuen Knoten **einfügen**, da alle Pfadlängen von der Wurzel zu einem Blatt gleich sein müssen.
- ▶ Daher **vergrößern** wir 2-3-4-Bäume stets oben an der Wurzel (durch Splitten der Wurzel erhöht sich die Baumhöhe um 1; splitten anderer Knoten ändern die Baumhöhe nicht, da mittlerer Schlüssel in den Vaterknoten nach oben geschoben wird).
- ▶ Um beim Einfügen Platz für den neuen Schlüssel zu sichern, haben wir bei der Suche entlang des Pfades von der Wurzel zur Einfügestelle jeden besuchten maximalen Knoten **aufgespalten**.
- ▶ Beim **Löschen** können wir genauso wenig einen einzelnen Knoten im inneren des Baumes ohne weiteres entfernen.
- ▶ Daher **verkleinern** wir 2-3-4-Bäume stets oben an der Wurzel (verschmelzen der Wurzel mit seinen Kindern verringert die Baumhöhe um 1).
- ▶ Um beim Löschen zu zu sichern, dass durch das Entfernen des Schlüssels der betroffene Knoten nicht leer wird, **vergrößern** wir alle minimalen Knoten entlang des Pfades von der Wurzel zur Stelle des Löschens (durch

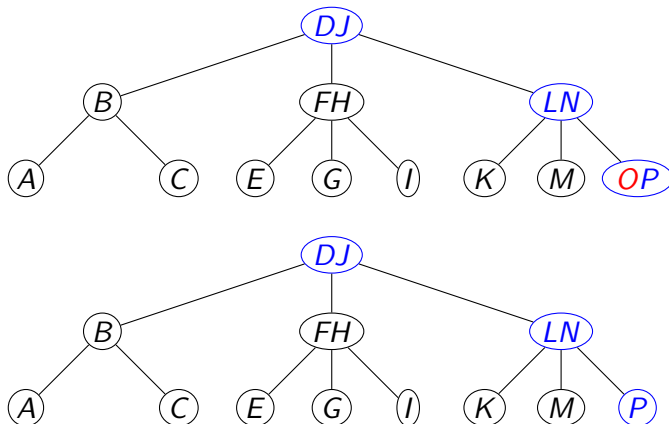
2-3-4-Bäume: Löschen eines Schlüssels

- ▶ Wenn wir den Knoten x , aus dem ein Schlüssel k entfernt werden soll, erreicht haben, ist dieser Knoten daher **nicht minimal**.
- ▶ Wenn x ein **Blatt** ist, können wir den Schlüssel k einfach entfernen.
- ▶ Wenn x **kein Blatt** ist, können wir den Schlüssel k nicht ohne Weiteres entfernen, denn dann hätte x ein Kind zu viel.
 - ▶ Wenn das Kind y von x , das dem zu löschenden Schlüssel k vorangeht, nicht minimal ist, dann im Teilbaum mit Wurzel y sind genug Schlüssel enthalten um den **Vorgängerschlüssel** k' von k rekursiv zu löschen, ohne die Teilbaumhöhe zu verändern. Wir löschen k' und ersetzen k durch k' (NB: da kein Schlüssel im Baum zwischen k und k' liegt, bleibt die Suchbaum-Eigenschaft erhalten).
 - ▶ Analog, wenn das Kind z von x , das auf k folgt, nicht minimal ist, können wir den **Nachfolgerschlüssel** k' im Teilbaum rekursiv löschen, ohne die Baumhöhe zu verändern, und k durch k' ersetzen.
 - ▶ Sonst, wenn sowohl y als auch z minimal sind, **verschmelze** y , z und k in y (oder in z , das ist egal; k wird dabei aus x in den verschmolzenen Knoten verschoben) und lösche den Schlüssel k in y rekursiv.

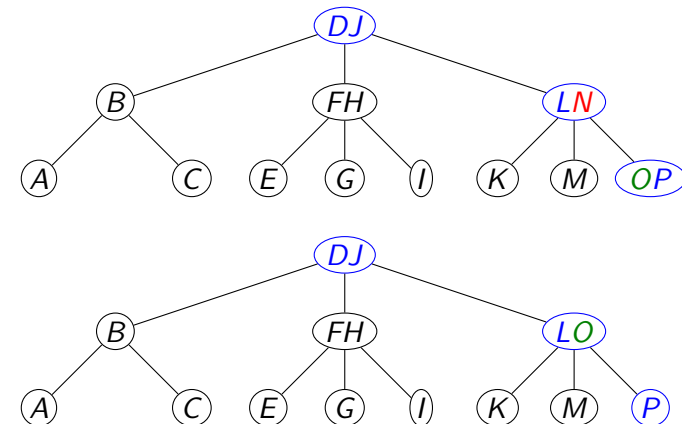
2-3-4-Bäume: Löschen eines Schlüssels

1. **Steige** von der Wurzel zum Knoten (nicht unbedingt Blatt), in dem der Schlüssel k gelöscht werden soll, hinunter (ähnlich zum Einfügen).
2. **Vergrößere** auf dem Weg alle minimalen Knoten durch
 - 2.1 **Verschiebung** von Schlüssel oder
 - 2.2 **Verschmelzung** von Knoten.
3. Fallunterscheidung Zielknoten:
 - 3.1 **Blatt**: lösche den Schlüssel k (beachte: Blatt ist nicht minimal!).
 - 3.2 **Innerer Knoten**:
 - 3.2.1 Wenn ein Kind, das entweder den Nachfolger- oder den Vorgängerschlüssel k' von k enthält, nicht minimal ist, dann lösche k' (rekursiv) und ersetze k durch k' .
 - 3.2.2 Sonst **verschmelze** die Kinder, die den Vorgänger- und Nachfolgerschlüsseln enthalten, mit k und lösche k rekursiv.

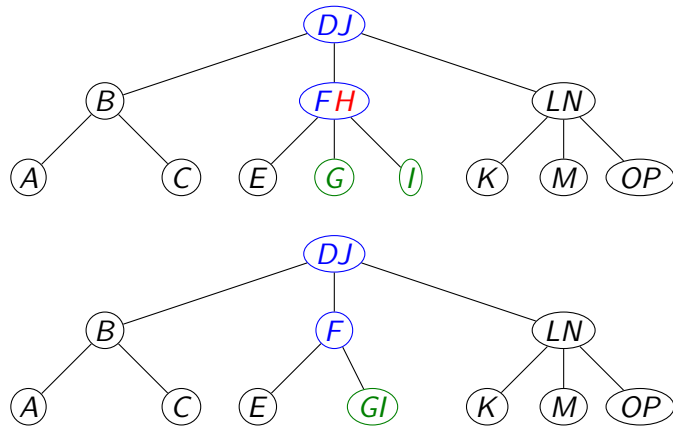
2-3-4-Bäume: Beispiel Löschen (Fall 3.1)



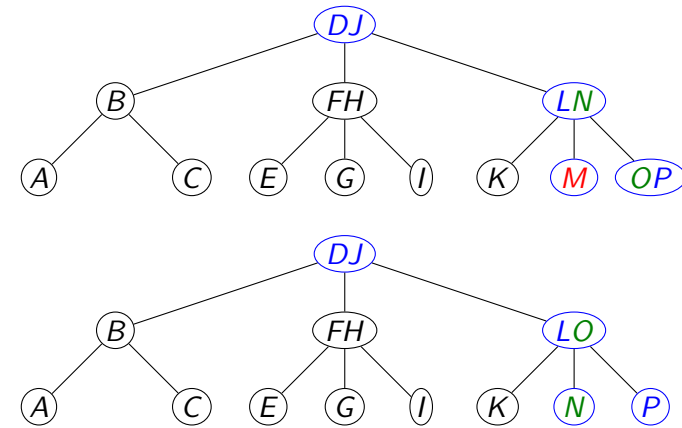
2-3-4-Bäume: Beispiel Löschen (Fall 3.2.1)



2-3-4-Bäume: Beispiel Löschen (Fall 3.2.2)



2-3-4-Bäume: Beispiel Löschen (Fall 2.1 + 3.1)



2-3-4-Bäume: Beispiel Löschen (Fall 2.2 + 3.1)

