IIF
INSTITUT FÜR
INFORMATIK
FREIBURG

Vorlesung Informatik 2
Algorithmen und Datenstrukturen

(20 - Bäume: Analyse nat. Bäume)

Prof. Dr. Susanne Albers

Binäre Suchbäume

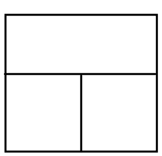
Binärbäume zur Speicherung von Mengen von Schlüsseln (in den inneren Knoten der Bäume), so dass die Operationen

- Suchen (find)
- Einfügen (insert)
- Entfernen (remove, delete)

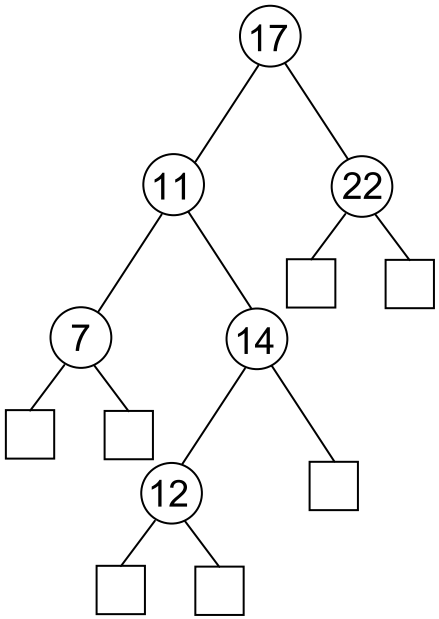
unterstützt werden.

Suchbaumeigenschaft: Die Schlüssel im linken Teilbaum eines Knotens *p* sind alle kleiner als der Schlüssel von *p*, und dieser ist wiederum kleiner als sämtliche Schlüssel im rechten Teilbaum von *p*.

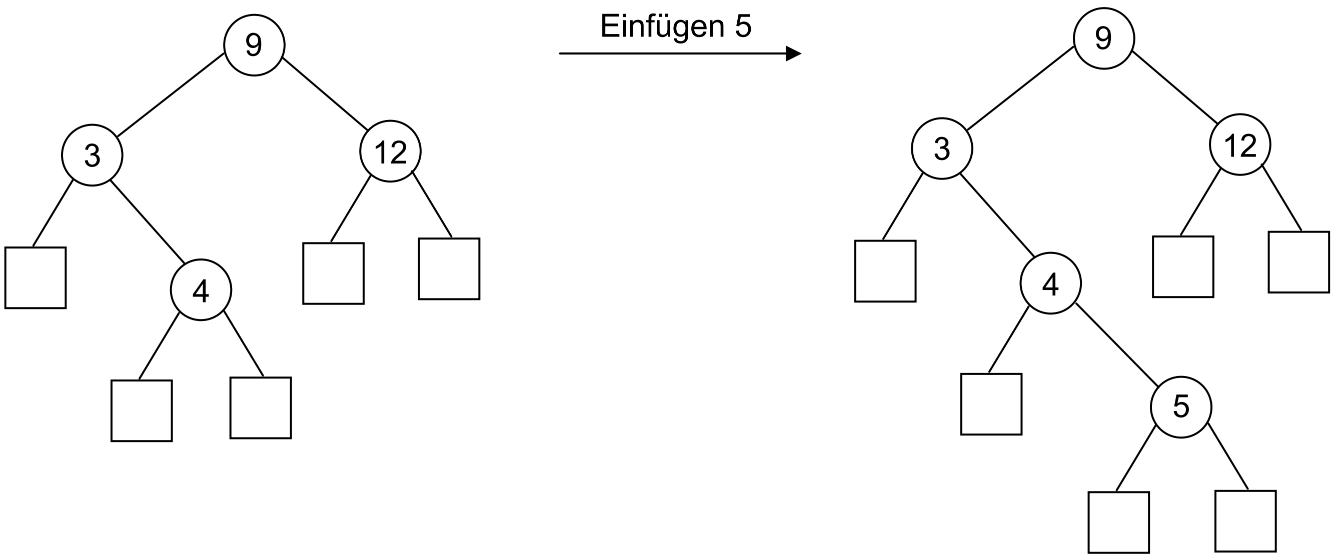
Implementierung:



Beispiel für Suchen, Einfügen, Entfernen



Natürliche Bäume



- Baum-Struktur hängt von Einfügereihenfolge in anfangs leeren Baum ab
- Höhe kann linear zunehmen, sie kann aber auch in O(log *n*) sein, genau

  $\lceil \log_2(n+1) \rceil$

Analyse natürlicher Suchbäume

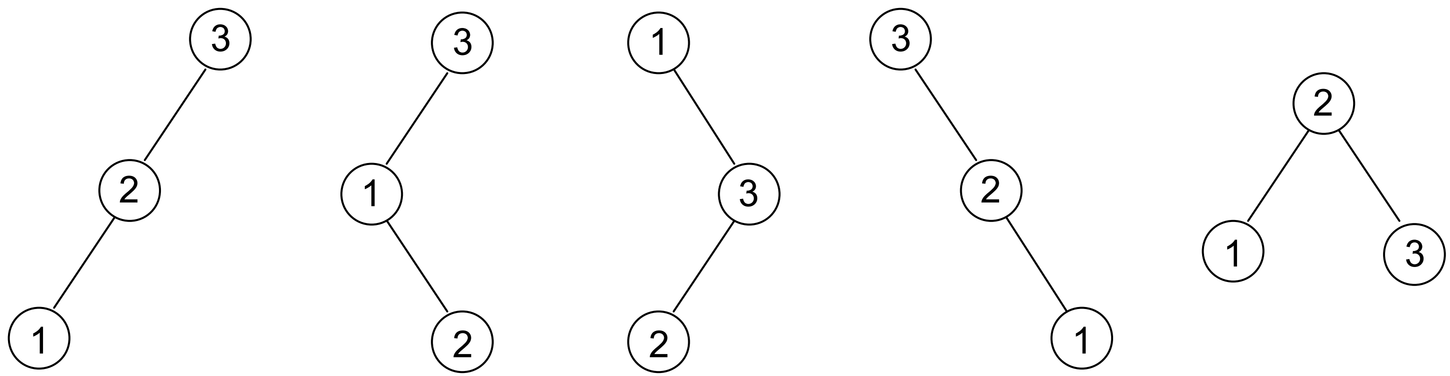
Zwei alternative Vorgehensweisen zur Bestimmung der internen Pfadlänge:

1. Random-Tree-Analyse, d.h. Mittelwert über alle möglichen Permutationen der (in den anfangs leeren Baum) einzufügenden Schlüssel.
2. Gestaltanalyse, d.h. Mittelwert über alle strukturell möglichen Bäume mit n Schlüsseln.

Unterschied des Erwartungswertes für die interne Pfadlänge:

1. $\approx 1.386\, n \log_2 n - 0.846\, n + O(\log n)$
2. $\approx n * \sqrt{\pi n} + O(n)$

Ursache für den Unterschied



3,2,1 3,1,2 1,3,2 3,2,1 2,1,3 und 2,3,1

→ Bei der Random-Tree-Analyse werden ausgeglichene Bäume häufiger gezählt.

Interne Pfadlänge

Interne Pfadlänge: Maß zu Beurteilung der Güte eines Suchbaumes.

Rekursive Definition:

1. Ist t der leere Baum, so ist

$$I(t) = 0.$$

2. Für einen Baum mit Wurzel t, linkem Teilbaum t_l und rechtem Teilbaum t_r gilt:

$$I(t) := I(t_l) + I(t_r) + \text{Zahl der Knoten von } t.$$

Offensichtlich gilt:

$$I(t) = \sum_{p} \left(Tiefe(p) + 1 \right)$$

P innerer Knoten von t

Durchschnittliche Suchpfadlänge

Für einen Baum t ist die durchschnittliche Suchpfadlänge definiert durch:

$$D(t) = I(t)/n, \; n = \text{Anzahl innerer Knoten in } t$$

Frage: Wie groß ist D(t) im

- besten
- schlechtesten
- mittleren Fall

für einen Baum t mit n inneren Knoten?

Interne Pfadlänge: Bester Fall

Es entsteht ein vollständiger Binärbaum

Interne Pfadlänge: Schlechtester Fall

Zufällige Bäume

- Seien oBdA die Schlüssel $\{1,\ldots,n\}$ einzufügen.
- Sei ferner $s_1,\ldots, s_n$ eine zufällige Permutation dieser Schlüssel.
- Somit ist die Wahrscheinlichkeit $P(s_1 = k)$, dass s_1 gerade den Wert k hat, genau $1/n$.
- Wenn k der erste Schlüssel ist, wird k zur Wurzel.
- Dann enthalten der linke Teilbaum $k - 1$ Elemente (nämlich die Schlüssel $1,\ldots,k - 1$) und der rechte Teilbaum $n - k$ Elemente (d.h. die Schlüssel $k + 1,\ldots,n$).

Erwartete interne Pfadlänge

$EI(n)$: Erwartungswert für die interne Pfadlänge eines zufällig erzeugten binären Suchbaums mit n Knoten

Offensichtlich gilt:

$$EI(0) = 0$$

$$EI(1) = 1$$

$$EI(n) = \frac{1}{n} \sum_{k=1}^{n} (EI(k-1) + EI(n-k) + n)$$

$$= n + \frac{1}{n} \left(\sum_{k=1}^{n} EI(k-1) + \sum_{k=1}^{n} EI(n-k) \right)$$

Behauptung: $EI(n) \approx 1.386n \log_2 n - 0.846n + O(\log n)$.

Beweis (1)

$$EI(n+1) = (n+1) + \frac{2}{n+1} * \sum_{k=0}^{n} EI(k)$$

und daher

$$(n+1) * EI(n+1) = (n+1)^2 + 2 * \sum_{k=0}^{n} EI(k)$$

$$n * EI(n) = n^2 + 2 * \sum_{k=0}^{n-1} EI(k)$$

Aus den beiden letzten Gleichungen folgt

$$(n+1)\ EI(n+1) - n * EI(n) = 2n + 1 + 2 * EI(n)$$

$$(n+1)\ EI\ (n+1) = (n+2)\ EI(n) + 2n + 1$$

$$EI(n+1) = \frac{2n+1}{n+1} + \frac{n+2}{n+1} EI(n).$$

Beweis (2)

Durch vollständige Induktion über *n* kann man zeigen, dass für alle n ≥ 1 gilt:

$$EI(n) = 2(n+1)\ H_n - 3n$$

$H_n = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$ ist die *n* -te harmonische Zahl, die wie folgt abgeschätzt werden kann:

$$H_n = \ln n + \gamma + \frac{1}{2n} + O\left(\frac{1}{n^2}\right)$$

Dabei ist $\gamma = 0.5772\ldots$ die so genannte Eulersche Konstante.

Beweis (3)

Damit ist $$EI(n) = 2n\ \ln n - (3-2\gamma) * n + 2\ln n + 1 + 2\gamma + O\left(\frac{1}{n}\right)$$

Und daher $$\frac{EI(n)}{n} = 2n\ \ln n - (3-2\gamma) + \frac{2\ln n}{n} + \ldots$$

$$= \frac{2}{\log_2 e} * \log_2 n - (3-2\gamma) + \frac{2\ln n}{n} + \ldots$$

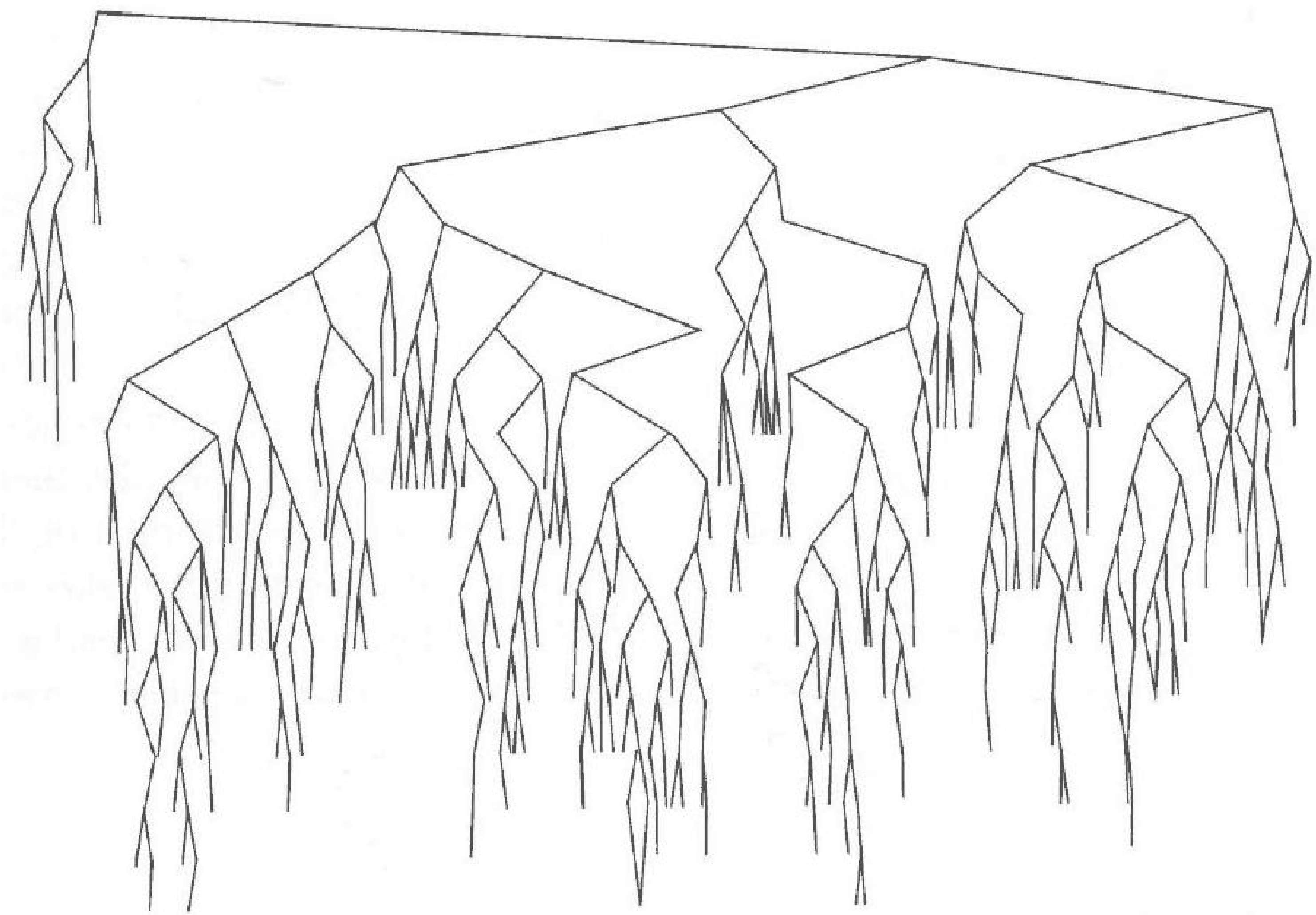
$$= \frac{2\log_{10} 2}{\log_{10} e} * \log_2 n - (3-2\gamma) + \frac{2\ln n}{n} + \ldots$$

$$\approx 1.386\ \log_2 n - (3-2\gamma) + \frac{2\ln n}{n} + \ldots$$

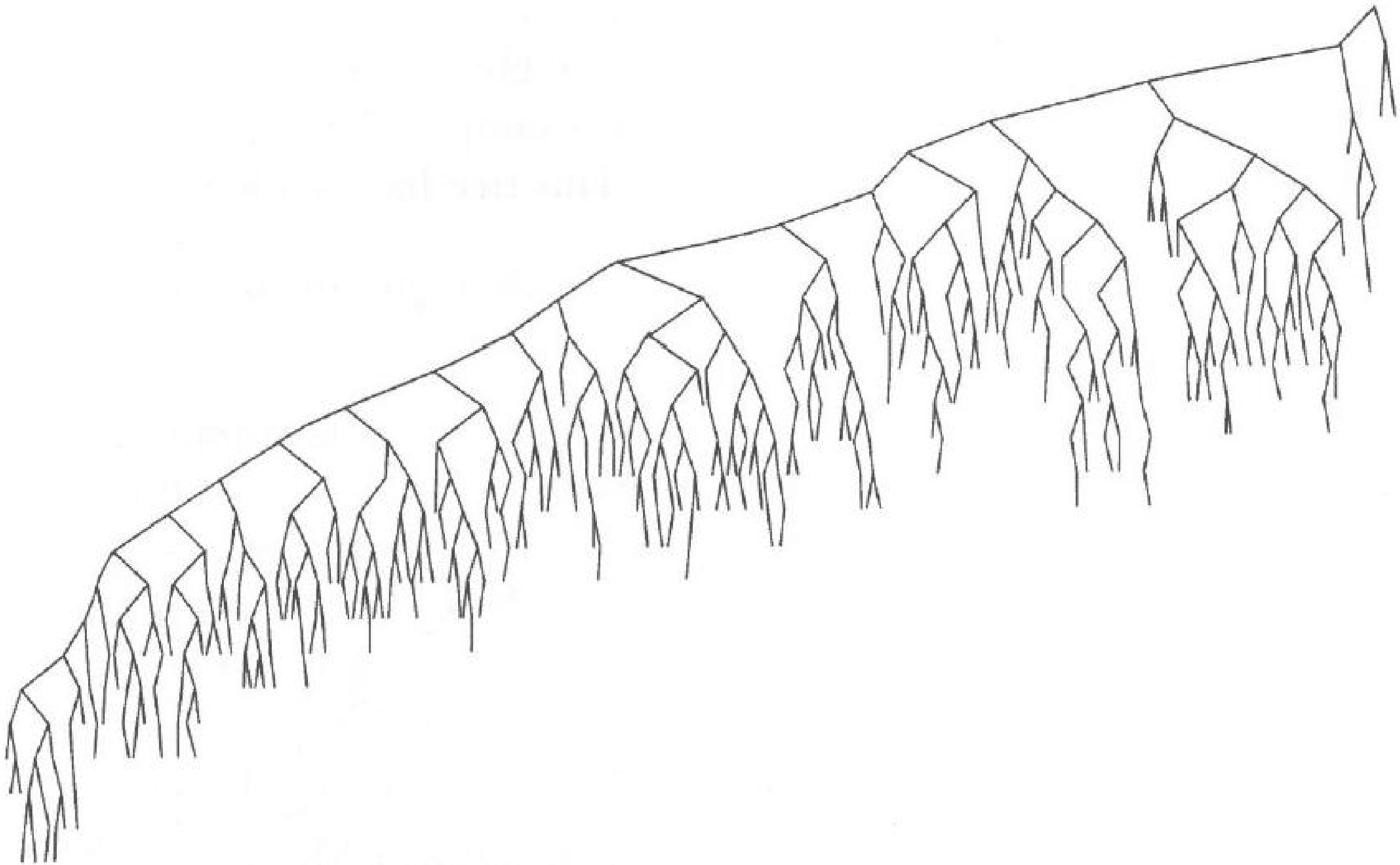
Beobachtungen

- Suchen, Einfügen und Entfernen eines Schlüssels ist bei einem zufällig erzeugten binären Suchbaum mit n Schlüsseln im Mittel in $O(\log_2 n)$ Schritten möglich.
- Im schlechtesten Fall kann der Aufwand jedoch $\Omega(n)$ betragen.
- Man kann nachweisen, dass der mittlere Abstand eines Knotens von der Wurzel in einem zufällig erzeugten Baum nur etwa 40% über dem Optimum liegt.
- Die Einschränkung auf den symmetrischen Nachfolger verschlechtert jedoch das Verhalten.
- Führt man in einem zufällig erzeugten Suchbaum mit n Schlüsseln n^2 Update-Operationen durch, so ist der Erwartungswert für die durschnittliche Suchpfadlänge lediglich $\Theta(\sqrt{n})$.

Typischer Binärbaum für eine zufällige Schlüsselsequenz



Resultierender Binärbaum nach n^2 Updates



Strukturelle Analyse von Binärbäumen

Frage: Wie groß ist die durchschnittliche Suchpfadlänge eines Binärbaumes mit n inneren Knoten, wenn man bei der Durchschnittsbildung jeden strukturell möglichen Binärbaum mit n inneren Knoten genau einmal zählt?

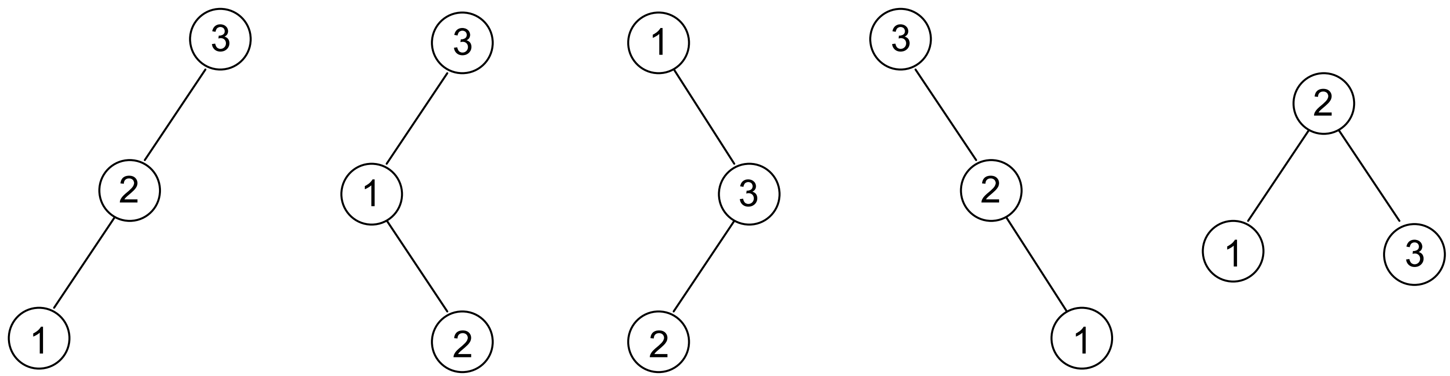
Antwort: Sei

I_n = gesamte interne Pfadlänge aller strukturell verschiedenen Binärbäume mit n inneren Knoten

B_n = Anzahl aller strukturell verschiedenen Bäume mit n inneren Knoten

Dann ist I_n/B_n =

Strukturell verschiedene Binärbäume



3,2,1 | 3,1,2 | 1,3,2 | 3,2,1 | 2,1,3 und 2,3,1

Anzahl strukturell verschiedener Binärbäume

Gesamte interne Pfadlänge aller Bäume mit n Knoten

- Für jeden Baum t mit linkem Teilbaum t_l und rechtem Teilbaum t_r gilt:

Zusammenfassung

Die durchschnittliche Suchpfadlänge in einem Baum mit n inneren Knoten (gemittelt über alle strukturell möglichen Bäume mit n inneren Knoten) ist:

$1/n\ I_n/B_n$