RWTH Aachen **SS 2016**
Lehrgebiet Theoretische Informatik Lösungsvorschlag 2
Rossmanith–Hensel–Kuinke–Sánchez 25. April 2016

Übung zur Vorlesung Datenstrukturen und Algorithmen

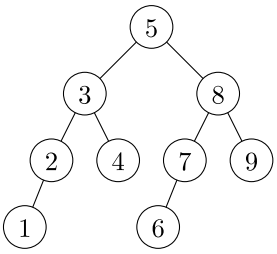
Aufgabe T5

a) Z.B.: $5, 3, 8, 2, 4, 7, 9, 1$

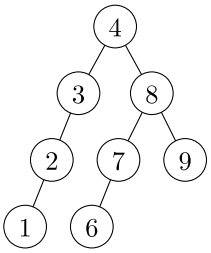
b) Es gibt mehr, nämlich genau 210. Hier sind sie:

53214879 53214897 53218479 53218497 53218749 53218794 53218947 53218974 53241879 53241897 53248179 53248197 53248719
53248791 53248917 53248971 53281479 53281497 53281749 53281794 53281947 53281974 53284179 53284197 53284719 53284791
53284917 53284971 53287149 53287194 53287419 53287491 53287914 53287941 53289147 53289174 53289417 53289471 53289714
53289741 53421879 53421897 53428179 53428197 53428719 53428791 53428917 53428971 53482179 53482197 53482719 53482791
53482917 53482971 53487219 53487291 53487921 53489217 53489271 53489721 53821479 53821497 53821749 53821794 53821947
53821974 53824179 53824197 53824719 53824791 53824917 53824971 53827149 53827194 53827419 53827491 53827914 53827941
53829147 53829174 53829417 53829471 53829714 53829741 53842179 53842197 53842719 53842791 53842917 53842971 53847219
53847291 53847921 53849217 53849271 53849721 53872149 53872194 53872419 53872491 53872914 53872941 53874219 53874291
53874921 53879214 53879241 53879421 53892147 53892174 53892417 53892471 53892714 53892741 53894217 53894271 53894721
53897214 53897241 53897421 58321479 58321497 58321749 58321794 58321947 58321974 58324179 58324197 58324719 58324791
58324917 58324971 58327149 58327194 58327419 58327491 58327914 58327941 58329147 58329174 58329417 58329471 58329714
58329741 58342179 58342197 58342719 58342791 58342917 58342971 58347219 58347291 58347921 58349217 58349271 58349721
58372149 58372194 58372419 58372491 58372914 58372941 58374219 58374291 58374921 58379214 58379241 58379421 58392147
58392174 58392417 58392471 58392714 58392741 58394217 58394271 58394721 58397214 58397241 58397421 58732149 58732194
58732419 58732491 58732914 58732941 58734219 58734291 58734921 58739214 58739241 58739421 58793214 58793241 58793421
58932147 58932174 58932417 58932471 58932714 58932741 58934217 58934271 58934721 58937214 58937241 58937421 58973214
58973241 58973421

c)



d)



Aufgabe T6

Binsearch(1):

1. $23 > 1$? $(a[m] > x)$
2. $6 > 1$? $(a[m] > x)$
3. $2 > 1$? $(a[m] > x)$

Es wurden 3 Vergleiche durchgefuehrt

Binsearch(12):

1. $23 > 12?$ $(a[m] > x)$
2. $6 < 12?$ $(a[m] < x)$
3. $8 < 12?$ $(a[m] < x)$
4. $12 == 12?$ $(a[m] == x)$

Es wurden 4 Vergleiche durchgefuehrt

Binsearch(8):

1. $23 > 8?$ $(a[m] > x)$
2. $6 < 8?$ $(a[m] < x)$
3. $8 == 8?$ $(a[m] == x)$

Es wurden 3 Vergleiche durchgefuehrt

Binsearch(23):

1. $23 == 23?$ $(a[m] == x)$

Es wurden 1 Vergleiche durchgefuehrt

Binsearch(24):

1. $23 < 24?$ $(a[m] < x)$
2. $67 > 24?$ $(a[m] > x)$
3. $35 > 24?$ $(a[m] > x)$

Es wurden 3 Vergleiche durchgefuehrt

Binsearch(81):

1. $23 < 81?$ $(a[m] < x)$
2. $67 < 81?$ $(a[m] < x)$
3. $82 > 81?$ $(a[m] > x)$
4. $80 < 81?$ $(a[m] < x)$

Es wurden 4 Vergleiche durchgefuehrt

Binsearch(100):

1. $23 < 100?$ $(a[m] < x)$
2. $67 < 100?$ $(a[m] < x)$
3. $82 < 100?$ $(a[m] < x)$
4. $99 < 100?$ $(a[m] < x)$

Es wurden 4 Vergleiche durchgefuehrt

Aufgabe T7
Wir kennen folgende Laufzeiten:

- Sortieren eines Arrays: $\Theta(n \log n)$,
- lineare Suche: $\Theta(n)$,
- binäre Suche: $\Theta(\log n)$.

Wir müssen uns also überlegen, wann folgende Ungleichung gilt:

$$k \cdot \Theta(n) > \Theta(n \log n) + k \cdot \Theta(\log n)$$

Die Ungleichung ist genau dann erfüllt, falls

- $k \cdot \Theta(n) > \Theta(n \log n)$ und
- $k \cdot \Theta(n) > k \cdot \Theta(\log n)$.

Da Letzteres für beliebige k gilt, müssen wir nur untersuchen, wann $k \cdot \Theta(n) > \Theta(n \log n)$ gilt, und erhalten $k > \Theta(\log n)$.

Aufgabe H4
Sei $k \geq 0$, so daß entweder $n = 2k$ oder $n = 2k + 1$. In beiden Fällen ist $\lceil (n-1)/2 \rceil = k$. Dann ist

$$\lfloor \log \lceil (n-1)/2 \rceil \rfloor + 1 = \lfloor \log k + \log 2 \rfloor = \lfloor \log 2k \rfloor$$

Für $n = 2k$ ist die Aussage damit bereits gezeigt. Für ungerade $n = 2k + 1$ sei $q \geq 0$ maximal mit $2^q < n$, also $q = \lfloor \log n \rfloor$. Wegen $2^q \leq 2k$ gilt dann auch

$$q \leq \lfloor \log 2k \rfloor \leq \lfloor \log n \rfloor = q.$$

Aufgabe H5
Wir suchen einen binären Suchbaum mit n Schlüsseln und einer Höhe, die kleiner ist als $\sqrt{n} - 100$.
Eine möglichst kleine Höhe erreichen wir durch volles Besetzen jeder Ebene:

- Ebene 0: 1 Schlüssel (Wurzelknoten)
- Ebene 1: 2 Schlüssel
- Ebene 2: 4 Schlüssel
- ...
- Ebene n: 2^n Schlüssel

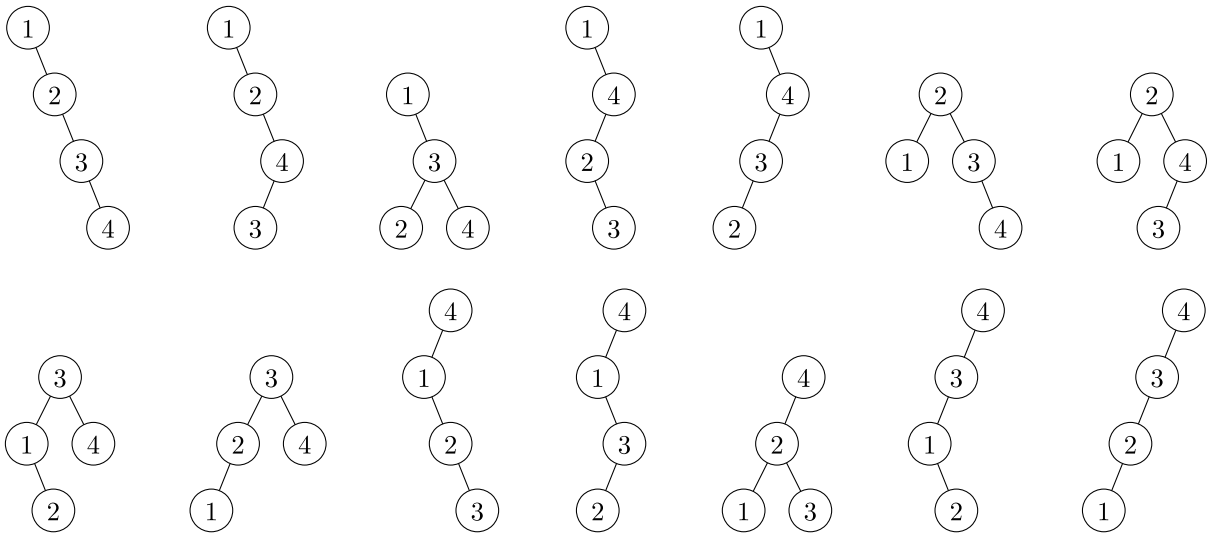
Ein Baum mit Höhe h (mit Ebenen 0 bis $h - 1$) hat dann

$$\sum_{k=0}^{h-1} 2^k = 1 + 2 + 4 + \ldots + 2^{h-1} = 2^h - 1$$

Schlüssel. Ein voll besetzter Binärbaum mit Höhe 14 hat also $2^{14} - 1 = 16383$ Knoten. Da $14 < 27,996 = \sqrt{16383} - 100$, ist dieser Baum ein Gegenbeispiel zur Behauptung.

Aufgabe H6

Es handelt sich um diese 14 Bäume:



Aufgabe H7

Die folgende Methode entscheidet, ob die Suchbaumeigenschaft im ganzen Baum erfüllt ist. Sie muß die Wurzel des Baums als Eingabe erhalten und testet dann die Suchbaumeigenschaft rekursiv.

```
boolean isBinarySearchtree(Searchtree⟨K, D⟩ tree) {
  Searchtreenode⟨K, D⟩ min = tree.root, max = tree.root;
  while(min.left ≠ null) min = min.left;
  while(max.right ≠ null) max = max.right;
  return isBinarySearchtree(tree.root, min.key, max.key);
}

boolean isBinarySearchtree(Searchtreenode⟨K, D⟩ node, K min, K max) {
  if(node == null) {
    return true;
  }
  if(node.key.compareTo(min) ≤ 0 || node.key.compareTo(max) ≥ 0) {
    return false;
  }
  return isBinarySearchtree(node.left, min, node.key)
      && isBinarySearchtree(node.right, node.key, max);
}
```