

Algorithmen und Datenstrukturen
SS 2018

Übungsblatt 8: Suchen

Tutorien: 05.06-11.06.2018

Aufgabe 8-1 *Lineare und Interpolationssuche*

- (a) Führen Sie eine lineare Suche auf dem Array $\{0,A; 6,E; 2,C; 5,D; 1,B\}$ nach 5 durch. Wieviele Schritte benötigen Sie? Wieviele Schritte werden, abhängig von der Listenlänge n , im schlechtesten Fall sowie durchschnittlich benötigt?

Lösungsvorschlag:

Es werden 4 Schritte benötigt, im schlechtesten Fall n , im Durchschnitt $n/2$.

- (b) Für welche Art von Listen ist lineare Suche sinnvoll einsetzbar? Wie kann eine schnellere Suche ermöglicht werden?

Lösungsvorschlag:

Lineare Suche ist für nicht sortierte Listen sinnvoll. Durch Sortieren von Listen können andere Sucharten wie z.B. Interpolationssuche oder Binäre Suche verwendet werden.

- (c) Sortieren Sie nun die Liste und führen Sie erneut eine Suche nach dem Schlüssel 2 mithilfe der Interpolationssuche durch. Wieviele Schritte benötigen Sie?

Lösungsvorschlag:

Sortiertes Array: $\{0,A; 1,B; 2,C; 5,D; 6,E\}$
1. Schritt: $low = 0; high = 4; a = 2;$
 $i = 0 + (2-0)*4/(6-0) = 1$
 a ist größer als 1, daher $low = i + 1 = 1 + 1 = 2$
2.Schritt: $low = 2; high = 4; a = 2$
 $i = 2 + (2-2) * (4-2) / (6-2) = 2 + 0 = 2$
 a ist gleich dem Element an Position 2. Suche beendet

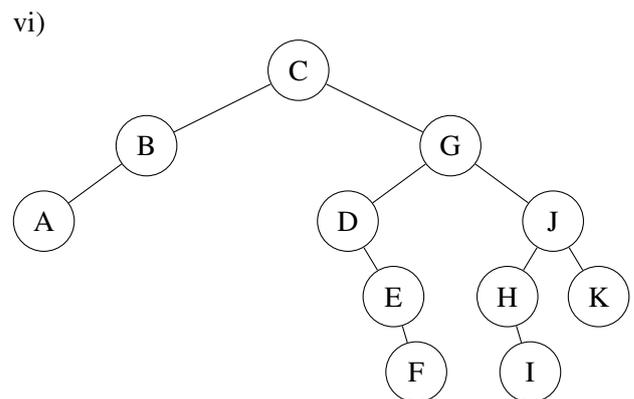
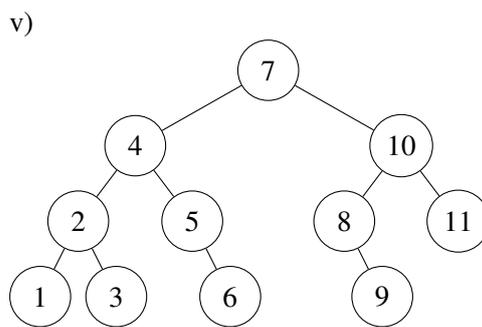
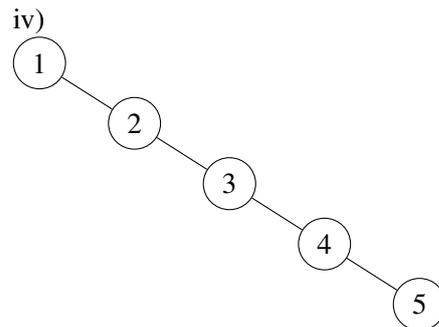
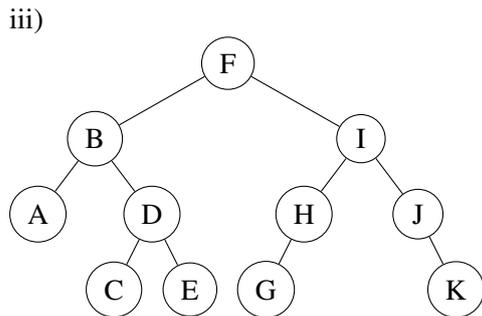
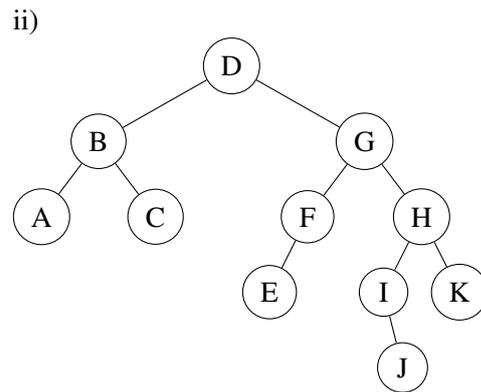
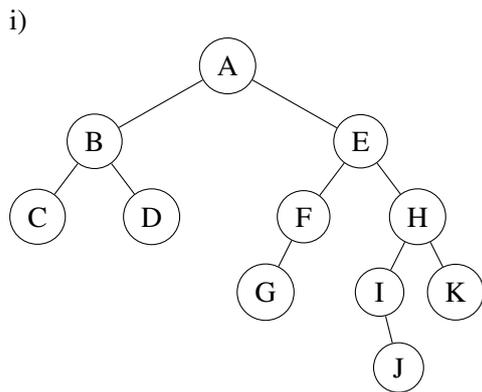
- (d) Wann ist keine Interpolationssuche sinnvoll auf einem Array möglich? Begründen Sie Ihre Antwort.

Lösungsvorschlag:

Wenn die Schlüsselwerte unregelmäßige Abstände haben, kann eine erhöhte Suchlaufzeit entstehen.

Aufgabe 8-2 *Baum-Vergleich und Splay-Bäume*

Gegeben ist ein Wald von binären Bäumen:



Beantworten Sie folgende Fragen:

- (a) Bei welchen dieser Bäume handelt es sich um Suchbäume, bei welchen nicht? Begründen Sie ihre Lösung.
- (b) Bei welchen dieser Bäume handelt es sich um AVL-Bäume, bei welchen nicht? Begründen Sie ihre Lösung.
- (c) (i) Angenommen Baum v) ist ein Splay-Baum. Es wird das Element '3' gesucht. Zeichnen Sie die Bäume, die nach den einzelnen Rotationen des Splays entstehen. Wie viele Rotationen sind ausgeführt worden?
- (ii) Repräsentieren der aus Baum v) entstandene Baum aus obiger Teilaufgabe und Baum vi) die gleichen Daten? Begründen Sie.
- (iii) Angenommen es wird n-mal nach Element '3' gesucht. Wie viele Vergleichsoperationen sind nötig, wenn es sich bei Baum v) um einen Splay-Baum handelt, wie viele, wenn Baum v) kein Splay-Baum ist.
Hinweis: Die Überprüfung und Einordnung, ob ein Element größer, kleiner oder gleich einem anderen Element ist, kann für die Aufgaben auf diesem Blatt als eine einzige Vergleichsoperation betrachtet werden.

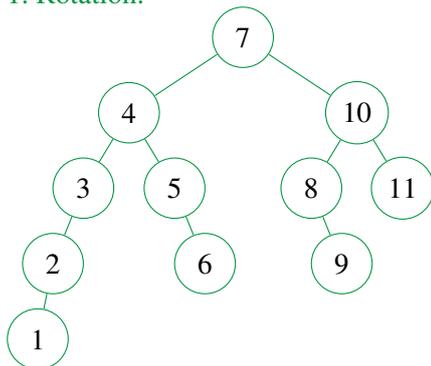
- (iv) Ausgehend von den Ergebnissen aus der vorherigen Teilaufgabe. Halten Sie es für sinnvoll Baum v) als Splay-Baum zu implementieren? Begründen Sie ihre Meinung.
- (v) Angenommen jeder Eintrag des Baumes v) wird mit einer Wahrscheinlichkeit von $\frac{1}{11}$ gesucht. Ist es nun sinnvoll diesen Baum als Splay-Baum zu implementieren?

Lösungsvorschlag:

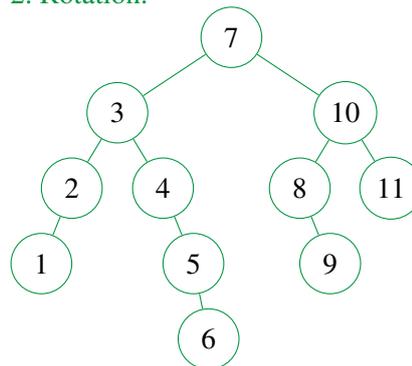
- (a)
 - (i) Ist **kein** Suchbaum: A ist beispielsweise lexikografisch kleiner als D, obwohl A rechts von D im Baum ist.
 - (ii) Ist **kein** Suchbaum: I und J sind lexikografisch größer als H, aber linke Kindknoten von H.
 - (iii) Ist **ein** Suchbaum: Alle Elemente sind lexikografisch aufsteigend von links nach rechts sortiert.
 - (iv) Ist **ein** Suchbaum: Alle Elemente sind aufsteigend von links nach rechts sortiert. Auch wenn es sich hierbei um einen entarteten Baum in Form einer Liste handelt, erfüllt er dennoch die Suchbaumeigenschaften.
 - (v) Ist **ein** Suchbaum: Alle Elemente sind aufsteigend von links nach rechts sortiert.
 - (vi) Ist **ein** Suchbaum: Alle Elemente sind lexikografisch aufsteigend von links nach rechts sortiert.
- (b)
 - (i) Ist **kein** AVL-Baum: Der Baum erfüllt nicht die notwendigen Suchbaumeigenschaften und kann somit erst recht kein AVL-Baum sein.
 - (ii) Ist **kein** AVL-Baum: Dieser Baum ist zwar ein Suchbaum, jedoch ist die Balance für die Wurzel +2 (die Höhe des rechten Teilbaums ist um 2 größer als die des linken). Somit handelt es sich nicht um einen AVL-Baum, deren für die jeder Knoten eine Balance von -1, 0 oder +1 haben muss.
 - (iii) Ist **ein** AVL-Baum: Es handelt sich um einen Suchbaum und außerdem besitzt jeder Knoten des Baumes eine Balance von -1, 0 oder 1. Somit sind alle Kriterien für einen AVL-Baum erfüllt.
 - (iv) Ist **kein** AVL-Baum: Dieser Baum ist zwar ein Suchbaum, jedoch ist die Balance für die Wurzel beispielsweise bei +4, weswegen die AVL-Eigenschaft der ausgeglichenen Höhe verletzt ist.
 - (v) Ist **ein** AVL-Baum: Es handelt sich um einen Suchbaum und außerdem besitzt jeder Knoten des Baumes eine Balance von -1, 0 oder 1. Somit sind alle Kriterien für einen AVL-Baum erfüllt.
 - (vi) Ist **kein** AVL-Baum: Dieser Baum ist zwar ein Suchbaum, jedoch ist die Balance für die Wurzel und für das Element 'D' beispielsweise bei +2, weswegen die AVL-Eigenschaft der ausgeglichenen Höhe verletzt ist.

Lösungsvorschlag:

c) (i) 1. Rotation:

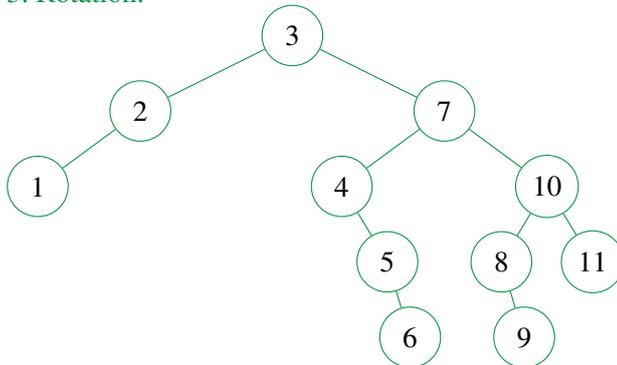


2. Rotation:



⇒ Die Rotationen 1 und 2 ergeben zusammen eine 'zig-zag' Doppelrotation.

3. Rotation:



⇒ Es werden insgesamt 3 Rotationen durchgeführt.

- (ii) Nein, tun sie nicht. Die Bäume besitzen zwar die gleiche Form und sind somit isomorph. Die Inhalte der Bäume unterscheiden sich jedoch. Bei dem einen Baum handelt es sich um Zahlen beim anderen um Buchstaben. Würde man nun die Buchstaben entsprechend ihres Auftretens im Alphabet durch Zahlen ersetzen, dann würden beide Bäume die gleichen Daten repräsentieren.
- (iii) Splay-Baum: 4 Vergleichsoperationen für die erste Iteration. Anschließend erfolgt ein Splay und 3 ist fortan die Wurzel. Danach ist jeweils nur 1 Vergleichsoperation nötig. Es wird n -mal iteriert
⇒ Es sind $4 + (n - 1) \cdot 1 = n + 3$ Vergleichsoperationen notwendig.
Kein Splay-Baum: Pro Iteration (bzw. Suchdurchlauf) sind 4 Vergleichsoperationen notwendig. Es wird n -mal iteriert.
⇒ Es sind $4n$ Vergleichsoperationen notwendig.
- (v) Der Vorteil eines Splay-Baumes ist es, dass die Einzelschritte bei Durchführung von Suchoperationen verringert werden, wenn öfter nach den gleichen Elementen gesucht wird. Dies wird durch Rotationen innerhalb des Baumes bewirkt. Da bei der n -fachen Suche nach der Zahl '3' durch die Splays weniger Vergleiche ausgeführt werden als wenn keine Splays durchgeführt werden würden, bietet es sich hier an, den Baum als Splay-Baum zu implementieren.
- (vi) Da hier jedes Element des Baumes gleich oft gesucht wird, bringen die Splays keine Suchvorteile. Im Gegenteil verbrauchen sie selbst Rechenzeit und die balancierte Struktur des AVL-Baumes wird aufgelöst. Im Mittel ist es also hier schlechter, wenn man den Baum als Splay-Baum implementiert. Deswegen sollte hier die Implementierung als AVL-Baum beibehalten werden.