

Selbsttest 4

1) Löschen in Binärbäumen

Beliebigen Binärbaum aufzeichnen (Knoten mit Grad 1 sollten darin vorkommen!). Wenn gewünscht, kann man hier auch das Einfügen noch mal üben.

Löschen von einigen Elementen (Beginnend mit Knoten mit Grad 0, danach Grad 1 und Grad 2).

Lösung:

- Löschen von Blättern nur durch Entfernen dieses Blattes
- Löschen von Knoten mit Grad 1 durch "Verbiegen" der Referenz des Vaterknotens auf den jeweiligen Sohn (nur einer da) des zu löschenden Knotens
- Löschen von Knoten mit Grad 2:
 - Suchen des zu löschenden Knotens
 - Suchen des nächstgrößeren Knotens (einmal nach rechts und danach so lange nach links gehen, bis dort kein Knoten mehr zu finden ist). Dieser Knoten muss ein Blatt oder ein Knoten mit Grad 1 sein => Entfernen des Knotens wie unter a) bzw. b)
 - Einfügen des entfernten Knotens statt des zu löschenden Knotens

2) 2-3-4 Bäume (zeichnen)

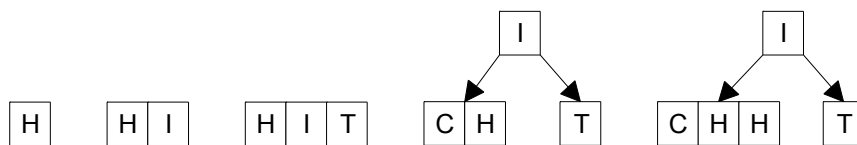
Simulieren des Einfügens der Buchstabenfolge "HITCHHIKERSGUIDE" in 2-3-4.

Lösung:

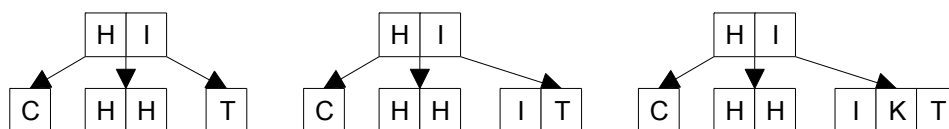
Einfügen immer in Blättern.

- Blatt suchen, in das eingefügt werden soll.
 - Wenn ein Knoten mit 3 Blättern gefunden, diesen **auf dem Weg nach unten splitten** (4-Knoten auf 2 2-Knoten aufteilen und mittleres Element nach oben reichen).
- Schlüssel einfügen (nur in Blatt)
 - Wenn bereits 1 oder 2 Schlüssel => einfach einfügen
 - Wenn bereits 3 Schlüssel => Blatt splitten auf 2 Blätter (mittleres Element kommt nach oben) und dann Schlüssel einfügen.

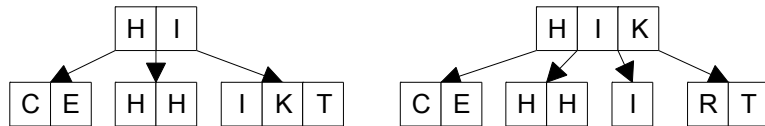
H, I, T, C, H



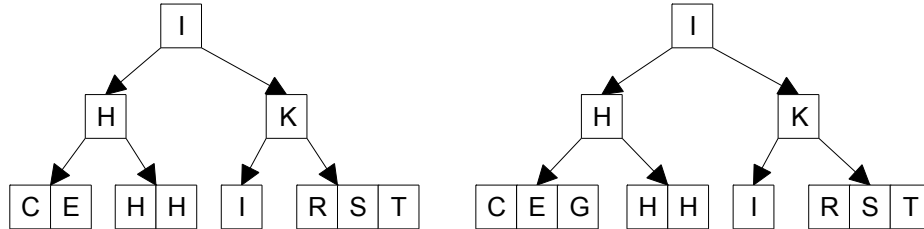
H, I, K



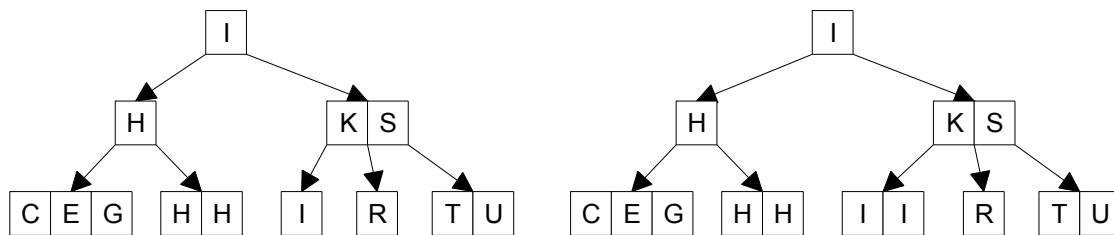
E, R



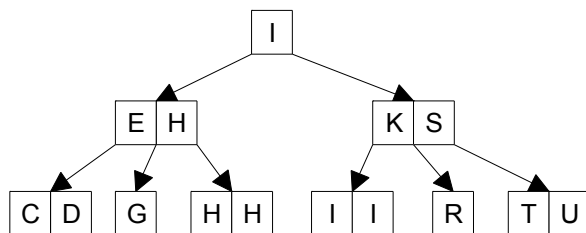
S, G



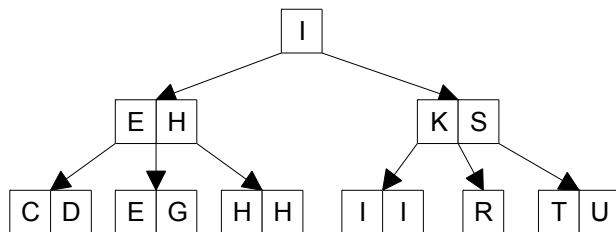
U, I



D



E



3) Balancieren von Bäumen - Diskussion

- Wie kann man feststellen, ob ein Baum balanciert ist oder nicht? Gib einige mögliche Kriterien dazu an.
- Wie kann man einen Binärbaum "im Hintergrund" balancieren, sodass der Anwender davon möglichst nichts mitbekommt?
- Welche der Kriterien unter a) muss man wie anwenden, sodass der Punkt b) auch eine Effizienzsteigerung ergibt?

Lösung:

a) Es gibt unterschiedliche Kriterien zum Feststellen der "Schiefe"(Grad der Entartung) eines Binärbaumes.

Was wären mögliche Maßzahlen, die als Grundlage verwendet werden könnten?

- Maximale Tiefe des Baums (Maximale Pfadlänge von der Wurzel bis zu einem Blatt)
- Minimale Tiefe des Baums (Minimale Pfadlänge von der Wurzel bis zu einem Blatt)
- Mittlere Tiefe des Baums (aller Elemente oder nur der Blätter)
- Anzahl der Knoten im Baum
- Anzahl der Knoten mit Grad 1

- ...

In Kombination (Beispiele):

Maximale Tiefe / $\lg(\text{Anzahl der Knoten}) > k$ ($k > 1$)

Maximale Tiefe / Minimale Tiefe $> k$ ($k > 1$)

=> Beschränken die maximale Suchzeit

Mittlere Tiefe / $\lg(\text{Anzahl der Knoten}) > k$ ($k > 1$)

=> Garantiert eine gewisse mittlere Suchzeit

Anzahl der Knoten mit Grad 1 / Anzahl der Knoten im Baum $> k$ ($k > 0$)

...

b)

Es muss regelmäßig überprüft werden, ob das jeweilige Kriterium erfüllt ist.

*Passend für eine solche Überprüfung sind Methoden, die den Baum strukturell verändern:
insert, remove, ...*

Unpassend sind Methoden, die den Baum nicht strukturell verändern (oder nicht entarten lassen):

find, printAll, removeAll, ...

c)

*Idee des Balancierens ist es die Baum-Datenstruktur (nach bestimmten Kriterien) effizienter zu machen. D.h. z.B. maximale od. mittlere Suchzeit beschränken (oder beides)
Dabei soll jedoch das Balancieren und das Feststellen der Schiefe nicht so viel Zeit benötigen, sodass man insgesamt länger braucht.*

2 Ansätze:

- Möglichst seltenes Überprüfen der Schiefe (nur bei jedem n-ten Einfügen überprüfen)
- Möglichst einfache Kriterien für die Schiefe anwenden (Mitzählen der Knotenanzahl, ... beim Einfügen/Löschen von Knoten => Maßzahlen müssen nicht erst aufwändig ermittelt werden)

Vorsicht! In der Regel muss nur sehr selten balanciert werden, meist neigen Bäume dazu von Haus aus (halbwegs) balanciert zu sein.