

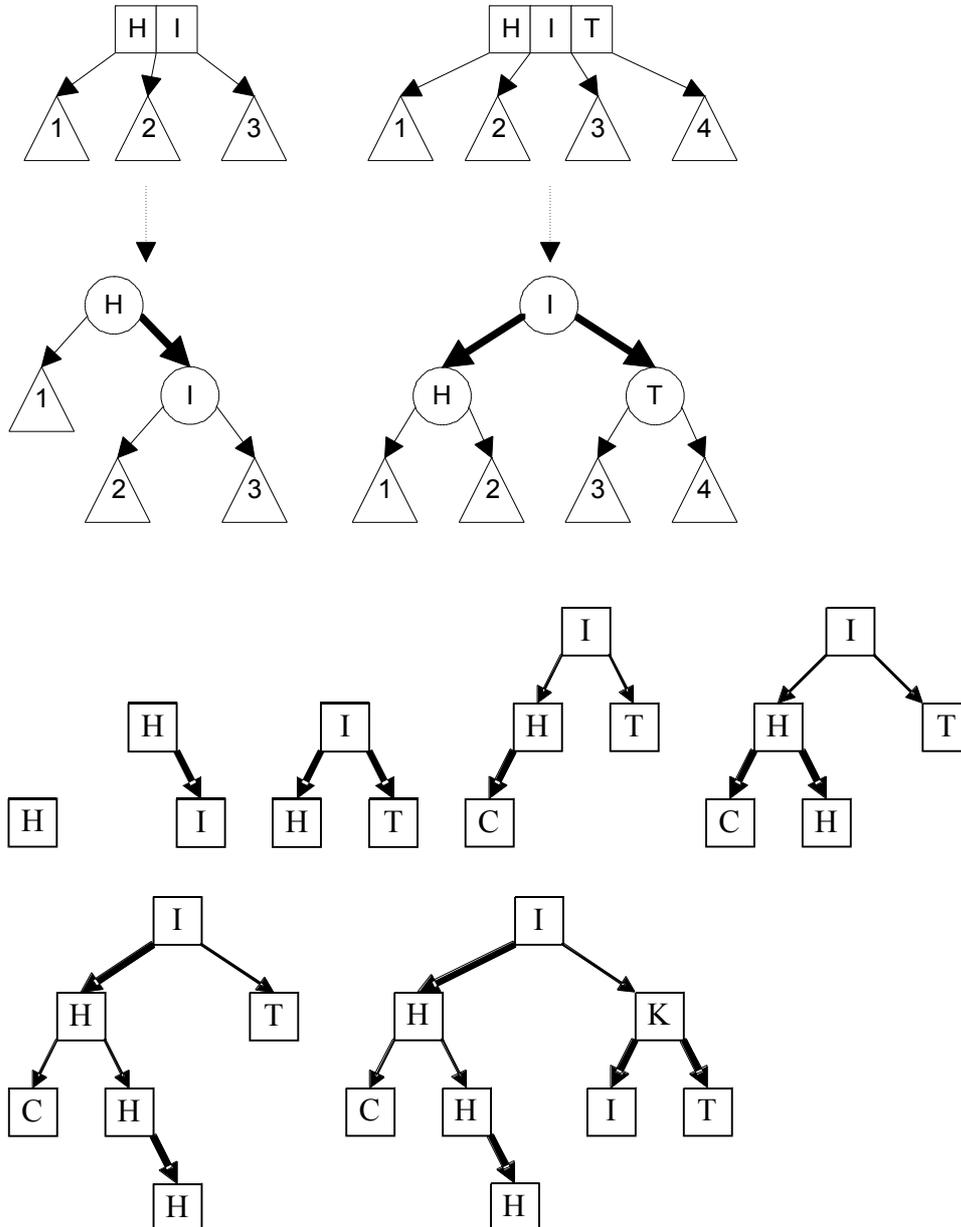
Selbsttest 5

1) Rot-Schwarz Bäume (zeichnen)

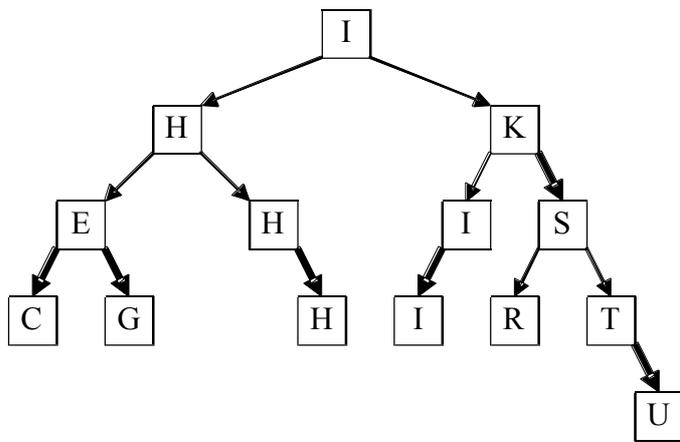
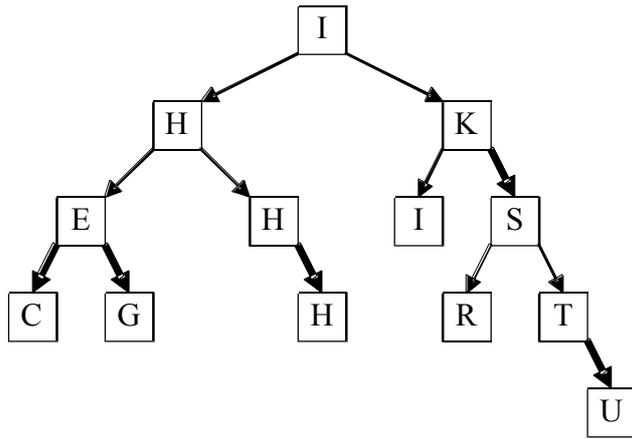
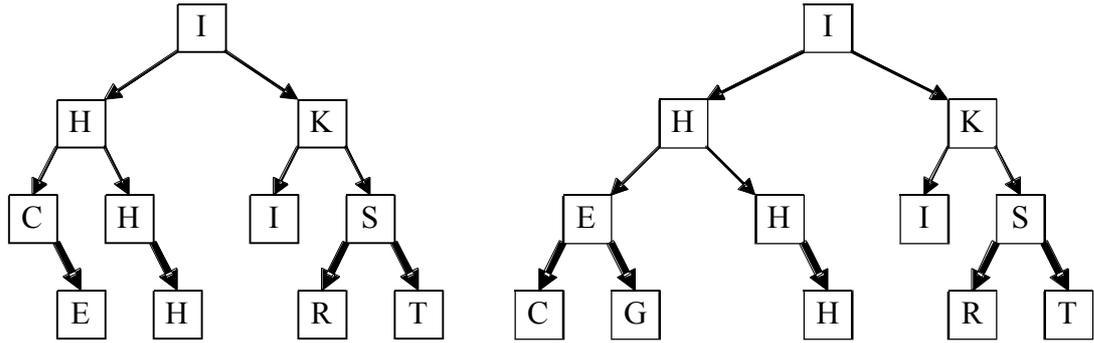
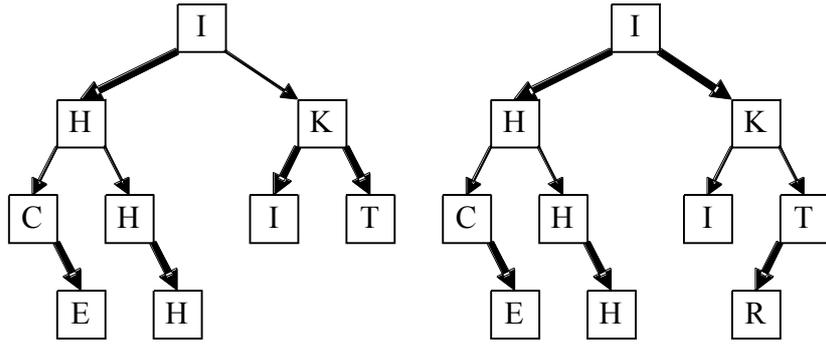
Geben Sie Rot-Schwarz-Bäume an (mit allen Zwischenschritten), die beim Einfügen der Buchstaben HITCHHIKERSGUIDE entstehen.

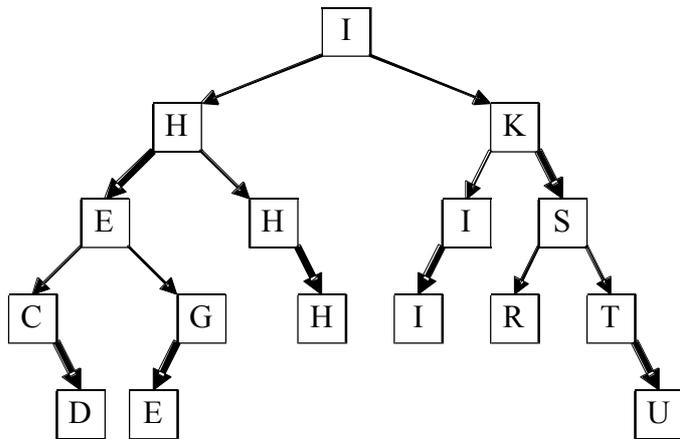
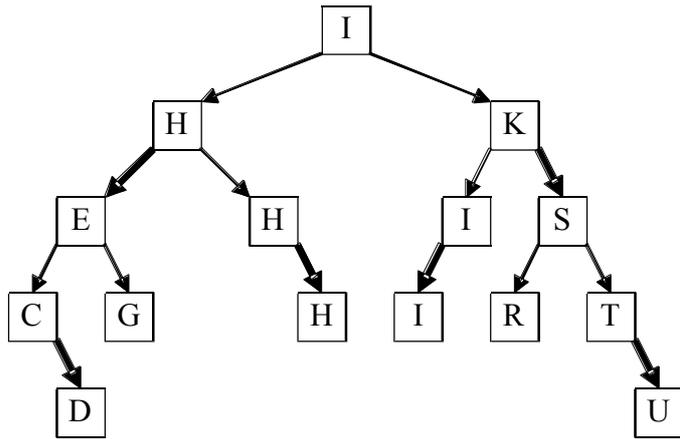
Lösung:

Die entsprechenden Rot-Schwarz-Bäume ergeben sich durch Ersetzen der 3- und 4-Knoten von 2-3-4 Bäumen mit Binärknoten und gefärbten Kanten (Rot-Schwarz-bäume sind gedachte 2-3-4 Bäume).



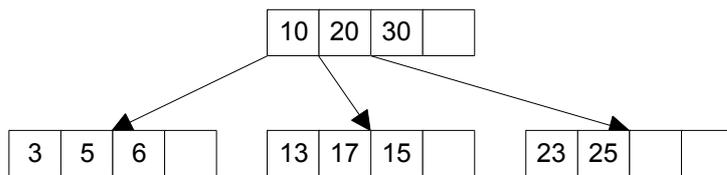
Balanciert? Ja! Von der Wurzel bis zu einem beliebigen Blatt gibt es immer gleich viele schwarze Kanten! Und da nie zwei rote Kanten hintereinander kommen können, ist die maximale Tiefe niemals größer als das Doppelte der minimalen Tiefe.





2) B-Bäume (zeichnen)

Gegeben ist folgender Mehrwegbaum vom Grad 2. Was müssen Sie überprüfen, um zu prüfen, ob dieser Mehrwegbaum ein B-Baum vom Grad 2 ist? Ist dieser Baum ein B-Baum vom Grad 2? Wenn es kein B-Baum ist, strukturieren Sie ihn in einen B-Baum vom Grad 2 um.



Führen Sie die folgenden Operationen durch (Ex = Einfügen des Schlüssels x, Lx = Löschen des Schlüssels x): E16, E27, E40, E2, E4, E1, L17, L10, L26, E35

Lösung:

1. Jeder Knoten hat mindestens 2, maximal 4 Schlüssel, Wurzel darf weniger haben.
2. Schlüssel sind in jedem Knoten aufsteigend.
3. Für jeden Schlüssel k in einem inneren Knoten gilt:
 - 3.1. Die linken und rechten Teilbäume ist nicht leer, d.h. ein innerer Knoten mit i Schlüsseln hat $i+1$ Söhne.
 - 3.2. Alle Schlüssel im linken Teilbaum sind kleiner als k .
 - 3.3. Alle Schlüssel im rechten Teilbaum sind größer als k .
4. Alle Blätter stehen auf selber Stufe.

Ist kein B-Baum, weil Prüfung 2 nicht OK ist.

Nach dem Vertauschen der Schlüssel 17 und 15 ist es immer noch kein B-Baum vom Grad 2, weil der rechte Teilbaum des Schlüssels 30 leer ist (Prüfung 3.1).

Man erhält einen gültigen B-Baum vom Grad 2, wenn man dieselben Schlüssel in einen anfangs leeren B-Baum einfügt (siehe unten, man kann aber auch von einem anderen Baum weg starten).

Vorgehensweise beim Einfügen:

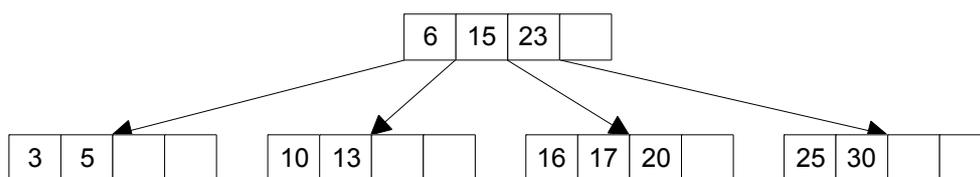
- Zuerst sucht man die Einfügeposition in einem Block b :
 - Wenn b ein Blatt ist, wird der Schlüssel in b eingefügt (Überlauf ist möglich).
 - Wenn b ein innerer Knoten ist, wird der Schlüssel rekursiv im entsprechenden Sohn eingefügt, das eigentliche Einfügen erfolgt nur in den Blättern.
- Behandlung des Überlaufs, wenn das Blatt schon voll war ($2n$ Schlüssel):
 - Blatt splitten (links n , rechts n Schlüssel), mittleren Schlüssel in Vaterknoten einfügen
 - Auch Vaterknoten kann bereits voll sein => Splitten
 - Splitten der Wurzel führt zu neuer Wurzel mit nur einem Element (Höhe des Baums wächst)

Vorgehensweise beim Löschen:

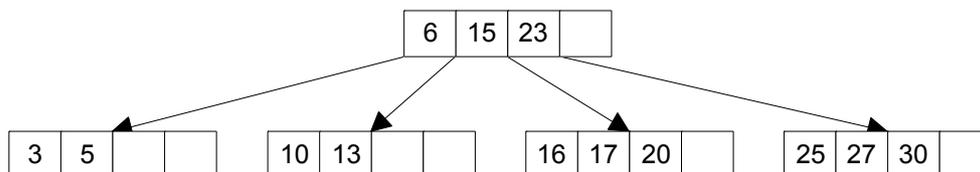
- Zuerst sucht man die Löschposition in einem Block b :
 - Wenn b ein Blatt ist, wird der Schlüssel in b gelöscht (Unterlauf ist möglich).
 - Wenn b ein innerer Knoten ist, wird der zu entfernende Schlüssel durch den nächstkleineren ersetzt, das eigentliche Löschen erfolgt nur in den Blättern.
- Behandlung des Unterlaufs, wenn das Blatt nur n Schlüssel enthielt:
 - Schlüssel mit Nachbarknoten ausgleichen
 - Wenn es trotzdem nicht genügend Schlüssel gibt (weniger als $2n$), Nachbarknoten mit dem Schlüssel aus dem Vaterknoten verschmelzen (ergibt einen vollen Knoten)
 - Wenn letztes Element aus der Wurzel gelöscht wird, schrumpft die Höhe des Baums um 1.

Nur als Beispiel:

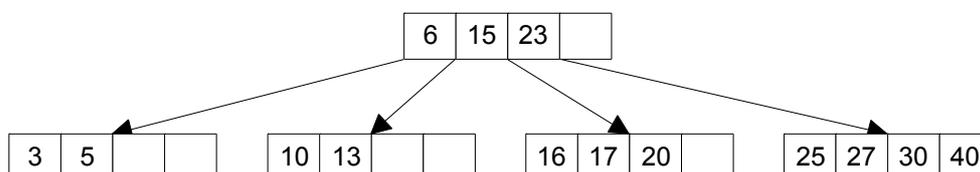
E16:



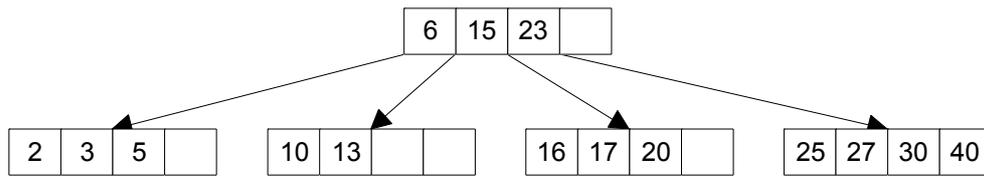
E27:



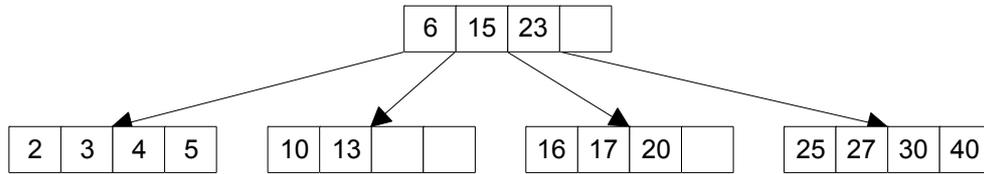
E40:



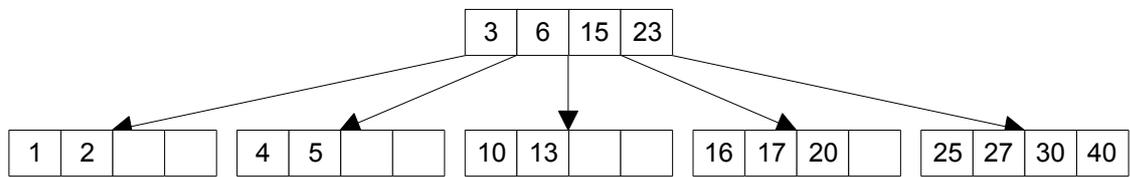
E2:



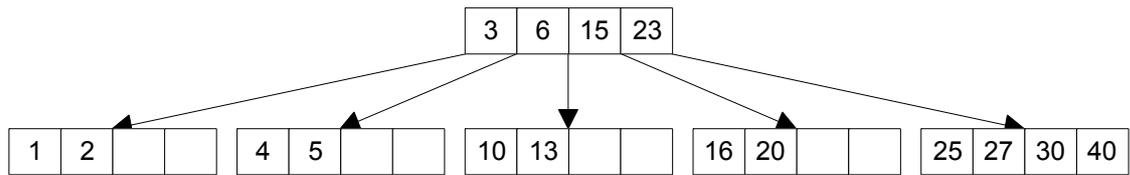
E4:



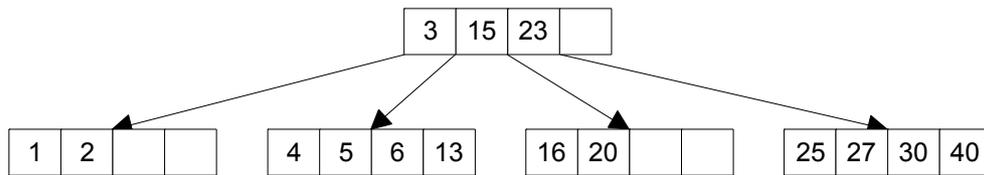
E1:



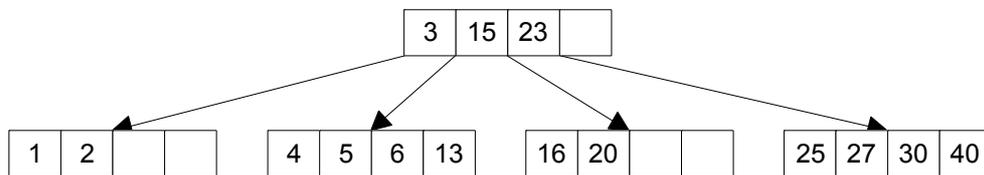
L17:



L10:



L26:



E35:

