

Allgemeine Hinweise:

- Die **Hausaufgaben** sollen in Gruppen von je **2-3 Studierenden** aus der **gleichen Kleingruppenübung (Tutorium)** bearbeitet werden. **Namen und Matrikelnummern** der Studierenden sind auf jedes Blatt der Abgabe zu schreiben. **Heften bzw. tackern Sie die Blätter!**
- Die **Nummer der Übungsgruppe** muss **links oben** auf das **erste Blatt** der Abgabe geschrieben werden. Notieren Sie die Gruppennummer gut sichtbar, damit wir besser sortieren können.
- Die Lösungen müssen **bis Mittwoch, den 3.6.2015 um 12:15 Uhr** in den entsprechenden Übungskasten eingeworfen werden. Sie finden die Kästen am Eingang Halifaxstr. des Informatikzentrums (Ahornstr. 55). Alternativ können Sie die Lösungen auch vor der Abgabefrist direkt bei Ihrer Tutorin/Ihrem Tutor abgeben.
- In Aufgaben, bei denen Sie Algorithmen implementieren sollen, dürfen Sie Ihre Lösung als Pseudo-Code abgeben. Abgaben in verbreiteten imperativen Sprachen wie Java oder C++ sind ebenfalls erlaubt.

Tutoraufgabe 1 (Vollständige Induktion):

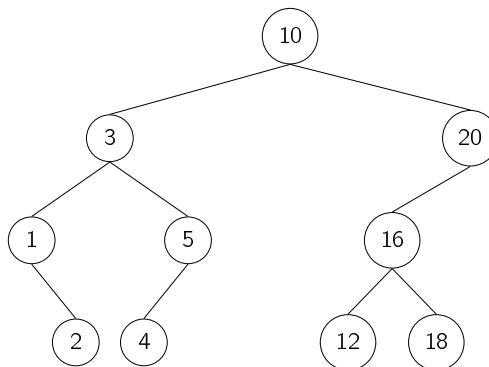
Finden Sie eine geschlossene Form für die Summe

$$\sum_{i=0}^n 2^i$$

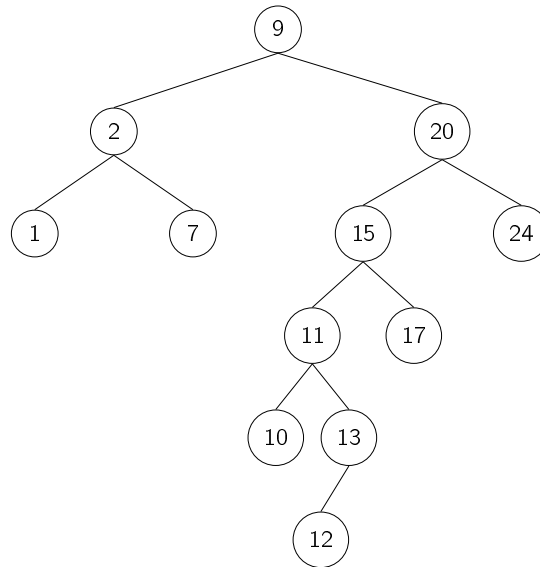
und beweisen Sie mittels eines geeigneten Verfahrens die Korrektheit der geschlossenen Form.

Tutoraufgabe 2 (Rotationen):

- a) Geben Sie für den folgenden binären Suchbaum die Binärbäume an, die entstehen, wenn eine **Rechtsrotation** auf den Knoten mit dem Wert **20** ausgeführt wird, dann eine **Linksrotation** auf den Knoten mit Wert **3** und zuletzt eine **Linksrotation auf die Wurzel** des entstandenen Baumes:



- b) Geben Sie **maximal vier Rotationen** an, die den folgenden Baum der **Höhe fünf** in einen Baum der **Höhe drei** transformieren. Geben Sie auch den Zustand des Baumes nach jeder Rotation an.



Tutoraufgabe 3 (AVL Bäume):

Führen Sie folgenden Operationen beginnend mit einem anfangs leeren *AVL-Baum* aus und geben Sie die entstehenden Bäume nach jeder *Einfüge-* und *Löschooperation* sowie jeder *Rotation* an:

1. 9 einfügen
2. 7 einfügen
3. 8 einfügen
4. 3 einfügen
5. 6 einfügen
6. 2 einfügen
7. 2 löschen
8. 3 löschen

Tutoraufgabe 4 (Rot-Schwarz Bäume):

Führen Sie die folgenden Operationen beginnend mit einem anfangs leeren *Rot-Schwarz-Baum* aus und geben Sie die entstehenden Bäume nach jeder *Einfüge-* und *Löschooperation*, jeder *Rotation* und jeder *Umfärbung* an.

Beachten Sie, dass rote Knoten rund und schwarze Knoten eckig dargestellt werden.

1. 9 einfügen
2. 7 einfügen
3. 8 einfügen
4. 3 einfügen
5. 6 einfügen
6. 2 einfügen
7. 6 löschen

Aufgabe 5 (Baumhöhe vollständiger binärer Bäume):

(3 Punkte)

Wie stellen zunächst einige Begrifflichkeiten klar: Sei B ein Baum mit Wurzel w , und sei v ein beliebiger Knoten von B . Die *Tiefe* von v in B ist definiert als der Abstand von v zu w . Die *Höhe* von B ist definiert als die Tiefe des tiefsten Knotens von B . Die *Ebene* k von B ist die Menge aller Knoten von B der Tiefe k . Sei h die Höhe von B , dann nennen wir B *vollständig*, genau dann wenn B auf jeder Ebene k für $0 \leq k \leq h$ maximal viele Knoten enthält.

Beweisen Sie nun, dass die Höhe eines vollständigen binären Baumes mit n Knoten in $\Theta(\log(n))$ liegt.

Hinweis: Folgende Teilschritte des Beweises könnten sich als hilfreich erweisen:

- Beweisen Sie wie viele Knoten ein vollständiger Baum auf jeder Ebene enthält.
- Beweisen Sie wie viele Knoten ein vollständiger Baum enthält.

Aufgabe 6 (Inorder-Traversierung binärer Suchbäume):

(3 + 3 Punkte)

Es soll bewiesen werden, dass die Inorder-Traversierung alle Schlüssel eines binären Baumes B genau dann in sortierter Reihenfolge ausgibt, wenn B ein Suchbaum ist. Führen sie dazu die folgenden Schritte durch:

- a) Beweisen Sie mittels struktureller Induktion über den Baumaufbau, dass die Inorder-Traversierung eines binären Suchbaumes alle Schlüssel des Baumes in sortierter Reihenfolge ausgibt!
- b) Beweisen Sie, dass ein binärer Baum B ein Suchbaum ist, wenn seine Inorder-Traversierung die Schlüssel in sortierter Reihenfolge ausgibt!

Aufgabe 7 (AVL Bäume):

(7 Punkte)

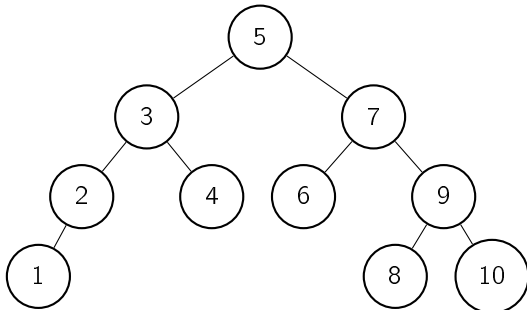
Führen Sie folgenden Operationen beginnend mit einem anfangs leeren *AVL-Baum* aus und geben Sie die entstehenden Bäume nach jeder *Einfüge-* und *Löschoption* sowie jeder *Rotation* an:

1. 3 einfügen
2. 10 einfügen
3. 6 einfügen
4. 1 einfügen
5. 2 einfügen
6. 4 einfügen

Aufgabe 8 (AVL Bäume):

(2 Punkte)

Löschen Sie den Wert 5 aus dem folgenden *AVL-Baum* und geben Sie die entstehenden Bäume nach jeder *Löschooperation* sowie jeder *Rotation* an:



Aufgabe 9 (Rot-Schwarz Bäume):

(7 Punkte)

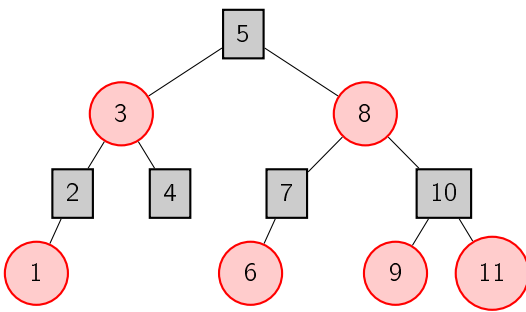
Führen Sie die folgenden Operationen beginnend mit einem anfangs leeren *Rot-Schwarz-Baum* aus und geben Sie die entstehenden Bäume nach jeder *Einfüge-* und *Löschooperation*, jeder *Rotation* und jeder *Umfärbung* an. Beachten Sie, dass rote Knoten rund und schwarze Knoten eckig dargestellt werden.

1. 7 einfügen
2. 0 einfügen
3. 4 einfügen
4. 9 einfügen
5. 8 einfügen
6. 6 einfügen

Aufgabe 10 (Rot-Schwarz Bäume):

(3 Punkte)

Betrachten Sie den folgenden *Rot-Schwarz-Baum*:



Führen Sie beginnend mit diesem Rot-Schwarz-Baum die folgenden Operationen aus und geben Sie die entstehenden Bäume nach jeder *Einfüge-* und *Löschoperation*, jeder *Rotation* und jeder *Umfärbung* an. Beachten Sie, dass rote Knoten rund und schwarze Knoten eckig dargestellt werden.

1. 7 löschen

2. 6 löschen