1. (10 Punkte)
   (a) Zeichnen Sie den Baum auf, der sich ergibt, wenn man nacheinander

   $$2 \quad 31 \quad 5 \quad 29 \quad 9 \quad 25 \quad 13 \quad 20 \quad 17$$

   in dieser Reihenfolge in einen anfangs leeren AVL-Baum einfügt. Bitte zeichnen Sie die Bäume aller Zwischenschritte auf.

   (b) Welche Bäume erhält man, wenn man im eben produzierten Baum den Schlüssel 13 zuerst löscht und dann diesen Schlüssel wieder einfügt?

2. (10 Punkte)
   Geben Sie Pseudo-Code für die in der Vorlesung besprochene Methode BALANCE(KNOTEN X) an, die einen Beinahe-AVL-Baum mit Wurzel x balanciert. Beschreiben Sie dabei insbesondere, wie der Sentinel in Ihrer Methode verwendet wird.

3. (15 Punkte)
   Geben Sie eine Familie von Beispielen an, die zeigt, dass es bei einer Löschoperation in AVL-Bäumen zu logarithmisch vielen Rotationen kommen kann.

4. (15 Punkte)
   Nehmen wir an, Sie hätten einen AVL-Baum T_1, der die Schlüsselmenge A_1, speichert und einen AVL-Baum T_2 für Schlüsselmenge A_2. Dazu gelte noch, dass alle Schlüssel in A_1 kleiner sind als alle Schlüssel in A_2.

   Sie wollen nun einen AVL-Baum T für $A_1 \cup A_2$ erzeugen. Zeigen Sie, dass so ein AVL-Baum T in $O(\log n)$ Zeit aus T_1 und T_2 erzeugt werden kann. Dabei ist n die Anzahl der Schlüssel in $A_1 \cup A_2$, und die Bäume T_1 und T_2 dürfen zerstört werden.

Sonderpunkteaufgabe (30 Punkte)
AVL-Bäume gibt es seit über 50 Jahren. Sie erlauben logarithmische Such- und Updatezeiten. Sie haben allerdings den kleinen Nachteil, dass einzelne Löschoperationen viele Rotationen verursachen können (siehe Aufgabe 3).

Bob Tarjan hat folgende Verallgemeinerung von AVL-Bäumen vorgeschlagen: Ein (Such)-Baum T heißt s-Baum, wenn jedem Knoten v von T eine Zahl $rk(v)$ zugwiesen werden kann, sodass folgende Eigenschaften erfüllt sind. Für jedes Blatt w gilt $rk(w) = 0$. Für jeden Knoten w mit nur einem Kind gilt $rk(w) = 1$. Für jeden Knoten v mit Vater w gilt $rk(w) - rk(v) \in \{1, 2\}$.

- Zeigen Sie, dass jeder AVL-Baum ein s-Baum ist.
- Zeigen Sie, dass ein s-Baum mit n Knoten Höhe $O(\log n)$ hat.
- Entwickeln Sie einen Einfüge- und einen Löschalgorithmus für s-Bäume, der nur Zeit $O(\log n)$ braucht und der jeweils mit höchstens einer einfachen oder einer doppelten Rotation auskommt.