

Aufgabe 1: AVL-Bäume

Fügen Sie die folgenden Zahlen nacheinander in einen *AVL-Baum* ein:

35 58 10 25 32 27

Zeichnen Sie den Baum vor und nach jeder durchgeführten Rotation. Geben Sie auch jeweils an, was für Rotationen Sie durchführen.

Aufgabe 2: AVL-Bäume

Fügen Sie die folgenden Zahlen nacheinander in einen *AVL-Baum* ein:

13 55 77

Zeichnen Sie den Baum vor und nach jeder durchgeführten Rotation. Geben Sie auch jeweils an, was für Rotationen Sie durchführen.

Aufgabe 3: Algorithmen (Entwurf)

Gegeben eine Liste von Zahlen, soll eine Liste von *Leadern* erstellt werden. Ein Element der Liste ist ein Leader, wenn es größer oder gleich allen Elementen rechts von ihm ist. Bestimmen Sie auch die Laufzeit-Komplexität Ihrer Lösung.

Beispiele:

Eingabe: [5, 3, 8, 2, 7, 4]

Ausgabe: [8, 7, 4]

Eingabe: [1, 2, 3, 4, 5]

Ausgabe: [5]

Eingabe: [5, 4, 3, 4, 1]

Ausgabe: [5, 4, 4, 1]

Anmerkung: Ihre Lösung wird sowohl anhand der Korrektheit als auch der Laufzeitkomplexität bewertet.

Hinweis: Sie müssen keinen Code für den Algorithmus angeben, eine Erklärung der Idee genügt. Eine optimale Lösung sollte in $\mathcal{O}(n)$ Zeit laufen. Genauer ist es sogar möglich, die Liste mit einem einzigen Durchlauf zu sortieren.

Aufgabe 4: Algorithmen (Entwurf)

Gegeben eine Liste, die $n - 1$ *verschiedene* Zahlen zwischen 1 und n enthält, soll die Zahl zwischen 1 und n bestimmt werden, die nicht in der Liste enthalten ist. Bestimmen Sie auch die Laufzeit-Komplexität Ihrer Lösung.

Beispiele:

Eingabe: [1, 3, 4, 5, 6]

Ausgabe: 2

Eingabe: [4, 2, 3, 6, 5]

Ausgabe: 1

Anmerkung: Ihre Lösung wird sowohl anhand der Korrektheit als auch der Laufzeitkomplexität bewertet.

Hinweis: Sie müssen keinen Code für den Algorithmus angeben, eine Erklärung der Idee genügt. Eine optimale Lösung sollte in $\mathcal{O}(n)$ Zeit laufen. Genauer ist es sogar möglich, die Liste mit einem einzigen Durchlauf zu sortieren.

Aufgabe 5: Algorithmen (Entwurf)

Gegeben eine Liste von Zahlen sowie eine Zahl n , soll die erste Teilliste der Liste bestimmt werden, deren Summe gleich n ist.

Beispiele:

Eingabe: $[1, 5, 2, 3, 4]$, Ziel: 10

Ausgabe: $[5, 2]$

Eingabe: $[1, 4, 3, 2, 5, 3]$, Ziel: 5

Ausgabe: $[1, 4]$

Eingabe: $[1, 5, 6, 2, 4]$, Ziel: 13

Ausgabe: $[5, 6, 2]$

Eingabe: $[1, 5, 6, 2, 4]$, Ziel: 23

Ausgabe: $[]$

Anmerkung: Ihre Lösung wird sowohl anhand der Korrektheit als auch der Laufzeitkomplexität bewertet.

Hinweis: Sie müssen keinen Code für den Algorithmus angeben, eine Erklärung der Idee genügt. Eine optimale Lösung sollte in $\mathcal{O}(n)$ Zeit laufen. Genauer ist es sogar möglich, die Liste mit einem einzigen Durchlauf zu sortieren.