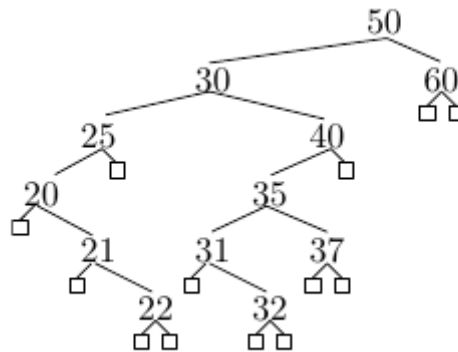


Problem 2: findRange

a) Betrachten Sie den folgenden binären Suchbaum: Wo befinden sich die Schlüssel, die kleiner sind als 37? Wo befinden sich die Schlüssel, die größer sind als 21? Wo befinden sich die Schlüssel, die zwischen 21 und 37 liegen?



Knoten < 37:

- Alle Knoten, die von den 37 Knoten im Baum übrig bleiben (visuell)
- Knoten 30 und der gesamte linke Teilbaum von 30 und der linke Teilbaum von 40 (ohne 37 selbst)
- Knoten < 37 umfassen: {30, 25, 20, 21, 22, 35, 31, 32}

Knoten > 21:

- Alle Knoten rechts vom 21. Knoten im Baum (visuell)
- Alle Vorgänger Knoten des Knoten 21, für die gilt: $n > 21$
- Die rechten Teilbäume dieser Vorgänger
- Knoten > 21: 22, 25, 30, 40, 35, 31, 32, 37, 50, 60

Knoten > 21 und Knoten < 37:

- Alle Knoten, die visuell links von 37 und rechts von 21 sind.
- Knoten im rechten Teilbaum von 21
- Der Knoten 30 und alle Knoten des linken Teilbaums von 30, für die vergoldeten $n > 21$
- Knoten im linken Teilbaum von 40 außer dem Knoten 37 selbst.
- **Knoten > 21 und Knoten < 37:** 22, 25, 30, 31, 32, 35

b) Beschreiben Sie, wie man in einem AVL-Baum mit n Schlüsseln die Operation $\text{findRange}(k_1, k_2)$ implementieren kann, die alle Schlüssel k liefert, für die $k_1 \leq k \leq k_2$ ist. Die Laufzeit soll $O(\log n + s)$ betragen. Dabei ist s die Anzahl der gelieferten Schlüssel.

Idee/Annahme:

- Ein AVL-Baum ist ein balancierter binärer Suchbaum, d.h. für jeden Knoten gilt:
- Alle Schlüssel im linken Teilbaum T_L sind kleiner als der Elternknoten
- Alle Schlüssel im rechten Teilbaum T_R sind größer als der Elternknoten
- Die allgemeine Laufzeit eines AVL-Baums beträgt $O(\log n)$

Herangehensweise:

- Wenn der Schlüssel $k_1 <$ Elternknoten ist gehen wir weiter in den linken Teilbaum T_L , bis wir bei k_1 oder $NULL$ sind.
- Wenn der Schlüssel $k_2 >$ Elternknoten ist gehen wir in den rechten Teilbaum T_R , bis wir bei k_2 oder $NULL$ sind.
- Sollte während des Vergleichs der Knoten: $k_1 \leq \text{Knoten} \leq k_2 \rightarrow$ speichern wir den Wert
- So laufen wir durch den gesamten Baum, bzw. durch alle Knoten die sich innerhalb des Intervalls von $[k_1, k_2]$ befinden.
- Dabei nutzen wir die Eigenschaften eines AVL-Baums aus (das er geordnet ist) und betreten beim Suchen nur die relevanten Teilbäume.