Assignment 6
Advanced Algorithms & Data Structures PS

Christian Müller 1123410
Daniel Kocher, 0926293

April 25, 2016

Aufgabe 11

i.) Fügen Sie die Schlüssel f, g, h, e, b, a, c in einen (anfangs leeren) Treap ein. Die Prioritäten dieser Schlüssel sind wie folgt gegeben: $a : 8$, $b : 15$, $c : 2$, $e : 3$, $f : 7$, $g : 6$, $h : 25$, $i : 22$, $j : 19$, $k : 13$.

ii.) Entfernen Sie e aus dem Treap.

iii.) Fügen Sie die Schlüssel i, j, k in einen anderen (anfangs leeren) Treap ein. Vereinigen Sie anschließend die zwei Treaps.

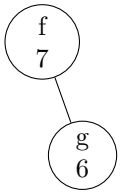
iv.) Führen Sie $Spalte\,(T, d, T_1, T_2)$ durch, wobei T der Treap aus dem vorigen Punkt ist.

Geben Sie den Treap vor und nach jeder Rotation an.

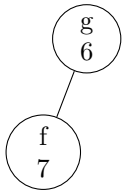
Für den Pseudocode bzw. die grundlegenden Vorgehensweise der Operationen Suchen, Einfügen, Rotiere-NachLinks/Rechts, Entfernen, Vereinige und Spalte, sei auf die Folien vom 14.04.2016 verwiesen.

i.)Fügen Sie die Schlüssel f, g, h, e, b, a, c in einen (anfangs leeren) Treap ein. Die Prioritäten dieser Schlüssel sind wie folgt gegeben: $a : 8$, $b : 15$, $c : 2$, $e : 3$, $f : 7$, $g : 6$, $h : 25$, $i : 22$, $j : 19$, $k : 13$.

Insert g_6:

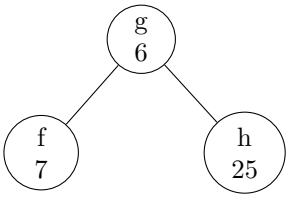


(a) Vorher



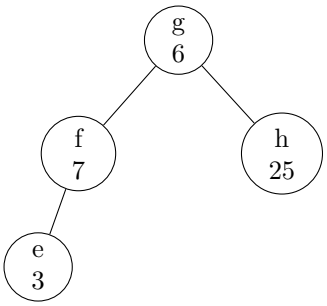
(b) Nachher

Insert h_{25}:

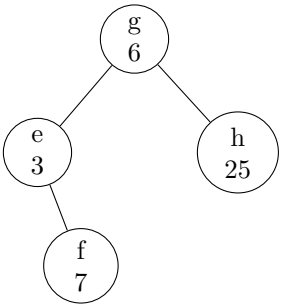


(a) Keine Rotation notwendig

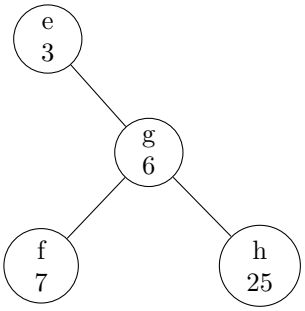
Insert e_3:



(a) Vorher

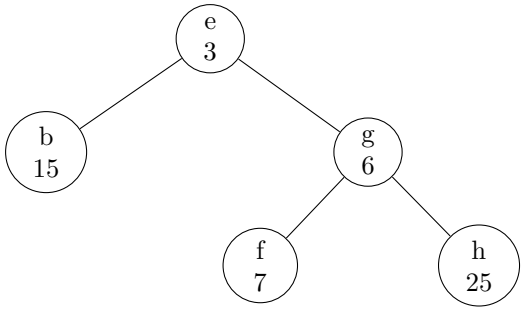


(b) nach: RotiereNachRechts(f_7)



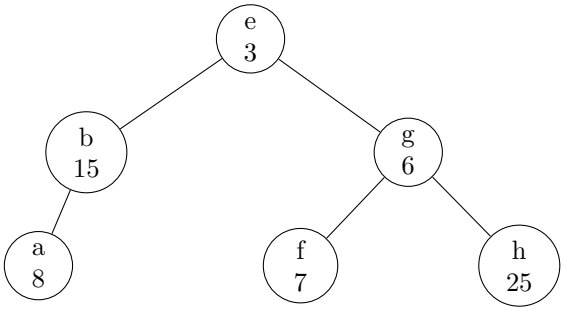
(c) Nachher (nach: RotiereNachRechts(g_6))

Insert b_{15}:

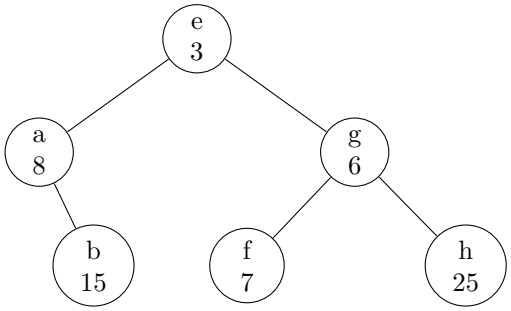


(a) Keine Rotation notwendig

Insert a_8:

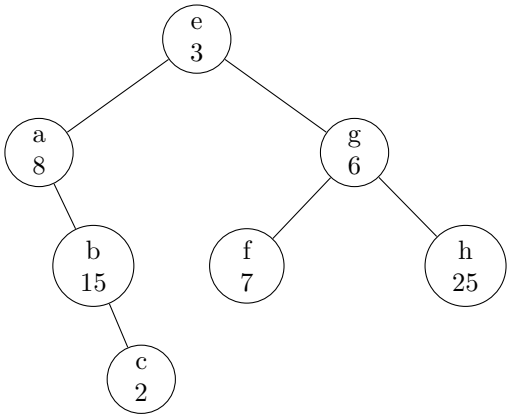


(a) Vorher

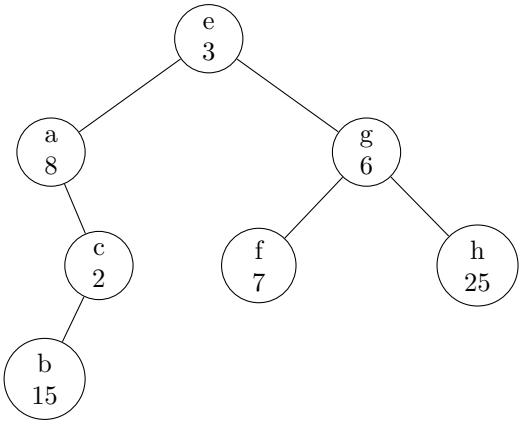


(b) Nachher (nach: RotiereNachRechts(b_{15}))

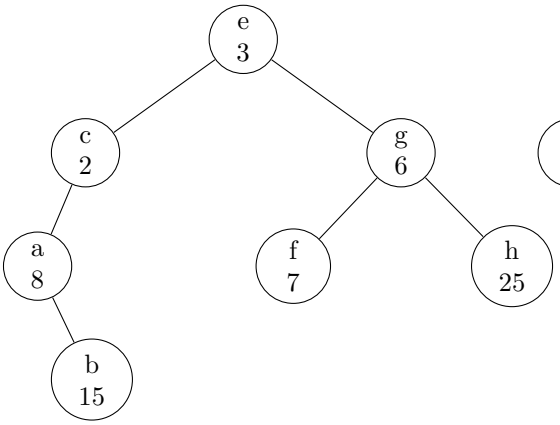
Insert c_2:



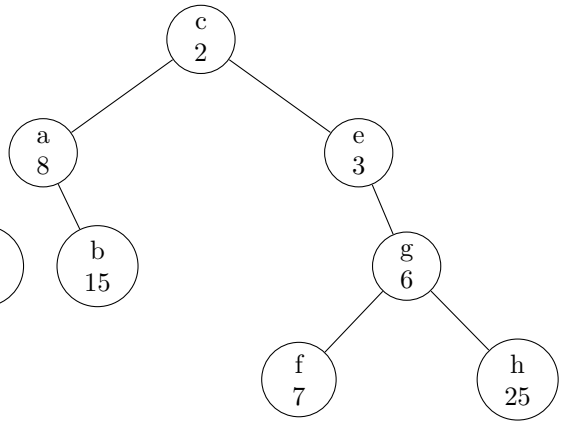
(a) Vorher



(b) nach: RotiereNachLinks(b_{15})

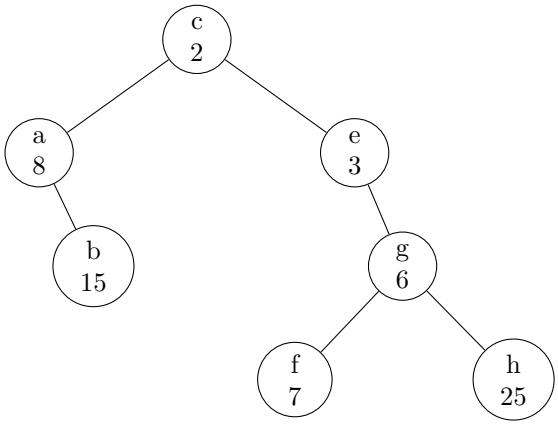


(c) nach: RotiereNachLinks(a_8)

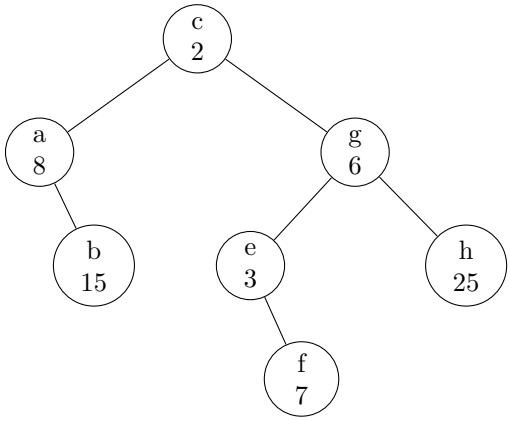


(d) Nachher (nach: RotiereNachRechts(e_3))

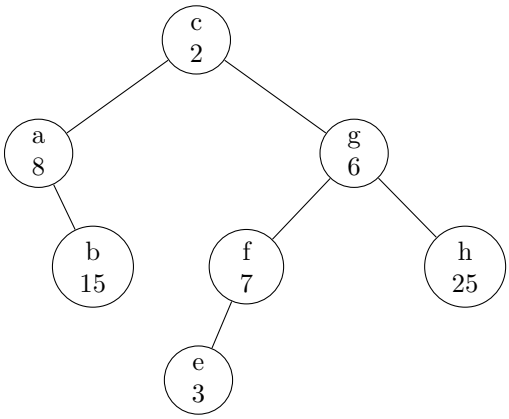
ii.) Entfernen Sie e aus dem Treap.



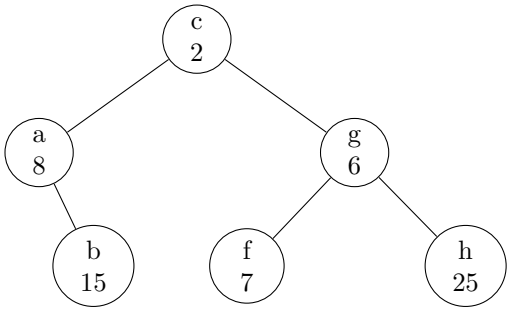
(a) Start



(a) nach: g_6 is rechtes Kind von e_3 $\Longrightarrow$ RotiereNachLinks(e_3)



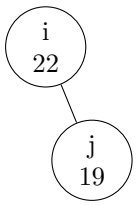
(b) nach: f_7 is rechtes Kind von e_3 $\Longrightarrow$ RotiereNachLinks(e_3)



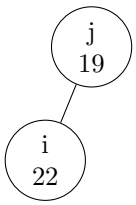
(a) Ende

iii.) Fügen Sie die Schlüssel i_{22}, j_{19}, k_{13} in einen anderen (anfangs leeren) Treap ein. Vereinigen Sie anschließend die zwei Treaps.

Insert j_{19}:

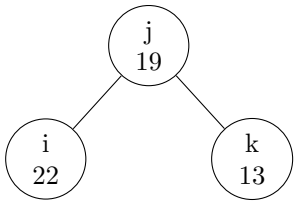


(a) Vorher

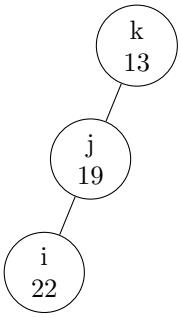


(b) Nachher (nach: RotiereNachLinks(i_{22}))

Insert k_{13}:

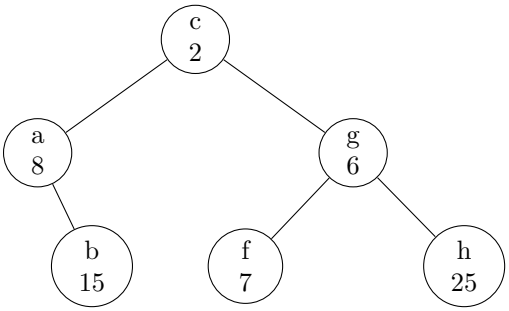


(a) Vorher

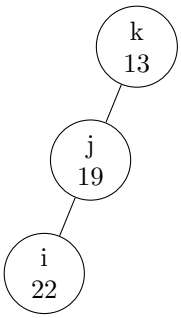


(b) Nachher (nach: RotiereNachLinks(j_{19}))

Vereinige(T_1,T_2):

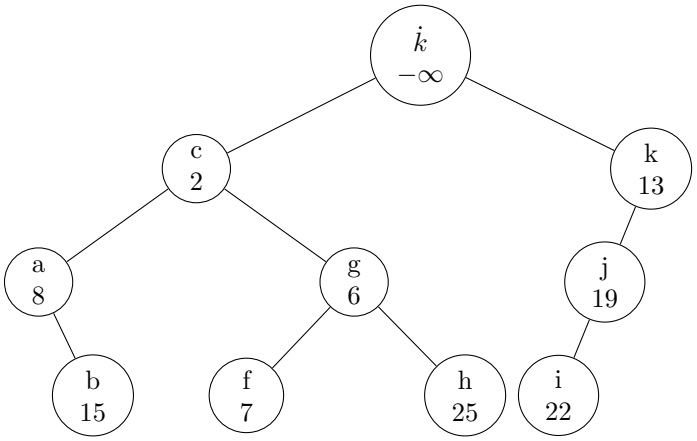


(a) T_1



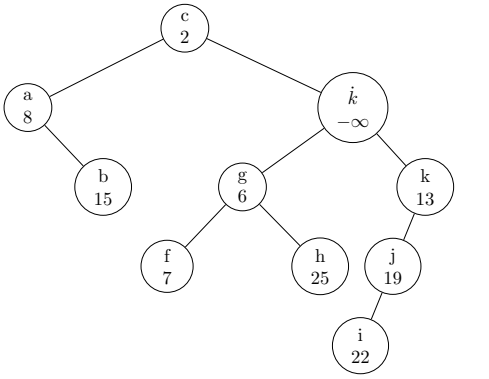
(b) T_2

Sei $\dot{k}$ ein Schlüssel mit $\text{key}(x_1) < \dot{k} < \text{key}(x_2)$ für alle $x_1 \in T_1$ und $x_2 \in T_2$. Ein echter Buchstabe kann hier nicht verwendet werden, da ein solcher im deutschen Alphabet (natürliche Ordnung) nicht existiert. Es gilt also: $a < b < c < ... < \dot{k} < i < j < k < ... < z$.

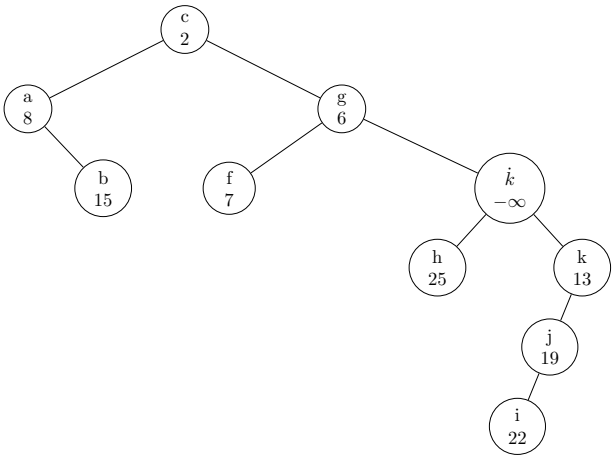


(a) Neuer Knoten mit Schlüssel $\dot{k}$ als Wurzel.

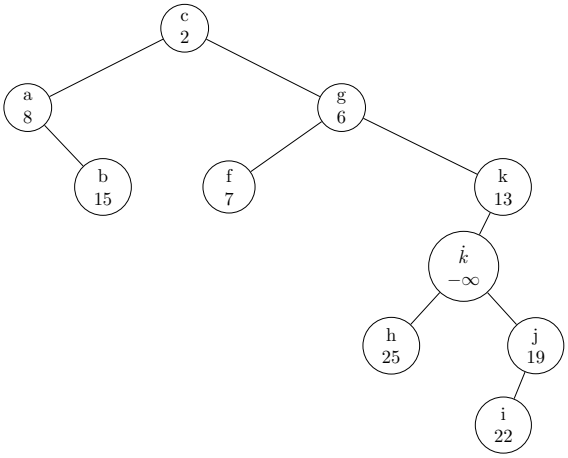
Entferne Wurzel aus Treap:



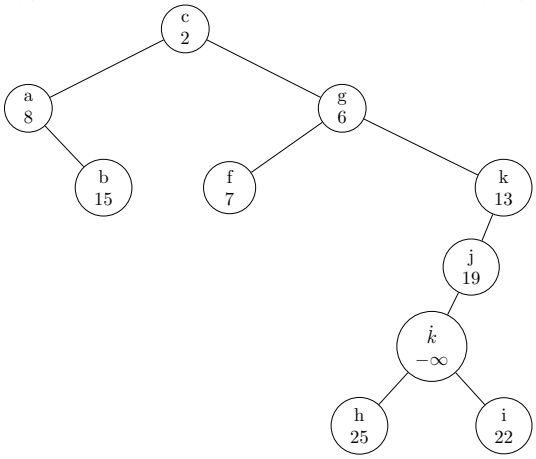
(a) nach: c_2 is linkes Kind von $\dot{k}_{-\infty} \implies$ RotiereNachRechts($\dot{k}_{-\infty}$)



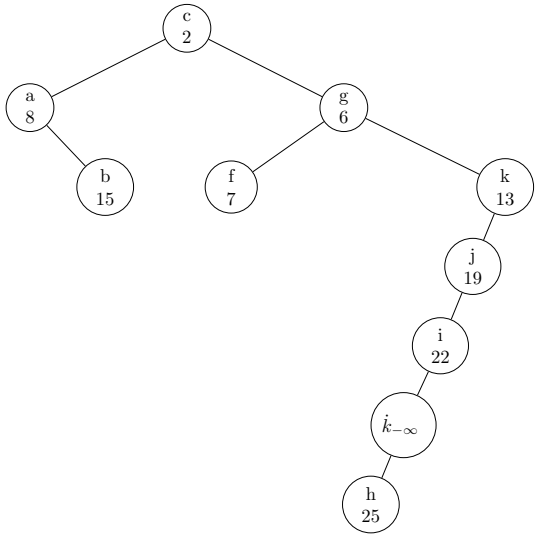
(b) nach: g_6 is linkes Kind von $\dot{k}_{-\infty} \implies$ RotiereNachRechts($\dot{k}_{-\infty}$)



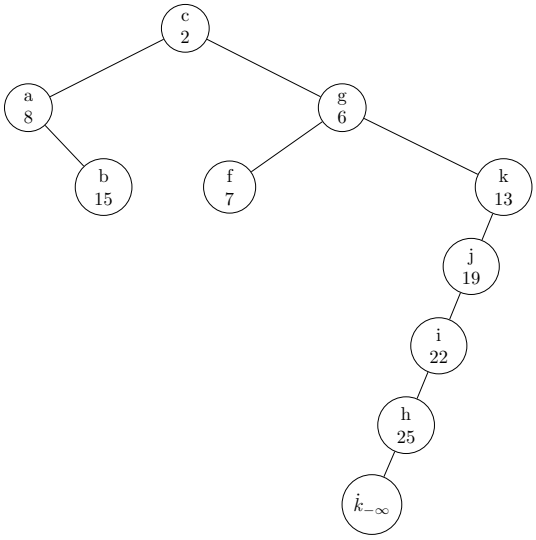
(c) nach: k_{13} is rechtes Kind von $\dot{k}_{-\infty} \implies$ RotiereNachLinks($\dot{k}_{-\infty}$)



(d) nach: j_{19} is rechtes Kind von $\dot{k}_{-\infty} \implies$ RotiereNachLinks($\dot{k}_{-\infty}$)



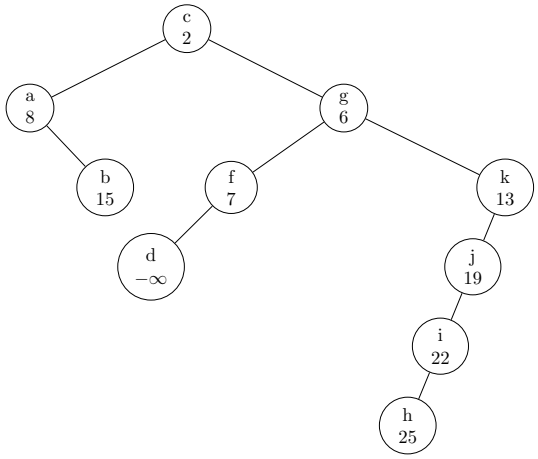
(e) nach: i_{22} is rechtes Kind von $\dot{k}_{-\infty} \implies$ RotiereNachLinks($\dot{k}_{-\infty}$)



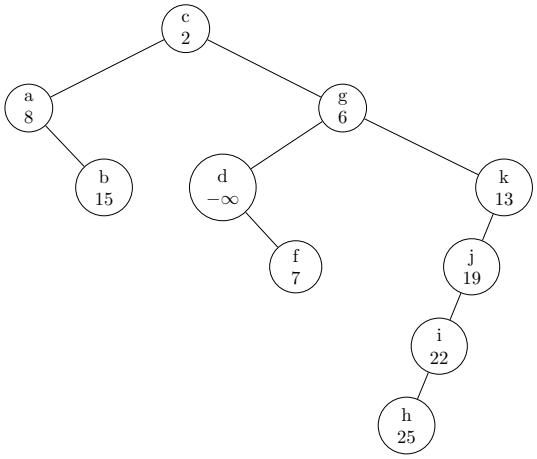
(f) nach: h_{25} ist linkes Kind von $\dot{k}_{-\infty} \implies$ RotiereNachRechts($\dot{k}_{-\infty}$)
Der Hilfsknoten $\dot{k}_{-\infty}$ ist jetzt ein Blatt und kann einfach entfernt werden.

iv.)Führen Sie $Spalte\,(T, d, T_1, T_2)$ durch, wobei T der Treap aus dem vorigen Punkt ist.

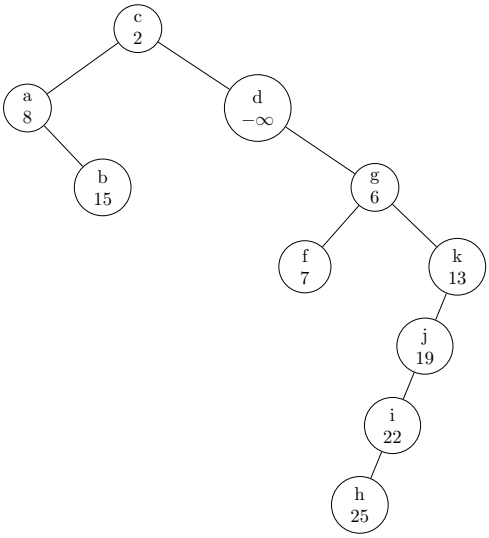
Füge Knoten $d_{-\infty}$ in T ein:



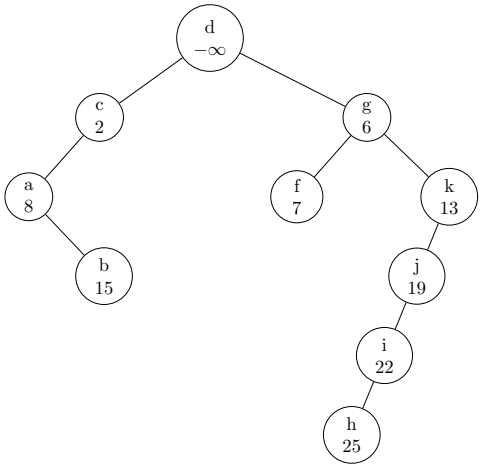
(a) Vorher



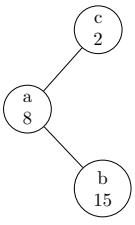
(b) nach: $d_{-\infty}$ ist linkes Kind von $f_7 \implies$ RotiereNachRechts(f_7)



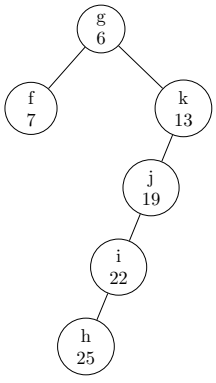
(c) nach: $d_{-\infty}$ ist linkes Kind von $g_6 \implies$ RotiereNachRechts(g_6)



(d) nach: $d_{-\infty}$ ist rechtes Kind von $c_2 \implies$ RotiereNachLinks(c_2)



(e) T_1



(f) T_2

Aufgabe 12

Der *linke Rand* in einem binären Suchbaum T ist der Pfad von der Wurzel zum Knoten mit dem kleinsten Schlüssel. Der *rechte Rand* in einem binären Suchbaum T ist der Pfad von der Wurzel zum Knoten mit dem größten Schlüssel. Betrachten Sie einen Treap T direkt nach dem Einfügen eines Objektes x. Sei C die Länge des rechten Randes des linken Unterbaums des Knotens mit dem Element x und sei D die Länge des linken Randes des rechten Unterbaums des Knotens mit dem Element x. Zeigen Sie, dass die Anzahl der Rotationen, die während des Einfügens von x durchgeführt wurden, $C + D$ ist.

Aufgabe 13

Sei $U = \{0, \ldots, N-1\}$, wobei N eine Primzahl ist und sei $m = 4$. Seien $a_i = 40i$ und $b_i = 60i$. Wir definieren folgende Klasse von Hashfunktionen:

$$H = \left\{ h_i(k) = \left((a_i k + b_i) \mod N - 1 \right) \mod m \right\} \text{ für } i \in \left\{ 1, \ldots, N(N-1) \right\} \tag{1}$$

Ist H universell? Warum? Falls H nicht universell ist, so modifizieren Sie h_i, a_i und b_i, sodass Sie eine universelle Klasse erhalten.