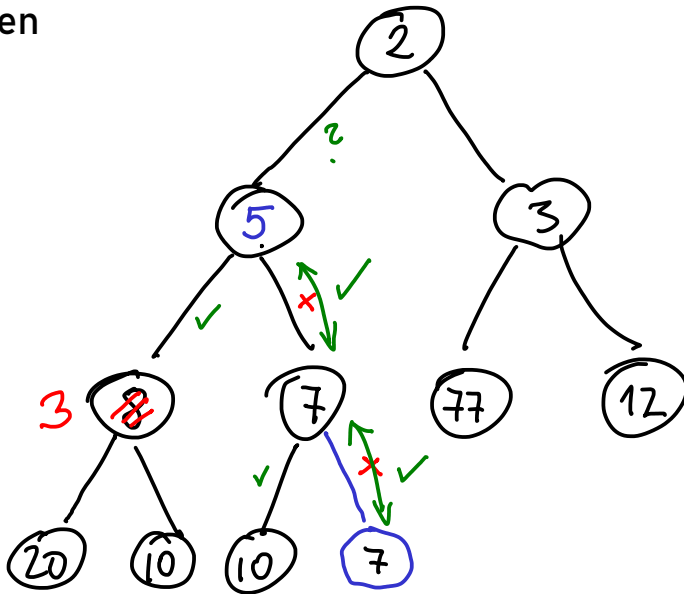


Halden

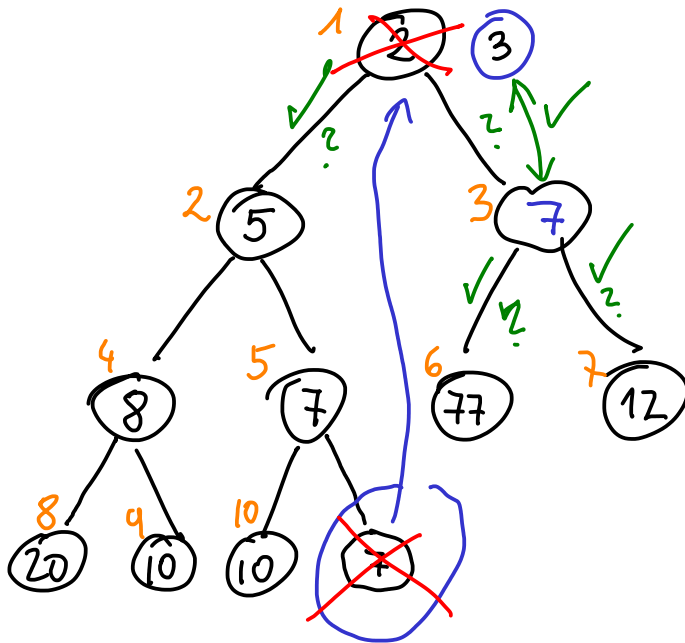


Halden eigenschaft:
Schlüssel eines Knotens
 \leq Schlüssel der Kinder
Minimum in der Wurzel

Einfügen $O(\log n)$

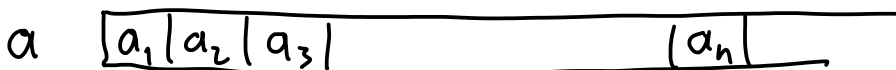
Entferne Min $O(\log n)$

Decrease Key $O(\log n)$



Kinder von a_i : a_{2i}, a_{2i+1}

Eltern von a_i : $a_{\lfloor i/2 \rfloor}$



Prioritätswarteschlange als abstrakter Datentyp

Alg. von Dijkstra:

bestimme u mit $d[u] = \min \{ d[v] \mid v \in U \}$; $U := U \setminus \{u\}$

$u := \text{deleteMin}()$

$d[v] := d[u] + c_{uv}$

→ einfüge (v , $d[v]$)

→ decreasekey (v , $d[v]$)

Halde

$[a_1 | a_2 | a_3 | \dots | a_i | \dots | a_n]$

$v_1 \ v_2 \ v_3$

v_i

Wo steht das Objekt?

einfügen(v) liefert einen Henkel zurück
(ein Objekt eines eigenen Typs). (engl. handle)

Mit dem Henkel kann das Programm später
bei decreasekey sich auf den Knoten v beziehen.

Henkel h :

Schlüssel	s	$d[v]$
Wert	x	v
Position	pos	(Position in der Halde)

Halde

$[h_1 | h_2 | h_3 | \dots | h_i | \dots | h_n]$

← h_i ist ein Henkel

$h[i].pos \equiv i$

Schlüssel $h_i.s$ statt a_i