

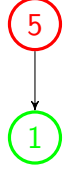
Aufgabe 30a

Zuerst soll unsere Halde (Min-Heap) der Reihe nach mit den vorgegebenen Zahlen gefüllt werden. Hierzu wird jedes Element der Reihe nach hinten an die Halde angefügt. In Schritt 1 wird also die 5 in die Halde gefügt. Nun wird das nächste Element der Liste, also die 1 an die Halde angefügt (Schritt 2). Wichtig ist hier, dass eine Halde immer von oben nach unten und von links nach rechts befüllt wird. Da es sich hier um einen Min-Heap handelt, ist dessen Bedingung verletzt und das Kindelement wird mit dem Elternelement getauscht (Schritt 3), sodass der Heap danach wieder der Bedingung entspricht, dass jedes Elternelement kleiner als seine Kindelemente ist. Nun wird die 4 an den Heap angefügt und da die Heapbedingung nicht verletzt wird, kann die 4 bleiben, wo sie ist (Schritt 4). Danach wird die 2 an den Heap angefügt und verletzt somit die Heapbedingung (Schritt 5), da das Elternelement, also die 5, größer als die 2 ist, werden die beiden Elemente getauscht (Schritt 6). Die 2 ist größer als das Wurzelement und somit ist die Heapbedingung wieder erfüllt. Nun wird die 7 an den Heap angefügt und da die Heapbedingung nach wie vor erfüllt ist, muss nicht getauscht werden (Schritt 7). In Schritt 8 wird die 3 an den Heap angefügt und verletzt somit die Heapbedingung. Deshalb wird in Schritt 9 die 4 mit der 3 getauscht. Im letzten Schritt wird die 6 an den Heap angefügt und da keine Bedingung verletzt ist, kann der Heap so bleiben. Der Heap ist nun vollständig belegt.

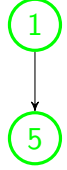
Schritt 1:



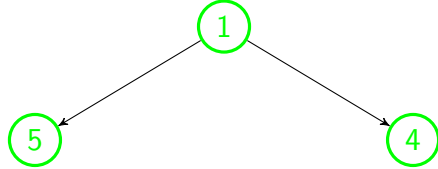
Schritt 2:



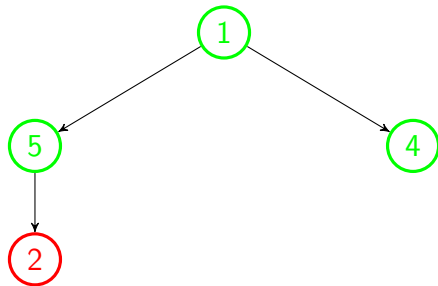
Schritt 3:



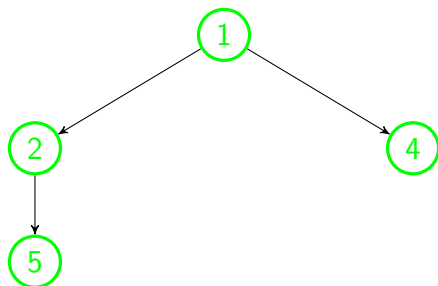
Schritt 4:



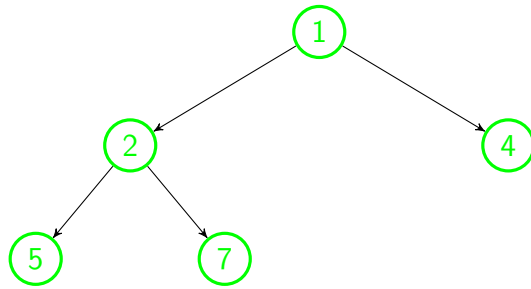
Schritt 5:



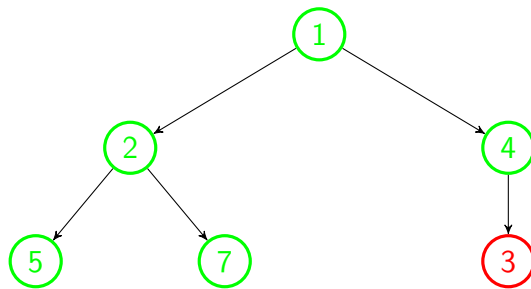
Schritt 6:



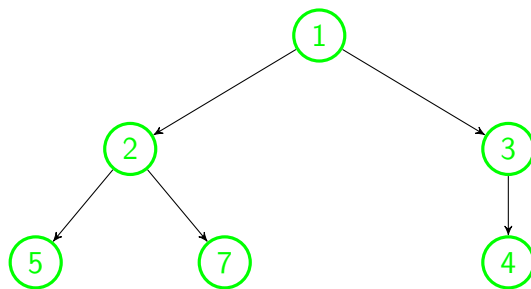
Schritt 7:



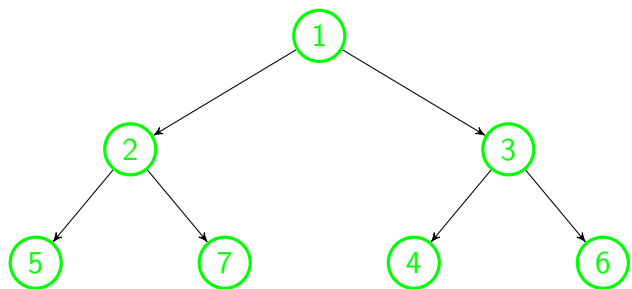
Schritt 8:



Schritt 9:

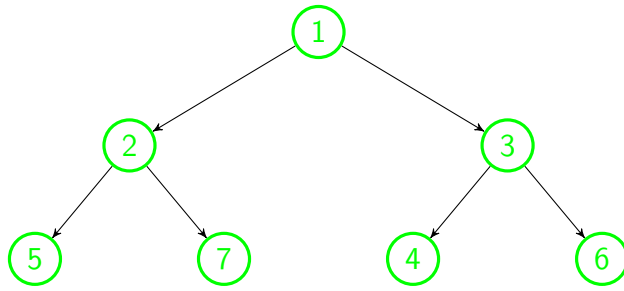


Schritt 10:

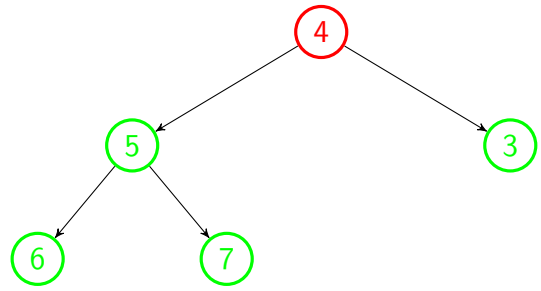


Nun geht es ans Sortieren! Zuerst wird das Wurzelement entfernt und in eine Liste geschrieben. Daraufhin wird das letzte Element des Heaps, also die 6 in Schritt 12 an die Stelle der Wurzel gesetzt. Nun ist die Heapbedingung verletzt und die 6 wird mit dem kleinsten der Kinder des Wurzelements getauscht, also der 2 (Schritt 13). Die Heapbedingung ist nach wie vor verletzt und deshalb wird die 5 mit der 6 getauscht (Schritt 14). Nun ist der Heap wieder richtig und es geht weiter. Nun wird die neue Wurzel, also die 2 aus dem Heap entfernt und an die Liste angefügt, in der bereits die 1 steht. Nun wird in Schritt 15 das letzte Element, also die 4 an die Stelle der Wurzel eingefügt. Nun ist wieder die Heapbedingung verletzt. Daraufhin wird in Schritt 16 die 4 mit der 3 getauscht und der Heap ist wieder richtig. Nun wird das Neue Wurzelement, also die 3, aus dem Heap entfernt und in die Liste gefügt und das letzte Element des Heaps, also die 7, an die Position der Wurzel geschrieben (Schritt 17). Nun ist die Heapbedingung wieder verletzt und es muss mit dem kleinsten Kind getauscht werden. Demnach wird in Schritt 18 die 7 mit der 4 getauscht und der Heap ist wieder richtig. Nun wird die 4 aus dem Heap entfernt und die 6 an die Position der Wurzel geschrieben (Schritt 19). Erneut ist die Heapbedingung verletzt und muss durch Tauschen mit dem kleinsten Kind wiederhergestellt werden. EWs wird demnach mit der 5 getauscht (Schritt 20). Es wird die 5 aus dem Heap entfernt und in die Liste geschrieben und das letzte Element des Heaps, also die 7, an die Position der Wurzel gesetzt (Schritt 21). Da die Heapbedingung weider verletzt ist, muss erneut getauscht werden. Denmach wird die 7 mit der 6 getauscht (Schritt 22) und der Heap ist wieder richtig. Zum Schluss enthält der Heap nur noch die 7, die dann auch entfernt und in die Liste geschrieben wird. Danach hat man eine sortierte Liste und einen leeren Heap.

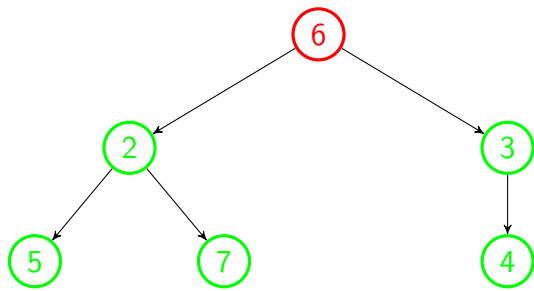
Schritt 11:



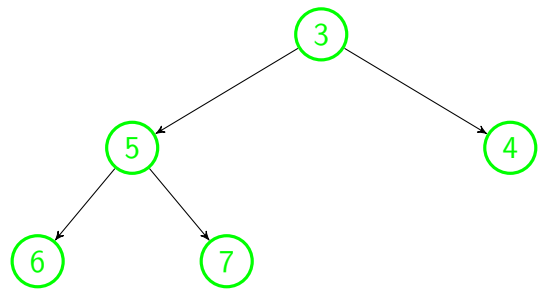
Schritt 15:



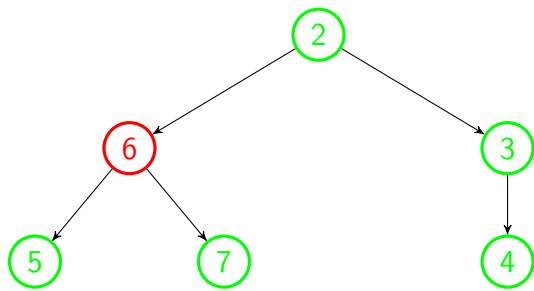
Schritt 12:



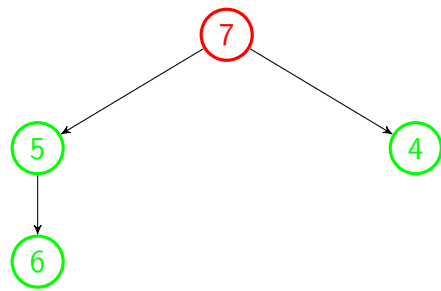
Schritt 16:



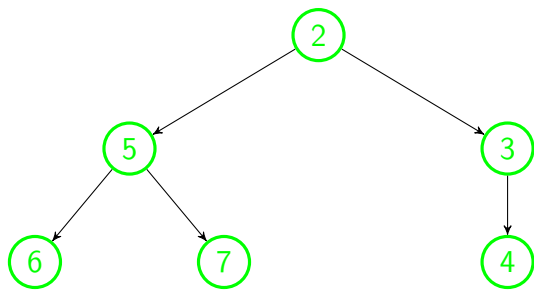
Schritt 13:



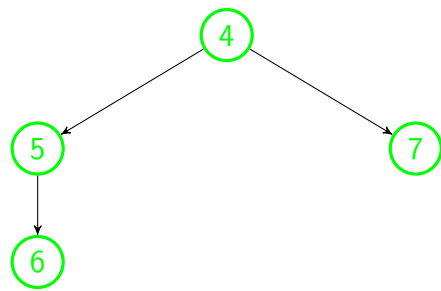
Schritt 17:



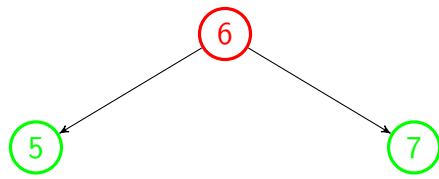
Schritt 14:



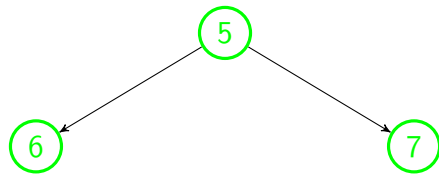
Schritt 18:



Schritt 19:



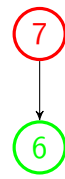
Schritt 20:



Schritt 23:



Schritt 21:



Schritt 22:

