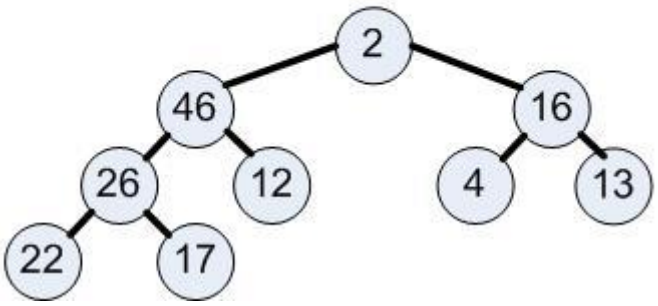
Informatik OTG	Sortieren	
	Heap-Sort	

Heap (Halde, Haufen)

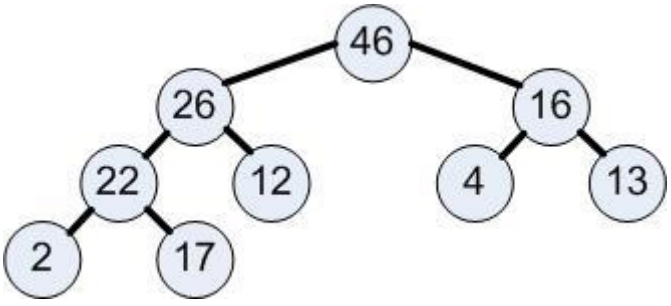
Heap-Sort ist ein höherer Sortieralgorithmus, der aus zwei Schritten besteht.

Vorbereitung: Beim folgenden Beispiel werden die unsortierten Elemente der Schlange in einen anfangs noch leeren Binärbaum eingefügt.
Bei einem Binärbaum gibt es eine Wurzel, jeder Knoten kann bis zu zwei Kindknoten haben. Knoten ohne Kinder nennt man Blätter.

Das erste Element aus der Schlange kommt an die Stelle der Wurzel, dann werden die Knoten von oben nach unten, von links beginnend mit den restlichen Elementen aus der Schlange gefüllt, bis die Schlange leer ist.



Schritt 1: Nun wird der **Heap** aufgebaut: In einem Heap ist der Wert der Söhne jedes Knotens immer kleiner als der Wert des Knotens selbst. Der größte Wert steht stets in der Wurzel (im Beispiel 46).



Hier ist also der Max-Heap mit dem größten Wert in der Wurzel.

Das Element an der Wurzel wird nun entnommen, es ist das erste sortierte Element in der Schlange mit den sortierten Elementen.

Schritt 2: An die nun freie Stelle an der Wurzel rückt das Element am letzten Blatt der binären Baums. Das ist im Beispiel die 17.

Die 17 steht nun an der Wurzel. Jetzt wird wieder der Heap aufgebaut (im Baum sortiert). Das Element an der Wurzel wandert in die Schlange mit den sortierten Elementen. Das Element an der Stelle des letzten Blattes kommt in die Wurzel usw., bis der Baum leer ist und alle Elemente in der sortierten Schlange stehen.

Hier nochmal eine ähnliche Erklärung:

1. Aus der Eingabefolge von sortierbaren Elementen wird ein binärer Baum konstruiert.
2. In diesem binären Baum wird beim sogenannten Versickern so lange ein Knoten mit dem größeren seiner Söhne vertauscht,
   bis das größte Element an der Wurzel steht. Das Ganze wird Heap genannt..
3. Das Element mit dem kleinsten Wert, also das letzte, das am tiefsten sitzende Blatt, kommt auf die Stelle der Wurzel.
   Nun wandert das Element von der Wurzel in das Array.
4. Weiter mit 2., bis die Wurzel leer ist.

Wie schnell ist diese Sortierung? (**Komplexität)**

- Aufbau des Heaps in **O(n)** Schritten (heißt: groß O von n, wobei n die Anzahl der zu sortierenden Elemente ist)
- Versickern eines Elements von der Wurzel: **O(n log n)** Schritte
- Insgesamt: **O(n log n)**
- **Worst case (schlimmster Fall)**:
  Die Daten sind weitgehend vorsortiert. **O(n log n)**
  Dazu muss man wissen, dass der Aufbau des Heaps eine schrittweise vollständige Umkehrung der Sortierreihenfolge darstellt.
- **Best case (bester Fall)**:
  Datenfeld (Schlange) liegt bereits in umgekehrt sortierter Weise vor - 1 Vergleich pro Element, keine Zuweisung. Das ist sehr unwahrscheinlich. Oder: Fast alle Daten sind identisch.