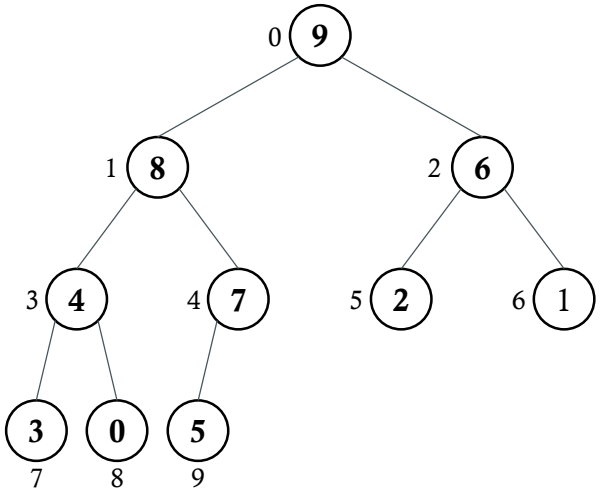
HEAP SORT

- A heap is a complete binary tree



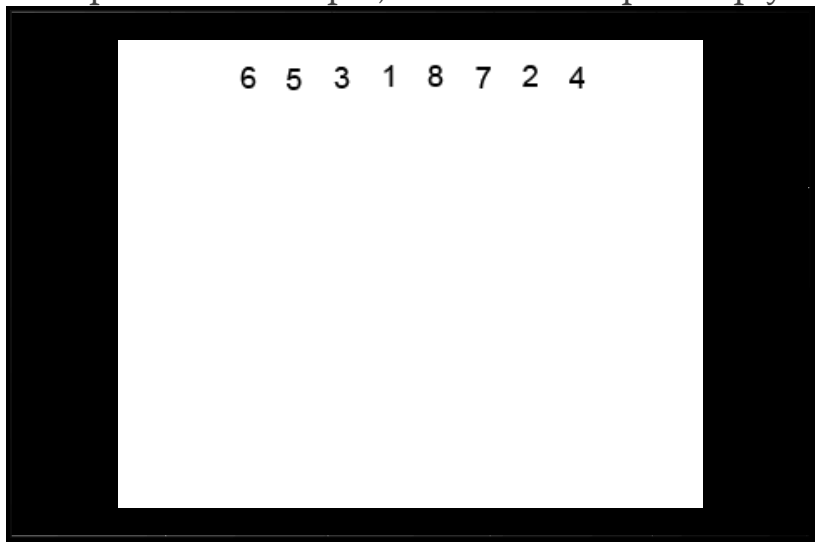
  - `node > lChild && node > rChild` ⇔ **max heap**
  - `node < lChild && node < rChild` ⇔ **min heap**

- An array is a heap

9	8	6	4	7	2	1	3	0	5
0	1	2	3	4	5	6	7	8	9

  - `parent = (node - 1) / 2`
  - `lChild = node * 2 + 1`
  - `rChild = node * 2 + 2`
  - `node != 0 && node % 2 == 1`: (may) has a next sibling
  - `node != 0 && node % 2 == 0`: has a previous sibling

- Heap sort (ascending):
  - Step 1 – build up the heap (max heap)
    - Append an element as the last leaf
    - If `leaf > parent`, then swap it up as new parent. Keep swapping up until root.
  - Step 2 – swap root with the last leaf
  - Step 3 – heapify the rest
    - If the new root is smaller than any of its two child, swap it with the larger child. Keep swapping down until leaf.
  - Step 4 – iterate step 2, 3 until the heap is empty



"Heapsort-example" by Swfung8 - Own work. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Heapsort-example.gif#/media/File:Heapsort-example.gif