Heapsort

Introduction:

Design Technique	**Illustrative Example**
Incremental Approach	Insertion Sort
Divide and Conquer	Merge Sort
Using a DS to manage Information during execution of the Algorithm	Heapsort

- Insertion sort takes $\Theta(n^2)$ time in the worst case and is an in-place sorting algorithm. Merge sort has a better asymptotic running time, Θ (n lg n), but the MERGE procedure it uses does not operate in place.
- Heapsort combines the better attributes of these two sorting algorithms:
    - ➢ Like merge sort, but unlike insertion sort, heapsort's running time is O(n lg n)
    - ➢ Like insertion sort, but unlike merge sort, heapsort sorts in place.

Heaps:

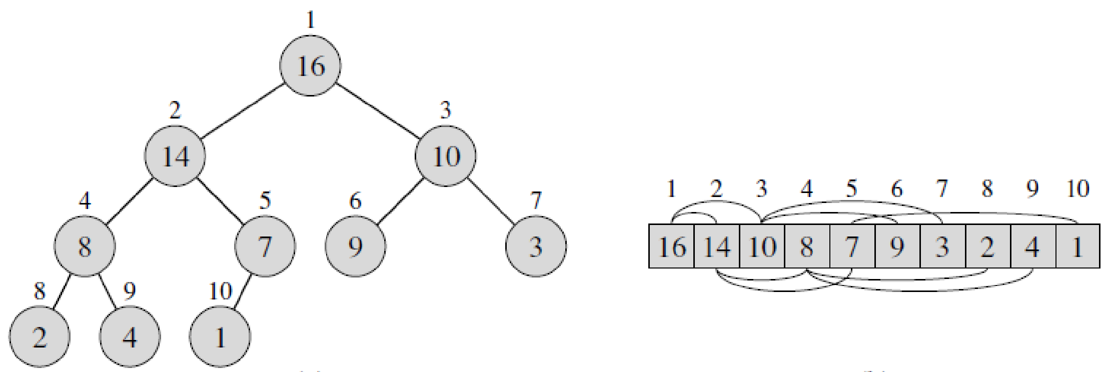
- The (binary) heap data structure is an array object that can be viewed as a nearly complete binary tree.
- Each node of the tree corresponds to an element of the array that stores the value in the node.
- The tree is completely filled on all levels except possibly the lowest, which is filled from the left up to a point.
- An array A that represents a heap is an object with two attributes:
  length[A]: which is the number of elements in the array
  heap-size[A]: the number of elements in the heap stored within array A
  where heap-size[A] $\leq$ length[A],
  The root of the tree is A[1], and given the index i of a node, the indices of its parent PARENT(i), left child LEFT(i), and right child RIGHT(i) can be computed:

PARENT(i)
    return $\lfloor i/2 \rfloor$

LEFT(i)
    return $2i$

RIGHT(i)
    return $2i + 1$

Eg:



(a) (b)

Here,

- A max-heap viewed as (a) a binary tree and (b) an array.
- The number within the circle at each node in the tree is the value stored at that node.
- The number above a node is the corresponding index in the array.
- Above and below the array are lines showing parent-child relationships; parents are always to the left of their children.
- The tree has height three

- There are two kinds of binary heaps: max-heaps and min-heaps. In both kinds, the values in the nodes satisfy a heap property.
- In a max-heap, the max-heap property is that for every node i other than the root,

$$A[\text{PARENT}(i)] \geq A[i]$$

(⇒ the value of a node is at most the value of its parent.)

Thus, the largest element in a max-heap is stored at the root, and the subtree rooted at a node contains values no larger than that contained at the node itself.

- A min-heap is organized in the opposite way; the min-heap property is that for every node i other than the root,

$$A[\text{PARENT}(i)] \leq A[i]$$

- For the heapsort algorithm, we use max-heaps. Min-heaps are commonly used in priority queues.