Tutoring - Heapsort Review

Background Info

- What is a Binary Tree?
  - a tree data structure in which each node can have at most two children, which are referred to as the left child and the right child.
- What is a Heap?
  - a special tree-shaped data structure in which the tree is considered complete (a binary tree in which all the levels are completely filled except possibly the lowest one, which is filled from the left).
  - I must also satisfy a Heap property (Max-Heap, Min-Heap)
- **Max Heap:** any given node is always greater than its child node/s and the key of the root node is the largest among all other nodes.
- **Min Heap:** any given node is always smaller than its child node/s and the key of the root node is the smallest among all other nodes.

Best & Worst Case Time Complexity: $O(n\log n)$

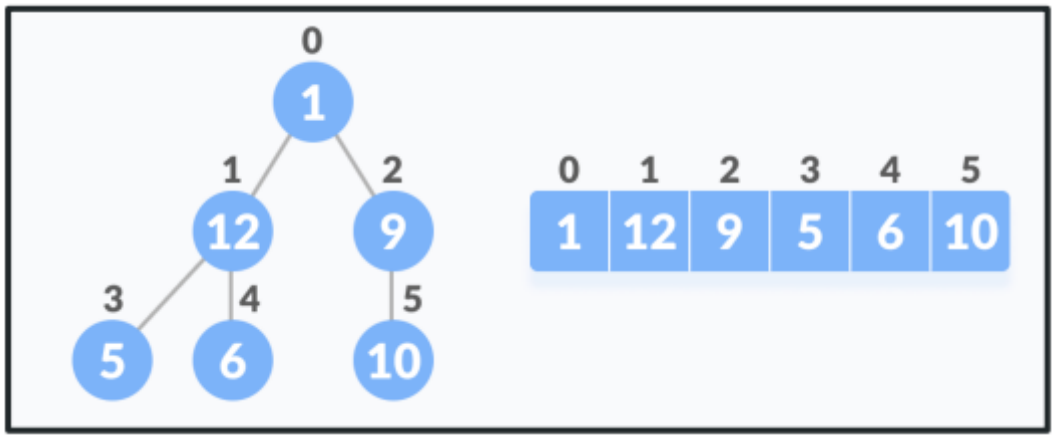
Building a Tree from Array

To build a complete binary tree from an array, we simply start filling in nodes from left to right. The first node being the root, the next two being the children, the next 4 being the grand children, etc.

To put another way, for any index *i* in an array:

- Parent index: $(i-1)/2$
- Left Child index: $2i+1$
- Right Child index: $2i+2$

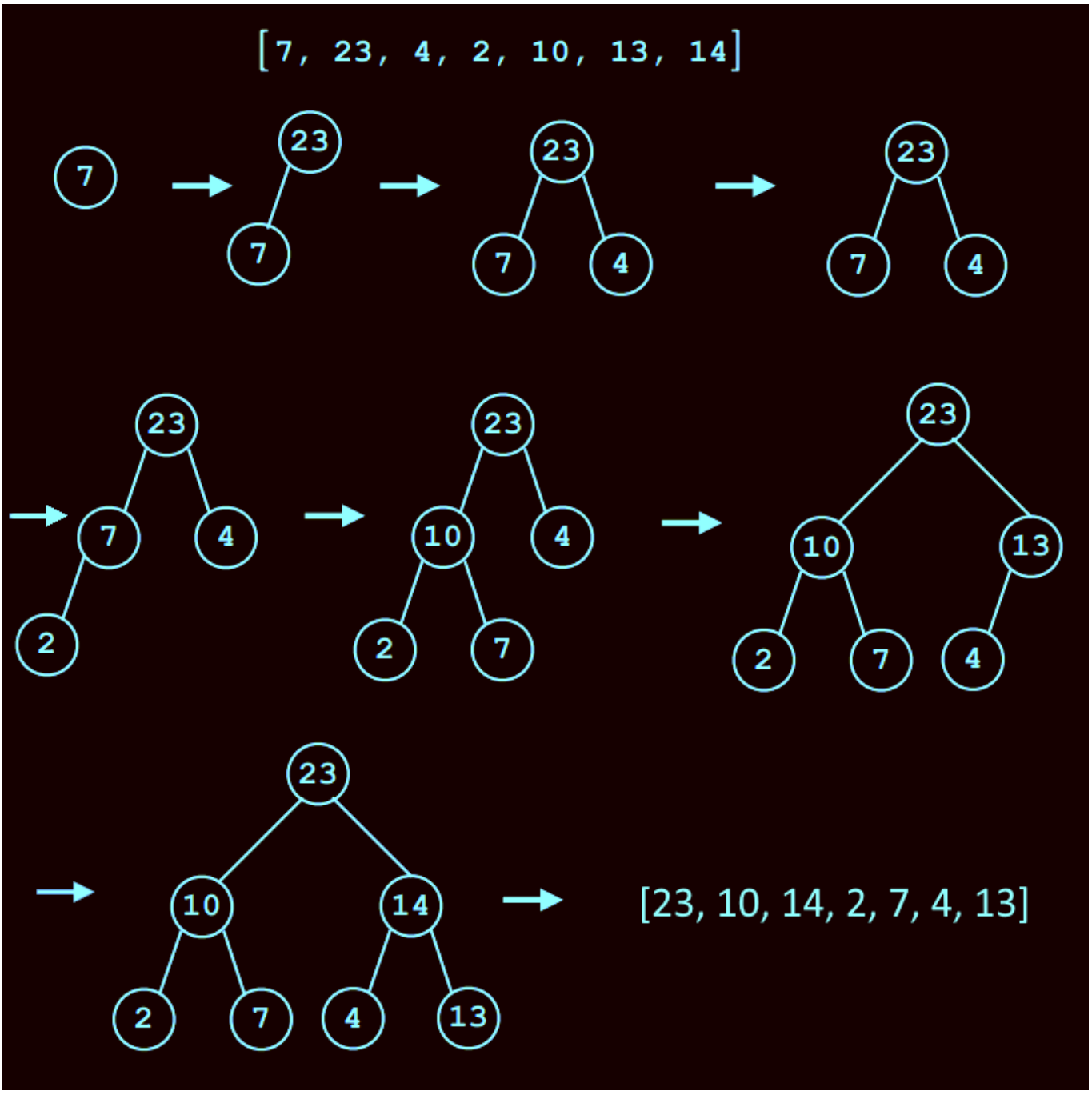
Example:



Heapify & Building Heaps

We turn an unsorted array into a heap, we make sure that this heap satisfies its property (either max or min heap) by calling heapify on the nodes we insert into the heap.

Here is an example with max-heapify:



Heapsort

Heapsort involves:

- create the heap
- swap the sorted (top) node with the last node
- remove the last node from the heap (this is now in its correct place in the array)

- call heapify again on the root node
- repeat until all values have been sorted

If you want to sort in ascending order, use a max-heap and if you want to sort in descending order use a min-heap.

External Resources:

- Heap sort in 4 minutes
- Heap Sort Algorithm