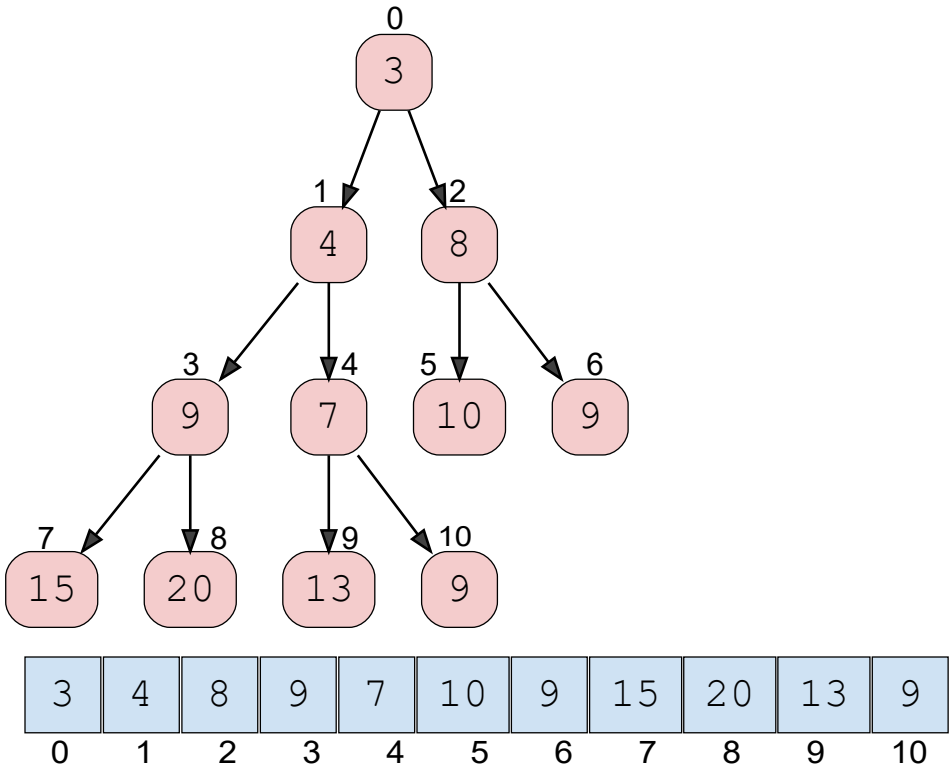
Introduction to Heap Sort

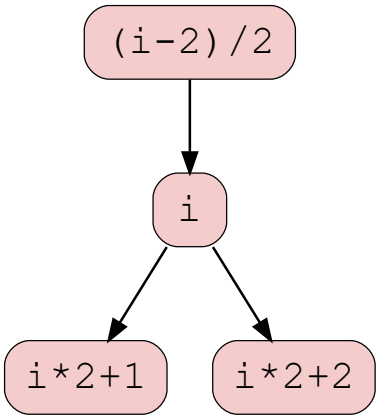
This algorithm creates a heap to sort the elements of an array. (Reading time: under 3 minutes)

In a heap, all nodes are stored based on the value of their parent node.

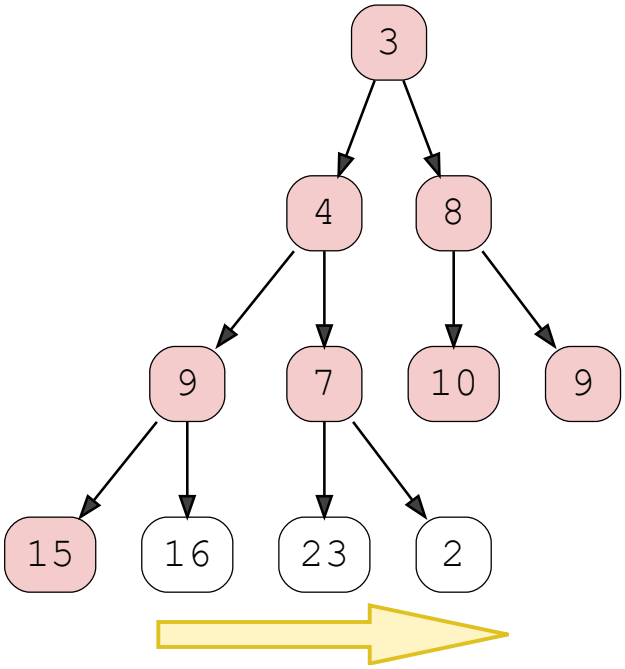


This is a **minimum heap**: the children of a node are always smaller than or equal to their parent. In a maximum heap, the children of a node are always bigger or equal to their parent. It is not sorted yet!

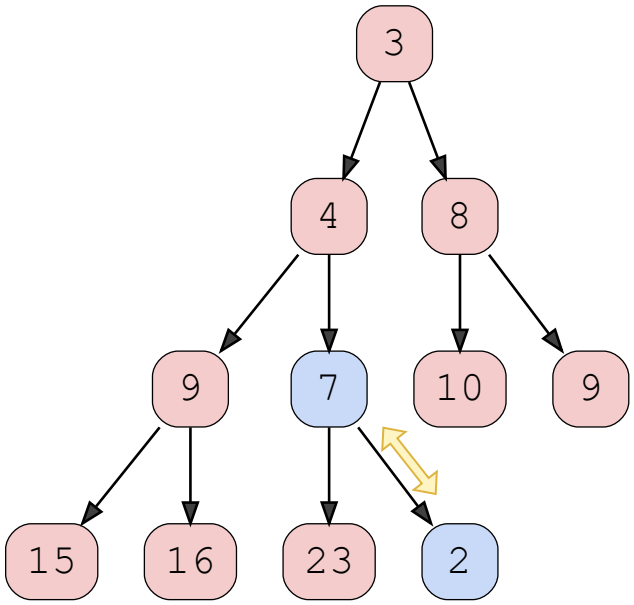
With a heap, you can easily find a node's parent node and children nodes, based on its index in the array.



Whenever you add new nodes, they're always added in the same way: they get added to the first free spot, from left to right. You don't specify a node's parent in a heap.

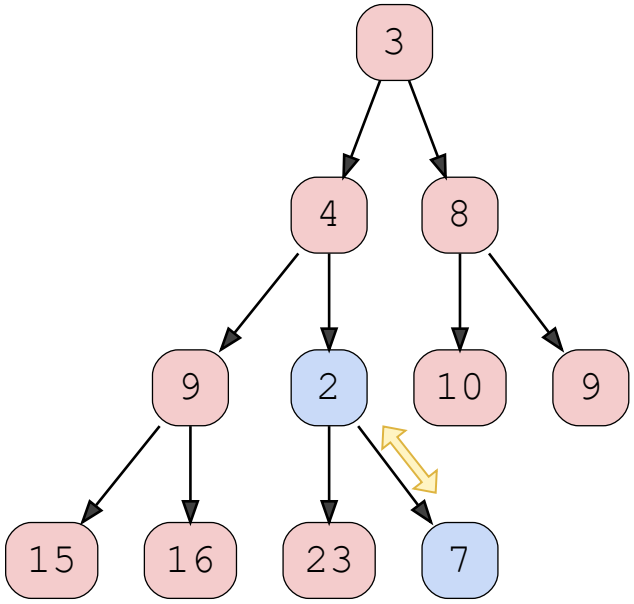


However, this means that the added node might not be in the right place, just like here with the number 2. In order to solve this, we compare the node with its parent node. If the node's value is smaller than that of the parent node, we swap them, until the node is in the right position.



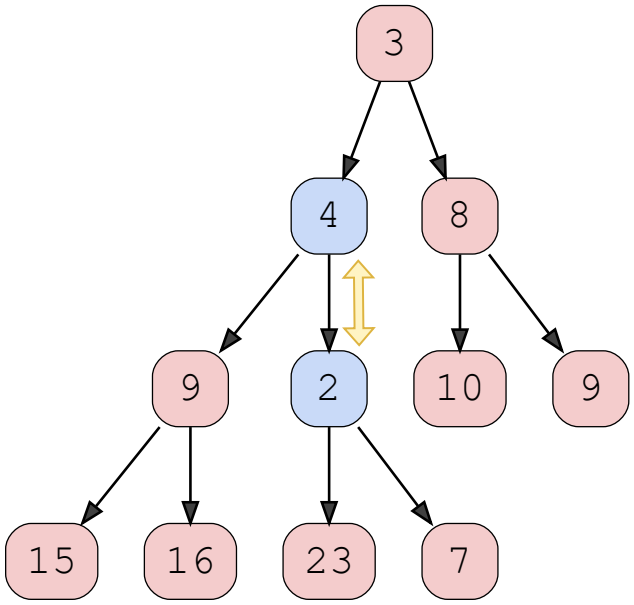
Comparing 2 and 7

1 of 7



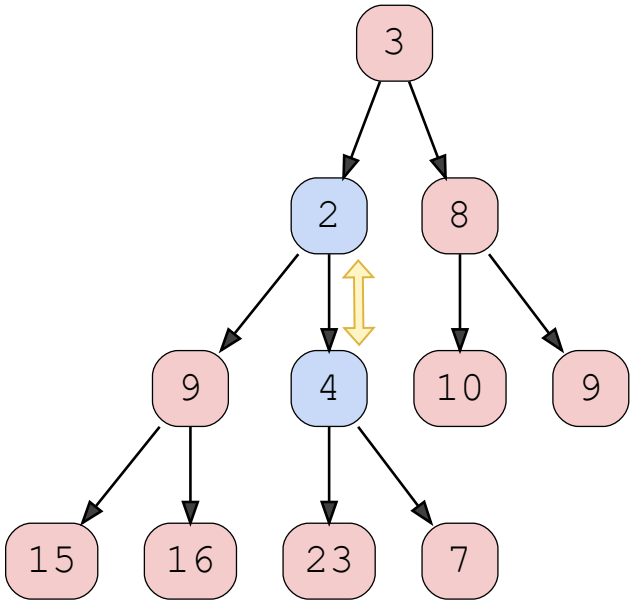
2 < 7, so swap

2 of 7



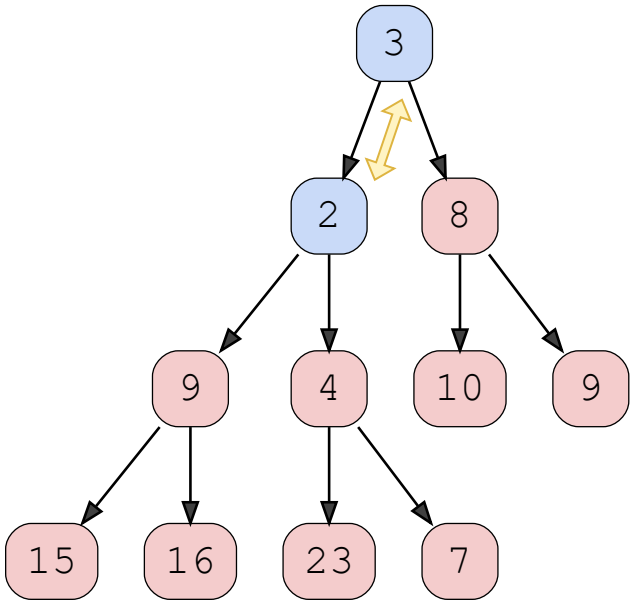
Comparing 2 and 4

3 of 7



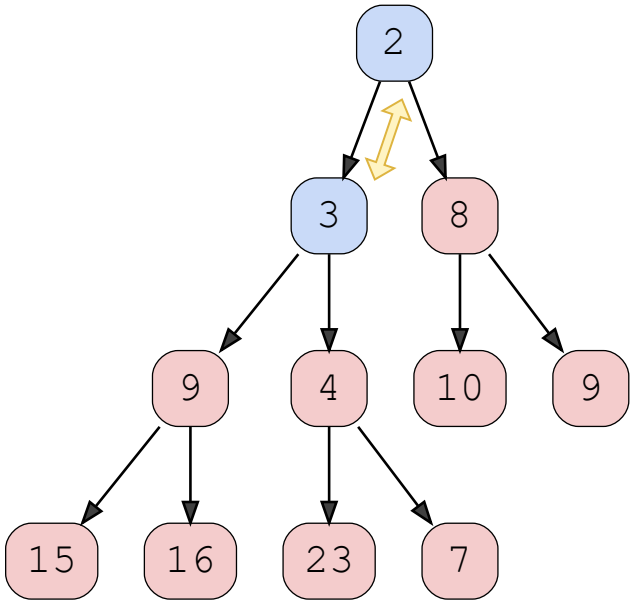
2 < 4, so swap

4 of 7



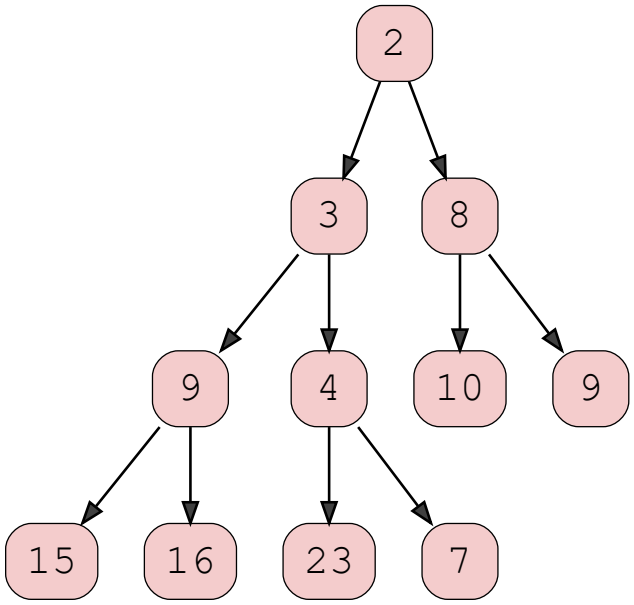
Comparing 2 and 3

5 of 7



2 < 3, so swap

6 of 7



Sorted!

7 of 7

Now, let's discuss the implementation of this algorithm.