Shape Properties of Heap
1. The tree is as short as possible for the given degree.
2. All leaves at the maximum depth are as far to the left as possible.

Order Properties of Heap
1. Every node's priority is greater than or equal to each of its children.
   == Every node's priority is less than or equal to that of its parent.
   In Max-Heaps larger values correspond to higher priorities.
   In Min-Heaps smaller values correspond to higher priorities.

parent0(i) = (i - 1)/2
leftChild0(i) = 2i + 1
rightChild0(i) = 2i + 2

parent1(i) = i / 2
leftChild1(i) = 2i
rightChild1(i) = 2i + 1

Insert into Heap:
(starting with the heap above)
Add new node to the end of the array.
Recursively enforce order property from there:
   compare new node with its parent
   swap when necessary
after swapping, recurse.

Remove from top of Heap:
(starting with the heap above)
Replace top item with the last one in array.
Recursively enforce order property from top:
   compare item with its children
   swap with best child, when necessary
after swapping, recurse.

Complexity: O( log(n) )