* Heap construction: O(n)
* How to remove the root from a Maxheap:
  + This is the same as removing the largest value in the heap, since the root of a maxheap is the largest value in that heap.
  + Steps:
    - Replace the data in root with the rightmost object on the bottom level of the tree.
    - Then shrink heap size by one (using an array representation of the heap). In other words: remove the rightmost object on the bottom level of the tree.
    - Apply siftdown on the new root.
  + O(log(n))
    - Swapping the data in the nodes is O(1), but performing siftdown is O(log(n))
* Delete an object at a specific location:
  + Steps:
    - Replace the object to be deleted with the last object in the heap (right-most object at the bottom level).
    - Decrease size by one. (for array implementation)
    - Check all ancestors of the pos position. (push-up operation)
      * If value at pos is bigger than its parent, then:
        + Swap value at pos with its parent’s value.
        + Pos = parent(pos).
      * Repeat until pos value is no bigger than its parent or until pos reaches the root (index 0).
    - Siftdown on the pos location.