* Heaps
  + Definition: It is a complete binary tree whose nodes contain comparable objects and are organized as follows:
    - Reminder: A complete binary tree with n levels will be filled (no null child references) in the first n-1 levels.
    - MaxHeap: Object in a node greater than or equal to its descendant objects. In other words: Any given node will contain a value that is >= any other node below it.
    - MinHeap: Object in a node less than or equal to its descendant objects.
  + The root of a maxheap contains the largest object in the heap.
  + The root of a minheap contains the smallest object in the heap.
  + The subtree of any node in a maxheap is also a maxheap.
  + The subtree of any node in a minheap is also a minheap.
  + In this class, we focus on maxheap. Minheap behaves in an analogous fashion.
  + Some operations in a MaxHeap:
    - Void add(Comparable element)
      * After add, it maintains the heap structure.
    - Comparable removeMax()
      * After remove, the rest of the data elements maintain the heap structure.
    - Comparable getMax()
* Implementation of a Maxheap:
  + We use the array implementation of the heap (complete binary tree).
  + LeftChild of index 1 = 2\*(1) + 1
  + RightChild of index 1 = 2\*(1) + 2
  + Parent of index 1 = ((1)-1) / 2
  + Index i is a leaf node if (and only if) i >= n/2, where n is the number of nodes in the heap (length of the array).
* Priority Queue:
* Homework 3: