Team SAK -- Jason Kim, Soojin Choi, Adrian Kloskowski
APCS2 pd8
HW#48 -- Heap o'Trouble
2018-05-14

Algorithm for add()

1) Assuming that we are "storing" this heap in an array, add the new element to the end of the array.
2) From there use the equation floor((newElementIndex - 1) / 2) to get the index of the parent.
3) From there, if the condition of the heap is met and is correct, then stop. If not, swap the two.
4) Repeat steps 2 and 3 until the condition is met or until the new element is at index 0.

Algorithm for remove()

1) First, we must check if the heap is empty or not.
2) If it has elements, we must then swap the root with the furthest right element of the bottom level (we will call that element x).
3) Remove the current furthest right element of the bottom level.
4) We take element x and check if it has children. If it has 2, it will swap with the smaller of the 2 as long as x is still larger. If it has 1, it will swap with that child as long as x is still larger.
5) Keep doing step 4 over and over again until x has no children or the children are larger than x.