1. Worst Case Algorithmic Asymptotic Complexity:
    a. Add : O(logn) – this occurs when the element added becomes the new minimum and must move all the way up to the root node.
    b. DeleteMin : O(logn) – although deleting the the minimum element happens instantaneously, replacing it can be of logarithmic complexity, given that the element that replaces it must work its way back to the bottom of the heap.
    c. DeleteMax : O(n) – This could have been better, but to find the maximum element, my program searches every element in the array. I really only need to search the bottom row, but, regardless, my implementation causes this function to grow with linear complexity.
2. Average Case Algorithmic Asymptotic Complexity:
    a. Add : SAME – on average, this will travel about halfway up to the tree, yielding .5 * log(n) complexity, which simplifies to log(n).
    b. DeleteMin : SAME
    c. DeleteMax : SAME