






Implement Heap Sort with a Min Heap

Now that we can add and remove elements let's see some of the applications heaps can be used for. Heaps are commonly used to implement priority queues because they always store an item of greatest or least value in first position. In addition, they are used to implement a sorting algorithm called heap sort. We'll see how to do this here. Heap sort uses a min heap, the reverse of a max heap. A min heap always stores the element of least value in the root position.

Heap sort works by taking an unsorted array, adding each item in the array into a min heap, and then extracting every item out of the min heap into a new array. The min heap structure ensures that the new array will contain the original items in least to greatest order. This is one of the most efficient sorting algorithms with average and worst case performance of $O(n\log(n))$.

Let's implement heap sort with a min heap. Feel free to adapt your max heap code [here](#). Create an object `MinHeap` with `insert`, `remove`, and `sort` methods. The `sort` method should return an array of all the elements in the min heap sorted from smallest to largest.

	The <code>MinHeap</code> data structure should exist.
	<code>MinHeap</code> should have a method called <code>insert</code> .
	<code>MinHeap</code> should have a method called <code>remove</code> .
	<code>MinHeap</code> should have a method called <code>sort</code> .
	The <code>sort</code> method should return an array containing all items added to the min heap in sorted order.