Assignment 1:

My algorithm didn't completely work. :(

From looking at the algorithm and comparing it to MergeSort, which runs in O(nLogn), the algorithm I wrote would be less efficient (time complexity) as I attempted to utilize insertion sort to sort on every element outside the 'runs'. (n^2) - takes four times as long if the input size doubles

I chose insertion sort because it made the most sense to me, in terms of performing a sort on the elements outside the 'runs'. I was contemplating between using insertion and quicksort, but because I was initially testing with a small array of values, insertion sort was faster, more stable, and required less memory(in this case) Quicksort has extra overhead from the recursive function calls as well. I do acknowledge that if I was testing on a HUGE array of elements, it would be wiser to utilize quick sort.