William Escamilla 12/14/2020

I pledge my honor that I have abided by the Stevens Honor System. SSW HW6

1. Runtime when array in reverse order
    1.1. Insertion Sort: O(n^2)
    1.2. Merge Sort: O(n*log(n))
    1.3. Quick Sort: O(n^2)

2. Sort { 8, 1, 4, 5, 9, 2, 6, 5 }
    2.1. Insertion Sort (Green = sorted):

8	1	4	5	9	2	6	5
1	8	4	5	9	2	6	5
1	4	8	5	9	2	6	5
1	4	5	8	9	2	6	5
1	4	5	8	9	2	6	5
1	2	4	5	8	9	6	5
1	2	4	5	5	6	8	9

    2.2. Merge Sort:

Start:	[8 1 4 5 9 2 6 5]
Split:	[8 1 4 5][9 2 6 5]
Split left:	[8 1][4 5][9 2 6 5]
Split left left:	[8][1][4 5][9 2 6 5]
Sort back up:	[1 8][4 5][9 2 6 5]
Split left right:	[1 8][4][5][9 2 6 5]
Sort back up:	[1 8][4 5][9 2 6 5]
Sort back up:	[1 4 5 8][9 2 6 5]
Split right:	[1 4 5 8][9 2][6 5]
Split right left:	[1 4 5 8][9][2][6 5]
Sort back up:	[1 4 5 8][2 9][6 5]
Split right right:	[1 4 5 8][2 9][6][5]
Sort back up:	[1 4 5 8][2 9][5 6]
Sort back up:	[1 4 5 8][2 5 6 9]
Sort back up:	[1 2 4 5 5 6 8 9]

    2.3. Quick Sort - format (pivot,to) [array]:
    (3, 7) [8 1 4 5 9 2 6 5]

```
(1, 2) [1 4 2]
(0, 1) [1 2]
(5, 7) [5 6 9 8]
(6, 7) [8 9]
[1 2 4 5 5 6 8 9]
```

3. Find 2 sum:

   3.1. O(N^2) Runtime:

```
For each index ‘x’ in array starting at ‘0’
    For each index ‘e’ in array starting at ‘x+1’
        If arr[x] + arr[e] == K - arr[x]
            print( “Yes” + x + e );
            return;
print( “No” );
return;
```

   3.2. O(NlogN) Runtime:

```
Sort the array
Int head = 0;
Int tail = arr.length-1;
while(head != tail) {
    if(arr[head] + arr[tail] < K)
        head++;
    else if(arr[head] + arr[tail] > K)
        tail--;
    else
        print( “Yes” + x + e );
        return;
}
print( “No” );
return;
```