Yang Hu

Professor Russell Lewis

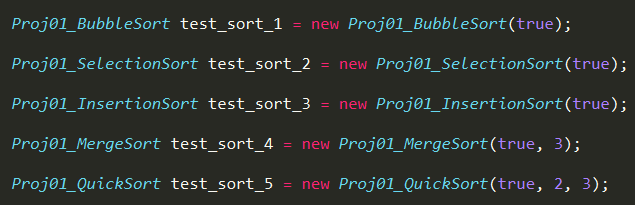
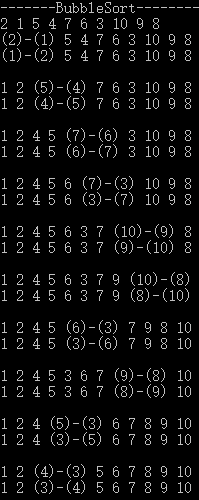
CSC345: Analysis of Discrete Structures

5 September 2018

Assignment 1 Debug Strategy

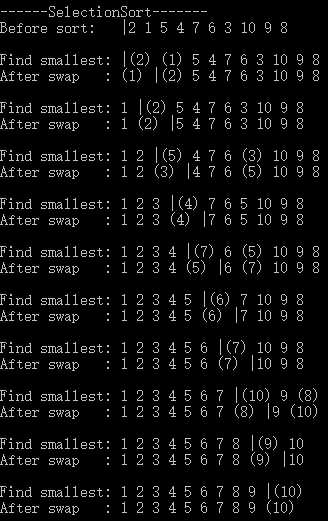
To demonstrate my debugger, I will use identical data to debug all my sorting classes, which is:

**Integer[] arr = {2, 1, 5, 4, 7, 6, 3, 10, 9, 8};**

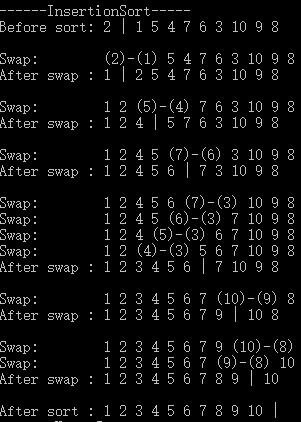
The code I used to initialize sorting classes is:

**Bubble Sort**

My Bubble Sort debugger will print out all of the swaps, decorated by parentheses. All of the two-line output has similar lines with former/later output, which clearly show which pair is swapped, and which pair is going to be swapped.

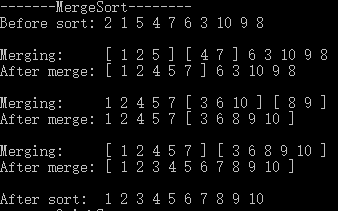
**Selection Sort**

Similar to Bubble Sort, the Selection Sort debugger will show all the swaps by parentheses.

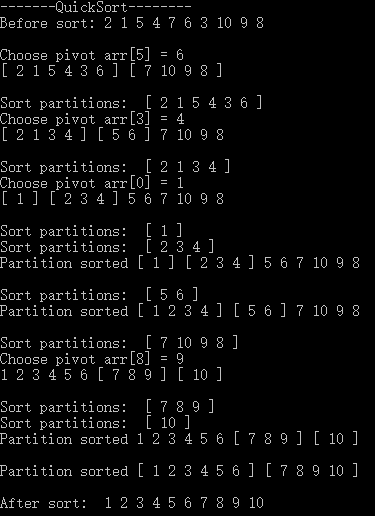
Meanwhile, it will print out the boundary between sorted and unsorted parts.

**Insertion Sort**

This debugger is almost same with Selection sort. Each group of output (Swap and after swap) shows one insertion which insert the smallest unsorted number into the sorted part.

**Merge Sort**

The Merge Sort debugger will show the merging process of the corresponding subarrays.

 **Quick Sort**

Quick Sort debugger is similar to the Merge Sort debugger. It highlights the undergoing parts (pivot, low, high) by square brackets, and prints the pivot before going into partitions.

Besides, debugger will display the O(n^2) sort on small partitions to make process clearer.