**Test Case 1:**

For the first version of the class, I was trying to achieve a few things. My end goal was to have a program with the ability to both add and remove items in a sorted array. My program can determine if the list is sorted, empty, and full. My program is also able to make the list empty. The strengths of this program are that it does what I want it to do. The weakness of this program is that it is slow.

**Test Case 2:**

For the second version of this class, I was trying to achieve essentially the same things as the first class. The only difference was that when adding and removing items, instead of starting from the front of the array, I started from the back. Some people think this method is faster than the first, but I disagree. I think this method averages to be just as fast as the first one.

**Test Case 3:**

For the third version of this class, I was trying to achieve essentially the same things as in the first two classes. The only difference is that when removing an item, I didn’t fill the gaps between items in the array. This made the program faster.

**Test Program 1:**30 items: A screenshot of a computer

Description automatically generated with low confidence Calendar

Description automatically generated with low confidence A picture containing text, calculator, scoreboard

Description automatically generated

10 items: A picture containing text

Description automatically generated A picture containing table

Description automatically generated

**Test Program 2:**

10 items: Calendar

Description automatically generated with medium confidence A picture containing table

Description automatically generated

30 items: A picture containing table

Description automatically generatedA picture containing text, calculator, file

Description automatically generatedA picture containing table

Description automatically generated

A picture containing table

Description automatically generated A screenshot of a computer

Description automatically generated with medium confidence

**Test Program 3:**

10 items: Graphical user interface, text

Description automatically generated

30 items: A picture containing table

Description automatically generatedA picture containing chart

Description automatically generated

**Increase the array to hold 50 (describe results and how they compare with the previous step):**

this array was really big and took a long time to sort for all three times. Although it took a long time, the results did not surprise me. It did exactly what I expected because that’s what the program told it to do.

**Decrease the array to hold 10 (describe results and how they compare with the previous step):**

This was a lot faster than the array of size 50. This is due to there being significantly less items in the array. The results from this test are essentially the same as that in the previous step. The only difference is that there were less items in the array.