**CS302: Homework 3**

An array of sorts.

**Case 1.**

The first assignment was to create a random number generator to be used as the inputs for both sorting algorithms. Users can select the number of elements to be randomized and printed to file.

**Case 2.**

The second portion of the assignment involves testing the efficiency of the algorithms with varying integer amounts. Below are the results of unsorted and sorted tests. All tests were done with a 2015 Macbook Pro using VirtualBox.

1000 unsorted items:

﻿| | Bubble Sort | Merge Sort |

| Times | 0.003349s | 0.000185s |

| Comparisons | 499500 | 9987 |

| Swaps | 241404 | 19974 |

10000 unsorted items:

﻿| | Bubble Sort | Merge Sort |

| Times | 0.434196s | 0.001909s |

| Comparisons | 49995000 | 133631 |

| Swaps | 24671573 | 267262 |

100000 unsorted items:

﻿| | Bubble Sort | Merge Sort |

| Times | 45.955843s | 0.023332s |

| Comparisons | 704982704 | 1668946 |

| Swaps | 2494565889 | 3337892 |

1000 sorted items:

﻿| | Bubble Sort | Merge Sort |

| Times | 0.002477s | 0.000106s |

| Comparisons | 499500 | 9987 |

| Swaps | 0 | 19974 |

10000 sorted items:

﻿| | Bubble Sort | Merge Sort |

| Times | 0.177232s | 0.001169s |

| Comparisons | 49995000 | 133631 |

| Swaps | 0 | 267262 |

100000 sorted items:

﻿| | Bubble Sort | Merge Sort |

| Times | 16.290444s | 0.014442s |

| Comparisons | 704982704 | 1668946 |

| Swaps | 0 | 3337892 |

An odd issue came from reading in file names, but a quick workaround solved that. A rookie mistake I made was not declaring initial values of the class members in the constructor. This caused the max value used to make the array to be a random int instead of what was read in.

This was a fun little assignment overall and helped to cement the logic behind choosing different algorithms as it becomes abundantly clear that there are discrepancies with higher numbers of elements.