# Lab Three – 2210 – Spring 2022

1. Download the project posted with these instructions on D2L.
2. Unzip the folder and open the project by double-clicking the .sln file.
3. Run the project to make sure everything is working correctly.
4. Insert code in the bubble sort method to count the comparisons and the swaps.
5. Run the program and note the number of comparisons and swaps. How does the number compare to the number of elements in the array (in this case 1000)?
6. Insert code in the sequential search method to count the number of comparisons.
7. Modify the call to SequentialSearch in line 17 of the driver to find the first item in the list, run the program, and note the number of comparisons required. **One comparison required.**
8. Modify the call to SequentialSearch in line 17 of the driver to find the last item in the list, run the program, and note the number of comparisons required to find the number. **1000.**
9. Consider the results. What does it tell you about the efficiency of sequential search? **It is very inefficient.**
10. Research merge sort OR quick sort and write an implementation of one of them in the Sorts class.
11. Test your sort by calling it in the driver and verifying the output.
12. Insert code into your new sort implementation that will count the number of comparisons and the number of swaps.
13. Run the program and note the number of comparisons and swaps required. How does it compare to bubble sort?
14. Demonstrate your code to the instructor. Comment your implementation of merge sort or quick sort. Include the answers to the above questions in the comments, including the number of operations for each observation. Submit the Sorts.cs file only to the drop box.