Sorting Exercises

Sorting algorithms range from simple to intricate, and provide an excellent range of examples of problem-solving. This exercise will come in three phases. In the first phase we will build a few of the simplest sorting algorithms. In the second phase we will tackle a few more complex algorithms. The final phase is completing algorithmic analysis and timing tests.

- Bubble Sort **(5 Points)**
- Insertion Sort **(10 Points)**
- Selection Sort **(10 Points)**
- Shell Sort **(15 Points)**
- Merge Sort **(20 Points)**
- Heap Sort* **(10 Points)**
- Quick Sort **(20 Points)**

*(You can use the existing C++ heap for 10 points, or build your own heap for 20 points)

Turning in Sorting code

The code includes everything you should need to complete the work. You can edit the .h files for the individual sorting methods, so long as you do not modify the tester and the tester still builds and runs. When you turn in the final solution include all .h and .cpp files (not the entire project as that is a lot of extra files).