

Objective: Implement a program that uses three different sorting algorithms (Selection, Quicksort, and Mergesort) to compare their behavior.

Activities:

- Create three different data sets (files) of three different magnitudes, (e.g., 100, 1000, 10000) for use by your program.
 1. Random
 2. Sorted—Low to High
 3. Sorted—High to Low

Note: Nine data sets/files total. Many people find it helpful to write helper functions to generate data sets, say one function that returns an array of n random numbers, another that gives an array of n sorted numbers, and so forth.

- **Do not** turn in your data files!
- Display all statistics (comparisons, swaps, etc.) for each sorting algorithm.
- Determine which algorithm works best for each of the data files.

Deliverables:

- A complete functional program with output.
- A summary of your findings.
- A program design sheet. Describe all classes and methods necessary to implement your program. Describe your data sets (files) also.
- Programming Log:
 - Record the time required to design and implement your program.
 - Record of things you encountered/learned while implementing your program.

Some sorting code:

<http://www2.cs.uidaho.edu/~bruceb/cs121/Code/index.html>