Program 3

Due Tue Mar 19 11:59 PM

# Selection Sort and Merge Sort

In ZyBooks, you are given implementations for both Selection Sort and Merge Sort.

Your task is to compare the performance of both sorting algorithms for various problem sizes.

Start by modifying the sort algorithms such that you count the number of relevant operations (See ZyBooks section 9.4). Depending on your environment, you may need to compile with the **-std=c++14** flag (otherwise you may get an error about *nullptr* being undeclared within scope).

Then, for each value of *n* from 2 to 100:

* Perform 10,000 trials where for each trial you:
  + Use **c-style** arrays with *n* random integers between 1 and 1000.
    - Do NOT use the <array> library
    - You may NOT use vectors, other than the vectors already used in merge sort.
    - Be sure to seed your random number generator
  + Sort the same sequence of random numbers using both selection sort and merge sort.
    - You MUST sort the same set numbers with each sort
    - Be sure you do NOT sort an array you have already sorted
* From the 10,000 trials, determine the worst-case number of operations for each sort algorithm
* Save your results to a .csv (comma separated values) file
* Use Excel to create a chart showing the growth of each algorithm as *n* increases
  + Save this chart to a Word document, along with a few paragraphs on whether or not the growth patterns match your expectation given your understanding of the Big-Oh for Selection Sort and Merge Sort.
  + In the word document, you must also write a few paragraphs for each of Selection Sort and Merge Sort explaining how they work. You need to be descriptive enough such that the grader can ascertain that you properly understand the algorithm.

# Implementation Details

Just like Program 2, you will split up your main function, sort function definitions, and sort function prototypes into three files. You will also need to submit your word doc and your executable.

* Main.cpp
* SortFunctions.cpp
* SortFunctions.h
* Summary.docx
* Program3.out

To canvas, you must turn in a file called **<LastName\_FirstInitial>Program3.zip** containing the files listed above.

g++ -o Program3.out Main.cpp SortFunctions.cpp