Lab 8

Submit Assignment

Due Sunday by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip

Searching and sorting

Goals

- Learn about simple searching methods
- Learn about algorithms for sorting

In this lab, we will use algorithms to search data and sort data.

Requirements

Note: The book or the lectures have several examples of searching and sorting algorithms.

Create Data Files

You need to **manually create** at least four text files with random integer values. You don't need large number of values, which makes testing easier.

For example, you can create a file with values: 1 2 3 4 5 6 7 8 9. Save it, and make 3 copies. In each copy put a single value of 0: one towards the beginning of file, one around the middle of file, and one towards the end of file. Rename the 3 copies with name such as "early.txt", "middle.txt", "end.txt". Keep the original one without a 0.

Note: All the text files will be used during the lab. Make sure they are included in submission.

The following is the flow of the program

1. **Read Values from Files**

The program should, for each file, read the file and put the values into an array at the start of program.

2. **Simple Search**

Implement a search algorithm for your program that searches for a target value in your four data files. When implemented, the program should first ask for target value and validate user input, then output whether target value is found for EACH file. Can we use binary search here? Think about it.

Example output

What is target value:

0

num.txt: target value not found

early.txt: target value found

mid.txt: target value found

end.txt: target value found

3. **Sorting**

There are a lot of sorting algorithms; find one and implement it in your program and **cite the sources**. When implemented, the program should, for each file, ask user for the output file name, then use the algorithms to sort the values, and finally **output the sorted values to the output files**. Don't forget to **print the sorted values** in each file onto the screen as well.

4. **Binary Search**

Find an algorithm for binary search. Implement it in the program that searches for target value in your sorted data files from task 3. The operation should be fairly similar to the simple search.

Note: This article about binary search might be interesting to you:

https://research.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html (https://research.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html)

What to submit

- All the program files including header and source files (.cpp/.hpp)
- Makefile

Important: Put all the files in a single .zip file and submit it on Canvas.

Grading

- Programming style: 10%
- Implement and test the searching algorithm: 30%
- Implement and test the sorting algorithm: 30%
- Implement and test the binary search algorithm: 30%

Lab 8

Criteria	Ratings	Pts
Programming Style		10.0 pts
Search Algorithm		30.0 pts
Sort Algorithm		30.0 pts
Binary Search		30.0 pts
		Total Points: 100.0