**Project 1**

**Searching, Step Counts, and Big O Analysis**

This is an individual assignment. You will have lab time to work on this assignment as well as homework time.

Create a project that analyzes the complexity of linear search and binary search using an empirical step count for each. Suggested array sizes will be 200, 400, 600, and 800.

* Create the arrays and assign randomized values for each element in the array. The randomized values should range from 1 to twice the size of the array. For example, for an array of 200, assign random values between 1 and 400. Use a constant for the size of the array. Edit the value of the constant in different runs of the program to account for the various array sizes.
* Perform the step count test for linear search. To get an appropriate average step count for linear search, run the search in a counting loop that executes 1000 times, and divide the total step count for all 1000 searches by 1000. Each search value should use its own random value covering the appropriate range of each array. Display both the total step count of the 1000 searches and the average step count for a single linear search.
* Sort the array.
* Similarly perform and display the step count test for binary search.
* You may use some of the code in LinearSearch.cpp and BinarySearch.cpp under our course Resources as a resource but be on the alert for any errors in these programs.
* When you display the arrays to demonstrate your work, display each value on its own line and show the index as a label for the element’s value. For example, if the first element has a value of 39:

[0]: 39

Submit:

1. Zipped project folder for investigating the step counts of each search. The project folder should be submitted in its own zipped project folder. Use only .zip folders (not .rar folder).
2. An Excel spreadsheet showing how the growth of the data set (n) affects the number of steps that each algorithm takes. The data should be shown in table and chart displays. See Unit 2 Part 3 of CSC 231 Units.

Grading

|  |  |
| --- | --- |
| Linear Search  Code is documented and uses meaningful identifiers  Code should be well-structured and easy to read  Search runs correctly  Step counts reasonably correct  Binary Search  Code is documented and uses meaningful identifiers  Code should be well-structured and easy to read  Search runs correctly  Step counts reasonably correct  Excel spreadsheet with chart(s) showing how the growth of the data set affects the number of steps that the algorithm takes. Data displayed in table and chart formats. | 10 points  10 points  10 points  10 points  10 points  10 points  10 points  10 points  20 points  Total 100 |