Surprise!

CS-2050

April 26, 2024

1 Requirements

In this lab, you are tasked with profiling a number of different algorithms to determine the performance difference between them. Your starter code for this lab includes a partially completed `main.c` file, as well as two separate libraries. The `sorting` library includes a number of sorting algorithms, all of which are written to support generic type arguments. Likewise, the `searching` library includes functions for searching generic data arrays. You can use the following command to compile the starter code:

```
gcc -Wall -Werror -Wpedantic -pg main.c sorting.c searching.c
```

1.1 Sorting

```
#include "sorting.h"
```

Info: Edit the main function to set the array to have a size of 100,000 elements. It's *highly* recommended to comment out the calls to `print_int_array`, since printing out 200,000 numbers makes things very difficult to read!

You must compare the performance of a **minimum of two** different sorting algorithms on this array. One of the algorithms in the comparison *must* have an average-case complexity of $O(n^2)$ (IE: bubble sort or selection sort), and one of the other algorithms *must* have an average-case complexity of $O(n\ log(n))$ (IE: quick sort or merge sort).

Copy your results from each profiling into a file called `sort_profile.txt` and save it in the same folder as your starter code.

1.2 Searching

```
#include "searching.h"
```

Info: After completing the above tasks, now you must compare searching the same sorted array with both a linear search and with binary search. You must search in the array at four different points:

1. `(int)(size * 0.33)`
2. `(int)(size * 0.49)`
3. `(int)(size / 2)`
4. `(int)(size * 0.79)`

There is an example in the starter code of how to go about doing this. Copy your results from each profiling into a file called `search_profile.txt` and save it in the same folder as your starter code.

Grading: 10 points This lab is graded on completion! Please zip up your result files along with your starter code, and submit it to the assignment on Canvas. Your finished code will not be graded, but please still include it with your submission in case we need to reference it while looking over your results.