

CS 202 Homework 1

Bryan Beus

February 6, 2020

Source Code Link: <https://github.com/siddhartha-crypto/cs202/tree/master/hw1>

1 Design

1.1 Main

I'm interested to see the data results from this project. The instructions provided are very clear and specific. I'll use a normal distribution in a few places, for fun. For the arbitrary functions to test from the stl, I'll use `sort()` and `reverse()`.

1.2 Time It Two

I am enjoying the opportunities to learn more about processing words and other textual data. As you know, these projects interest me.

I'll use some books from a Norse mythology research that I've been meaning to read more often.

For the code structure, I intend to follow the same pattern as the previous assignment.

2 Post Mortem

2.1 Main

The <chrono> library was quite difficult. I spent many hours yesterday and today trying to get the variables to match.

Finding a way to store the member variables of the class was the most difficult aspect.

Finding a way to calculate a duration between my chosen member variable types was the second most difficult.

Everything else was of a normal difficulty level.

2.2 Time It Two

Some of the time in this project went into learning how to properly use a list and in how to use a deque, both of which were relatively easy.

Outside of that challenge, the rest of the project was relatively straightforward.

3 Answers to Questions

- (Answer for questions one through three, as each are the same in my data) I take the word "should" to mean, in respect to the machine that I am using. In this case, with each step, the results increase by a logarithmic factor. I do not know how to find the specific logarithmic equation, but the logarithmic aspect is very apparent in my data.
- A class is a group of related data and code. The code has responsibilities to do things with that data. A struct is a group of related data, only.
- Private is viewable by outside members and functions, but is not changeable by outside members. Protected is neither viewable nor changeable for outside members and functions; only member functions of the class that declared the

protected member can use it. Public is available to all to utilize and update.

- A "method" is a very general word that we use extensively at my work. Method can describe almost anything in terms of functionality. A member function and a member data are both tied to a class. The function manages the data.
- No, a member function that is declared const is not the same as a member function that is not declare const.
 - byte: 8 bits
 - char: 1 byte
 - unsigned char: 8-bit unsigned integer
 - short: 2 bytes
 - int: 4 bytes
 - long int: 32 bits
 - long long int: 64 bits
- From my research, endianness is the order of writing data from left to right or right to left. This would matter if the hardware or software requires the developer to manage endianness manually. For the hex value, the calculation I receive from a calculator that claims to perform this task for the x86 architecture is "\x0B\xAA\xDF\x00", but I am not sure if I am understanding the question properly. Network protocols require big-endian values.

4 Sample Output

4.1 Main

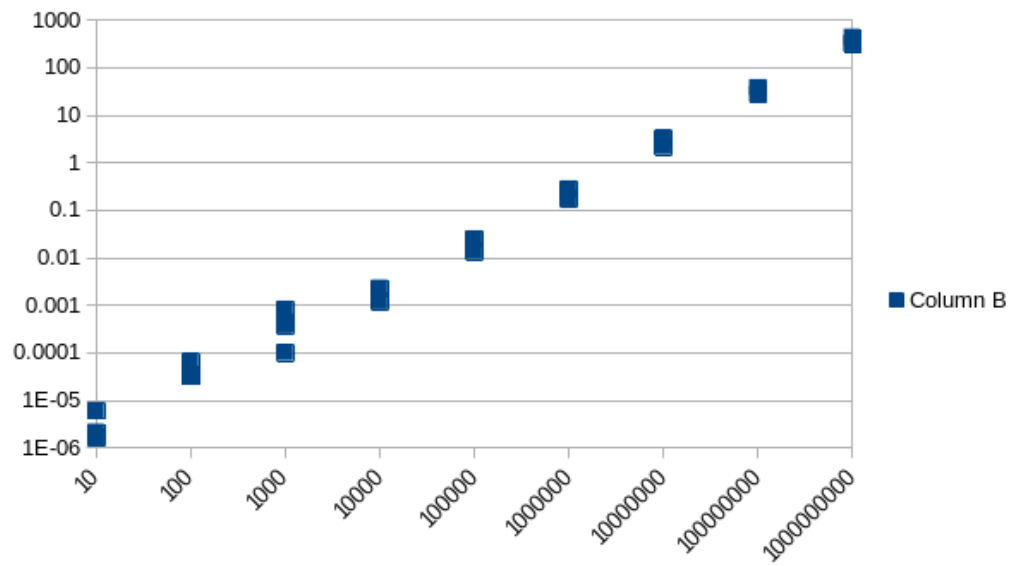


Figure 1: X and Y axis are logarithmic

4.2 Time It Two

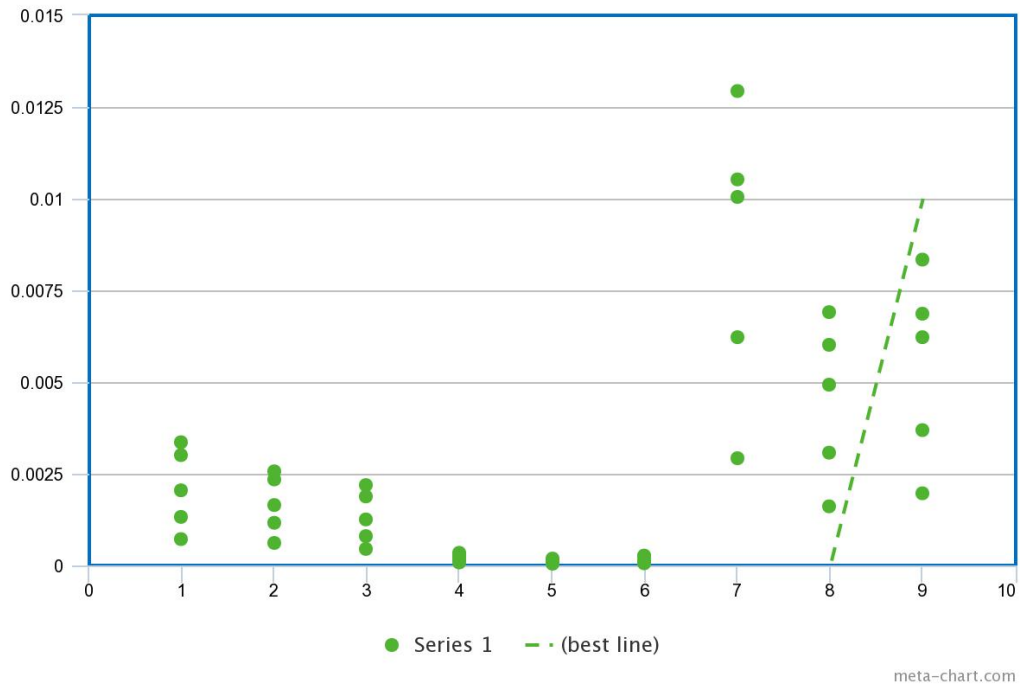


Figure 2: From Left to Right: List: Load, List: Find, List: Sort, Vector: Load, Vector: Find, Vector: Sort, Deque: Load, Deque: Find, Deque: Sort

5 My Programs

5.1 Main

```
1  /*
2  * main.cpp
3  * CS 202
4  * January 19, 2020
5  * Bryan Beus
6  * Main file for temperature main project in hw0
7  */
8
9  #include <iostream>
10 #include <iomanip>
11 #include <string>
12 #include <cstdlib>
13 #include <random>
14 #include <cmath>
15 #include <stdlib.h>
16 #include <chrono>
17 #include <vector>
18 #include <algorithm>
19 #include <time.h>
20
21 #include "Miscellaneous.hpp"
22 #include "TimeItFunctions.hpp"
23 #include "StopWatch.hpp"
24
25 using std::cout;
26 using std::cin;
27 using std::endl;
28 using std::vector;
29 using std::round;
30
31 int main()
32 {
33     // Declare pseudo-random device for creating seeds
34     std::random_device r;
35
36     // Create a seed sequence to feed to the generator
37     std::seed_seq seedObj{r(), r(), r(), r(), r(), r(), r(), r()};
38
39     // Declare random-number generator and provide with seedObj
40     ↪ sequence
41     std::mt19937 e1(seedObj);
42
43     // Declare number of times to multiply list size
44     int num_lists = 9;
45
46     // Create list of results
47     vector< vector<StopWatch>> results;
```

```

47
48 // Create random number that will serve as the search marker
49 vector<int> searchObj = calcObj(1, e1);
50
51 // Perform each round
52 for(int i = 1; i < num_lists + 1; i++) {
53     // Create an object of random numbers
54     vector<int> objList = calcObj(pow(10, i), e1);
55
56     // Create a Stopwatch vector to hold current results
57     vector<StopWatch> current_result = calcTime(objList,
58         ↪ searchObj);
59
60     // Add current result to total list of results
61     results.push_back(current_result);
62 }
63
64 clearConsole();
65
66 // Create marker to print and indicate the size of the object
67 ↪ list that is associated with printed data
68 double curr_counter = 10.0;
69
70 // Iterator through each of the result vectors
71 for (auto i: results) {
72     // Within each result vector, call printNextFive() to
73     ↪ print out individual data values
74     for (int j = 0; j < 4; j++) {
75         printNextFive(j, i, curr_counter);
76     }
77     cout << endl;
78
79     // Increase marker value to next step in data-size list
80     curr_counter *= 10;
81 }
82 }
83
84 return 0;
85 }

```

5.2 Time It Two

```

1 /*
2  * main.cpp
3  * CS 202
4  * February 5, 2020
5  * Bryan Beus

```

```

6  * Main file for time-it-ii project in hw1
7  */
8
9  #include <iostream>
10 #include <iomanip>
11 #include <string>
12 #include <cstdlib>
13 #include <random>
14 #include <cmath>
15 #include <stdlib.h>
16 #include <chrono>
17 #include <vector>
18 #include <algorithm>
19 #include <fstream>
20 #include <deque>
21 #include <list>
22
23 #include "Miscellaneous.hpp"
24 #include "StopWatch.hpp"
25 #include "TimeItTwo.hpp"
26
27 using std::cout;
28 using std::cin;
29 using std::endl;
30 using std::vector;
31 using std::round;
32 using std::ofstream;
33 using std::ifstream;
34 using std::string;
35
36 int main()
37 {
38     // Create a Random Device
39     std::random_device r;
40     std::seed_seq seedObj{r(), r(), r(), r(), r(), r(), r(), r()};
41     std::mt19937 e1(seedObj);
42
43     // Initiate Project Variables
44
45     // Create list of results
46     vector< vector<StopWatch>> results;
47
48     // Names of books
49     vector<string> book_names;
50
51     book_names.push_back("legends-of-the-middle-ages.txt");
52     book_names.push_back("myths-of-greece-and-rome.txt");
53     book_names.push_back("myths-of-the-norse-men.txt");
54     book_names.push_back("stories-of-the-wagner-opera.txt");
55     book_names.push_back("story-of-the-greeks.txt");
56
57     for(unsigned long i = 0; i < book_names.size(); i++) {
58

```



```

59         // Calculate the next result
60         vector<StopWatch> current_result =
            ↳ calcTime(book_names[i], e1);
61
62         // Add data to series
63         results.push_back(current_result);
64     }
65
66     clearConsole();
67
68     // Print the results to the console using the printNextSet()
        ↳ function
69     for (unsigned long i = 0; i < results.size(); i++) {
70         cout << book_names[i] << endl;
71         printNextSet(results[i], i);
72         cout << endl; }
73
74     return 0;
75 }

```

6 Commit History

```

1 a576704 (HEAD -> master, origin/master, origin/HEAD) recompile
    ↳ prog
2 3c6feb2 comment main.cpp file
3 ace3a10 Comment function and header files in time-it-ii in hw1
4 121607b comment files in main in hw1
5 90203cc separate files to header and source files in main in hw1
6 af81f47 time-it-ii debugged in hw1
7 1825ec8 add in vector and deque tests in time-it-ii proj in hw1
8 7ee1aa1 complete scatter plots for main hw1 project, initiate
    ↳ proj 2
9 0cf9f0d update output format for csv
10 d29f14e finish time-it hw1
11 45c5ea9 activate all search methods in time-it hw1
12 ea618b8 time works in time-it hw1
13 01d7d93 test different formats for chrono vs time_t in hw1 main
14 91d831e develop hw1

```
