Question 7 Whitepaper

Umar Kagzi

3/25/21

Question 7: C++ has several containers (https://www.cplusplus.com/reference/stl/). Pick three of them and compare similar operations by timing them on a large scale. **Explain a hypothesis, your results, and your explanation. –** I did this format for the whitepaper.

In this question, I decided to use the array, vector, and list containers in the C++ STL Library to test the time it took for each container to fill itself with random numbers, sort itself using std::sort, and finally printing themselves out with a print function.

My hypothesis was that a vector container with the aforementioned program would take the shortest time to complete since they are quite fast when inserting data into itself. After the vector, the fastest container would obviously be an array and then the final container, the list.

My method to test each container was simple. First, I timed and instantiated an array container with size of 10,000 and filled it with random numbers from 0-100. I then sorted the array using std::sort (uses quicksort, complexity N log(N)) and finally printed the array using a print function and calling the timer. I believed this process would take the containers a long time(seconds) and would give me adequate data, so I continued the same process with the other containers (vector, list).

The timings for the function are listed below, and I was surprised to see that the list had the fastest average time. Throughout the program, it had the most consistent timings, but I still feel that the vector should be faster overall since I used the push_back() function which allocates memory beforehand so the program doesn’t need to earlier. I believe if I did a few more test runs, I would see more accurate numbers. It is possible that the first test run might have some issues (which could be possible since the vector is much faster in the following tests). All in all, this was a great problem to examine the timings of different containers in C++.

In order to have average results**, I tested the function 3x to make sure they were accurate. The average results in the first three test runs were:**

Container Name	**Average Time (in Microseconds)**
Array	47,279
Vector	36,611
List	16,764

First Run (Microseconds)

List: 19331

vector: 23364

array: 39257

Second Run (Microseconds)

List: 14274

vector: 5758

array: 35853

Third Run (Microseconds)

List: 16687

Vector: 7489

Array: 66727