Joe Parisia  
CSC382 – Data Structures and Algorithms

Algorithm Analysis

\*\* I am analyzing the “get\_value()” function. The “die” function is just to show that there wasn’t much added in terms of Big O.

******

***Determine and report the Big O of your algorithm. You may want to report the Big O for the best case, worst case, and average case.***

Originally, I thought it was going to be a nice simple O(n) but when I looked up what the “sort” function does in C++, I found that my answer changed to O(n log n). This is because of the sort function’s algorithm and since I am not further nesting any for loops I believe it stays this way.

***What computational resources does your algorithm/code use?***

RAM for variable storage and CPU for processing of the algorithm.

***What computational resources impact the performance of your algorithm?***

CPU would have the most impact since the variable storage is such a low number.

***In what cases, situations, or scenarios would your algorithms performance be enhanced or diminished?***

As is, my algorithm wouldn’t have any substantial enhancements since the array being processed is so small. However, if you were to use my algorithm on much larger arrays of data then you will it start to slow down.

***Show the lines or sections of code that would be affected and explain why.***

These two lines would be affected the most by a change in the size of array.

* int ability\_array[4];
* std::sort(ability\_array, ability\_array + 4);

This is because the more data you stuff into the array the longer it will take the “sort” algorithm to process it all. The rest of the code is just simple addition of numbers which wouldn’t add anything substantial.

References:

<https://en.cppreference.com/w/cpp/algorithm/sort>