**CSE 212 – Programming with Data Structures**

**W02 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Ethen Campbell |
| **Date:** | January 20, 2024 |
| **Teacher:** | Brother Alvey |

*It is a violation of BYU-Idaho Honor Code to post or share this document with others or to post it online. Storage into a personal and private repository (e.g. private GitHub repository, unshared Google Drive folder) is acceptable.*

**Question 1: From Part 1, what is the big O notation for the sort\_list function?**

O(n^2)

**Question 2: From Part 1, what is the big O notation for the standard\_deviation\_1 function?**

O(2n) or O(n)

**Question 3: From Part 1, what is the big O notation for the standard\_deviation\_2 function?**

O(n^2)

**Question 4: From Part 1, what is the big O notation for the standard\_deviation\_3 function?**

O(n)

**Question 5: From Part 1, put the following big O notations in order from best performance to worst performance: O(n^2), O(1), O(2^n), O(n log n), O(log n), O(n).**

O(1), O(log n), O(n), O(n log n), O(n^2), O(2^n)

**Question 6: From Part 2, what is the performance (using big O notation) for the search\_sorted\_1 function?**

O(n)

**Question 7: From Part 2, what is the performance (using big O notation) for the search\_sorted\_2 function?**

O(log n)

**Question 8: From Part 2, which function (search\_sorted\_1 or search\_sorted\_2) has the better performance?**

Search\_sorted\_2

**Question 9: From Part 2, for both functions (search\_sorted\_1 and search\_sorted\_2), explain in detail how you determined the big O notation by just looking at the code without the benefit of observing actual execution results?**

For search\_sorted\_1, it has a single for loop. I find that as an indication to it being O(n). For search\_sorted\_2, it has two if statements that help it sort faster. It splits it easier and makes it more efficient. I realized that this was O(log n) because of the two if statements. With these guesses, I guessed that the second one would do better and then when it did, I knew I was right or at least close.

**Question 10: From Part 2, it is possible in the best case for each of these functions (search\_sorted\_1 and search\_sorted\_2) to complete in O(1) time even if the size of the list was very large. What input scenarios would give this result for both functions?**

Yes, it is possible! In search\_sorted\_1, it runs through the list one at a time. It would take O(1) if the wanted item was the first in the list. Since it’s O(n), it makes sense that when n=1, it would become O(1).

In search\_sorted\_2, it runs through the list starting from the middle and comparing. It would take O(1) if the wanted item was in the middle of the list because that’s where it starts.