**CSE 212 – Programming with Data Structures**

**W02 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Zach Newby |
| **Date:** | September 23, 2023 |
| **Teacher:** | Zachariah 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) = 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(n), O(n log n), O(2^n), O(n^2)

**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?**

Well, I determined search\_sort\_1 had a notation of O(n) because it loops through the data once, and the worst result is going through the entire list without finding the target.

As for search\_sort\_2, I remember hearing in the reading material for this week, that the method used was faster than looping through, (although I don’t think it is actually faster for smaller collections, larger I can easily see.) so I predicted it would be faster based on that information.

**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. For search\_sorted\_1 the first item would have to be the target, and for search\_sorted\_2 the length would have to be zero or the value at the middle index would have to be the target.