**CSE 212 – Programming with Data Structures**

**W02 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Jonathan Starks |
| **Date:** | 5/2/2024 |
| **Teacher:** | Brother Comeau |

*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 SortArray function?**

O(n­2)

**Question 2: From Part 1, what is the big O notation for the StandardDeviation1 function?**

O(n)

**Question 3: From Part 1, what is the big O notation for the StandardDeviation2 function?**

O(2n)

**Question 4: From Part 1, what is the big O notation for the StandardDeviation3 function?**

O(n)

**Question 5: Put the following big O notations in order from best performance to worst performance: 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 SearchSorted1 function?**

O(n) or O(1) if it’s the first item in the list.

**Question 7: From Part 2, what is the performance (using big O notation) for the SearchSorted2 function?**

O(log n)

**Question 8: From Part 2, which function (SearchSorted1 or SearchSorted2) has the better performance?**

It depends on where the desired index is, if its near the beginning, then it’s the searchsorted1, but if its anywhere else its searchsorted2

**Question 9: From Part 2, for both functions (SearchSorted1 and SearchSorted2), explain in detail how you determined the big O notation by just looking at the code without the benefit of observing actual execution results?**

I tried to visualize what the code did. The searchsorted1 code just goes through the list in order, whereas the searchsorted2 code cuts the list in half each time it advances.

**Question 10: From Part 2, it is possible in the best case for each of these functions (SearchSorted1 and SearchSorted2) 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?**

**For searchsort1 if the item is the first one in the list then it will have an efficiency or O(1).**

**For searchsort2 if the item is right in the middle of the list then it will have an efficiency of O(1).**