**CSE 212 – Programming with Data Structures**

**W02 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Jackson Chan |
| **Date:** | 27th April, 2022 |
| **Teacher:** | Brother Kay |

*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(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 has better performance

**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?**

I would look at the loops first to determine if there is an (n) in the algorithm. If there is a nested loop then it would be another (n), which will be (n\*n = O(n^2)). if there are two loops, (n + n = O(n)).

If there is no loop, I look for recursion, if there is recursion, it is most likely (log n). If there is a loop and recursion, then it would be (n log n).

**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?**

Search\_sorted\_1: When the target number is the first one in the list. (So, you find it the first time you run the function.)

Search\_sorted\_2: When the target number is this middle number in the list. (So, you find it the first time when you slice the length of the list in half.)