**CSE 212 – Programming with Data Structures**

**W02 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Trent Black |
| **Date:** | 04-26-2020 |
| **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(2^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(log n),O(1), 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?**

Sort\_time\_2 has the faster 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?**

In search\_sroted\_1\_ I was able to tell that the big O(n) because it uses a for loop which is an n. I know that 2 for loops makes n^2 and since there was only one for loop it was just an n

In search\_sorted\_2 I was able to tell that it was O(log n)because it searched in the middle of the list and looked at both ends of the list and splits the data to find the target

**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?**

**To set the test\_data manually to a certain for (number x for x in range(target)) instead of using n**

target = 2 #or could be 200 or 2000

test\_data = [x for x in range(target)]

**this way they are getting o(1) time every time.**