**CSE 212 – Programming with Data Structures**

**W05 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Jackson Chan |
| **Date:** | 18th May 18, 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, how did you answer the interview question for the Set Operations problem (should be no more than 30 seconds if spoken aloud)?**

For intersection, I will create a new set, add intersected items in the new set by running a for loop to check. The for loop will be looping through set1 or set2, each iteration will check if item is in set2 or set1, add the item to the new set if returns true. This will be O(n).

For union, I will also create a new set, running a for loop to check if an item in set1 is in set2, if yes, add it to the new set.

**Question 2: From Part 2, how did you answer the interview question for the Find Pairs problem (should be no more than 30 seconds if spoken aloud)?**

I will create a function to reverse a string. Then create 2 new sets, one will use the reverse function to reverse all the items from `words` and store them. The second will be empty.

Then create a for loop for `words`, adding 3 conditions, if the item meets these conditions, it and it’s reversed version will be displayed, then adding the items that have been iterated to the empty set.

The three conditions are:

1. An item in `words` is in the reversed set.
2. An item is not equal to the reversed version of itself.
3. The reversed version of the item is not in the second set.

Remember: You need to submit the following code files in addition to this document:

* 05-prove\_set\_operations.py
* 05-prove\_find\_pairs.py