**CSE 212 – Programming with Data Structures**

**W05 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Luis Roberto Cardenas Melgar |
| **Date:** | 5/21/22 |
| **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, how did you answer the interview question for the Set Operations problem (should be no more than 30 seconds if spoken aloud)?**

To find the intersection of two set I would write a function that would loop through the first set and check if its values are in the second set. If some of the values of the first set are in the second set, it would add those values to a third set. Sets allows to check faster if an element in a set is in another set because it does not have repeated values. The third set would be the compilation of the values repeated in both sets. To find the union of two sets I would write a function that would add the values of the first set and the values of the second set into a third set. Due to a set does not allow duplicate, any duplicated value wouldn’t be added to the third 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)?**

To find the symmetric pairs of two letters words in a list in O(n) time using a set, I could loop through each element in the list and compare the letters in the word. If they are different, I can flip the letters of the words to create its symmetric pair and add the symmetric pair into an empty set. Then inside the same loop, I can check if the original word is in the set. If it’s in the set, it will print the original word and its symmetric pair. If the symmetric pair of a word is in the same list, it should have been added to set, and by checking if the original word is in the set, we would be finding the resulting word of flipping the letters of its symmetric pair. As things are not repeated in a set, it’s faster to check if an element is in the 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