CS-375 Assignment 1 – Alex Eskenazi – Fall 2025

**Algorithm Explanations**

**Part B.1: Sorting A, B, C characters**

Sorts characters so all A's come first, then all B's, then all C's.

Algorithm:

* Use three pointers: low, mid, high
* Move A's to the left, C's to the right, B's stay in middle

Each character is looked at only once, so we get O(n)

Time Complexity: O(n) - linear time

Space Complexity: O(1) - only uses a few variables which are constant size.

**Part B.2: Finding Key as Difference**

Find numbers in the array where one number equals the difference of two other numbers.

Algorithm Steps:

* Sort the array using merge sort - O(n log n)
* For every pair of numbers (i, j) - O(n²)
* Calculate their difference - O(1)
* Search for that difference in the sorted array using binary search - O(log n)
* If found, print the result - O(1)

Time Complexity Analysis:

* Sorting: O(n log n)
* Nested loops: O(n²)
* Binary search inside loops: O(log n)

Total: O(n log n) + ( O(n²) \* O(n log n) ) ---> O(n²)O(n log n) ----> O(n² log n)

Space Complexity: O(n) - for the sorted copy of the array