**Leetcode Solving Techniques**

1. Two Sums:

|  |  |
| --- | --- |
| **Brute Force Solution** | **Optimized Solution** |
| **Nested for loops**: When i iterates once, j iterates completely until values are found that when the add up the sum up to the target, the index of i and j is returned | **Using HashMap**:  Create a HashMap, loop through the numbers, in the loop we will minus the iterated value from the target if the result is already in the hash map, we use the get function to return the index of the value and the current iteration that gave us that value, and we return an empty array is nothing is found |

1. Contains Duplicate:

|  |  |
| --- | --- |
| **Brute Force Solution** | **Optimized Solution** |
| First, we sort the Array using Arrays.sort(nums)  The we loop through the sorted values and compare the current value with the last one if they are equal return true | We create an Hashset, we loop through the array, if we cannot add a current value to the Hashset it returns true for duplicate, else it returns false |

1. Valid Anagram

|  |  |
| --- | --- |
| **Brute Force Solution** | **Optimized Solution** |
| We convert the 2 strings into lowercase char array, we also create and array to keep track  First, we loop through the first string and for every iteration we increment the iteration – ‘a’ to the tracker array to 1  We do same for the second char array and this time decrement is from 1 to 0(tracker[tArray[i] – ‘a’] --)  Now we loop through the tracker array if the values are not 0, we know it was not a valid anagram | Basically, it the same as the brute force solution but we avoid unnecessary decelerations and use enhanced for loops |