**Part B- 2(b)**

**Algorithm used- It’s implementation:**

* I have used ifstream to read the data from input.txt file and stored it into an array
* I have calculated the size of the above array and passed the array and size through function matchdiff.
* This array is stored in an unordered\_set.
* Now I have calculated the difference of each element by iterating through double for loop and checked if the difference is present in unordered set.
* If the difference is present in the unordered set, we will get the desired set of numbers and we will store the difference and those two numbers which are giving the difference in an arraylist in sequential order.
* If the arraylist is empty it means no match is found and we will print the respective ouput.
* Else we will run the loop and print the sets in order of : difference larger\_number smaller\_number.

**worst-case time complexity**

* Below is the time complexity of the code:

1.Suppose here the length=n

for ( int h = 0 ; h < length; h++) - - - -> This has time complexity of O(n)

{

hashSet.insert(array[h]);

}

2. for (int i = 0; i < length; i++) - - - -> “This has the time complexity of O(n) and the

{ loop will operate for (n+1) time”

for (int j = i + 1; j < length;j++) - ->”Time complexity: O(n^2) and

{ the loop will operate for n(n+1) times.

diff = std::abs(array[j] - array[i]);

And all other function inside the loop will give either O(n) or O(n^2) time complexity depending on the time complexity of the loops.

**So the worst case time complexity is : O(n^2)**