COMP3121 21T2 Assignment 1 Q1

Written by Zheng Luo (z5206267)

Part A

In order to satisfy the equation:

$$m^2 + s = k + p^2$$

We need to create an array B, sort the given array A and place into array B by using merge sort algorithm with time complexity of $O(nlog(n))$, therefore array B will be used for integer s and k, since they are integers without square term. Also, we need to design an array for squared integers m^2 and p^2, hence we square each term in array B and place the result into array C with time complexity of $O(n)$.

Perform a double-array addition algorithm between array B and C, and record the result in array D.

```
for (int i = 0; i < len(B), i++) {
        for (int j = 0; j < len(C), j++) {
                arrayD[i+j] = arrayB[i] + arrayC[j];
        }
}
```

However the question specified integers must be distinct, hence $i! = j$ in the pseudo code above. This results to $(n)(n-1)$, which is $O(n^2)$ time complexity.

```
int k = 0;
for (int i = 0; i < len(B), i++) {
        for (int j = 0; j < len(C), j++) {
                if (i != j) {
                        arrayD[k] = arrayB[i] + arrayC[j];
                        k++;
                }
        }
}
```

Eventually, we can sort the array D by using merge sort with the time complexity of:

$$n^2 log(n^2) = 2n^2 log(n) = O(n^2 log(n))$$

Hence in sorted array D, there are four distinct integers m, s, k, p can satisfied the equation if there is any duplicated value existed.

Part B

The problem can be solved by same approach for all steps but last one:

1. Sort array A into array B by merge sort with time $O(nlog(n))$.
2. Square array B into array C with time $O(n)$.
3. Perform double-array addition between array B and C with time $O(n^2)$.

However, we store the result into array D by using hash table with time complexity of $O(1)$ for each search and insertion, so the distinct integers existed if there is a hashing clash, which means there is a same value exist when inserting the current value. Hence, this algorithm will run in time $O(n^2)$, since there is no terms larger than $O(n^2)$.