HOMEWORK 1 – Q1

MINGLANG XIE

z5228006

1. You are given an array *A* of *n* distinct integers.
2. You have to determine if there exists a number (not necessarily in *A*) which can be written as a sum of squares of two distinct numbers from *A* in two different ways (note: and counts as a single way) and which runs in time in the **worst case** performance. Note that the brute force algorithm would examine all quadruples of elements in *A* and there are such quadruples. (10 points)
3. Solve the same problem but with an algorithm which runs in the **expected time** of . (10 points)

**Solution:**

1. First, we sort the array – we can do this in in the worst case, for example, using Merge Sort.

Second, we need to create a new empty array *T* to store the result of the calculation.

Third, for each element we can calculate its sum of squares with all the other element , and the worst case for this part is . For every calculation we do, we can use binary search to check that the array *T* has the same solution in time. There is no special case because all elements in array *A* are different integers.

Hence, we take at most time per calculation, in the worst case we must do time calculations, which is giving an algorithm.

Time

Merge Sort (A)

for i in A:

for j in :

calculation =

binary search(calculation)

T.append (calculation)

1. We take similar approach as in (a), except using a hash table to check if the same solution exists in the array *T*, each insertion and lookup takes expected time.

* First, we can store all calculation into the hash table by simply iterate it.
* Second, at index , we assume all calculation is already stored in the hash table. Then we check if is in that hash table in , then insert into the hash table, also in .