COMP3121 ASSIGNMENT1 QUESTION1

1. You are given an array A of n distinct integers.

   (a) 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: m^2+n^2 and n^2+m^2 counts as a single way) and which runs in time $n^2 \log n$ in the **worst case** performance. Note that the brute force algorithm would examine all quadruples of elements in A and there are $\binom{n}{4} = O(n^4)$ such quadruples. (10 points)

   (b) Solve the same problem but with an algorithm which runs in the **expected time** of $O(n^2)$. (10 points)

Answer:

(A) Frist, we create and initialization array B and size of $\frac{n(n-1)}{2}$which to store the all result for pair. O($\frac{n(n-1)}{2}$) = O(n^2)

Then each element in A, we need go through all $\frac{n(n-1)}{2}$pairs (A[K], A[M]), K < M, of **distinct** integers in A. And for every pair we need compute ($O(1) \cdot n^2$) and store the result(A[K]^2+A[M]^2) in array B($O(1) \cdot n^2$). It will take O(n^2) + O(n^2) + O(n^2)= O(n^2) time.

Then we sort array B. because the size of b is $\frac{1}{2}$(n^2-n) < n^2 we use n^2 for further prove for convenient. We use merge sort can sort in O($n \log n$) in worst case. because the size of n should be n^2 in our case. It become $O(n^2 \log n^2) = O(2n^2 \log n) = O(n^2 \log n)$

Then for each element in B, we use binary search that whether same element in B. If find it should return true if not return false. In the worst case it will go through every element in B and do binary search every time. It will take $O(n^2 \log n)$.
So worst case should take $O(n^2 \log n) + O(n^2 \log n) + O(n^2) = O(n^2 \log n)$

(B) Frist, we create and initialization array B and size of $\frac{n(n-1)}{2}$which to store the all result for pair. O($\frac{n(n-1)}{2}$) = O(n^2)

Then each element in A, we need go through all $\frac{n(n-1)}{2}$pairs (A[K], A[M]), K < M, of distinct integers in A. And for every pair we need compute ($O(1) \cdot n^2$) and store the result(A[K]^2+A[M]^2) in array B($O(1) \cdot n^2$). It will take O(n^2) + O(n^2) + O(n^2)= O(n^2) time.

Then we do not sort and use hash table to check if a same element exist in the array: each insertion and lookup takes O(1) expected time.
Then we hash all elements of B and then go though elements of B again, this time for each element in B checking in O (1) time at whether appear multiple in the corresponding slot of the hash table. If appear multiple it is true, not are false. This will be done in expected time of O(n^2).