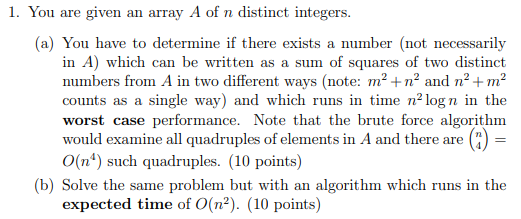
**COMP3121 ASSIGNMENT1 QUESTION1**



**Answer:**

1. Frist, we create and initialization array B and size of which to store the all result for pair. O() = O()

Then each element in A, we need go through all pairs (A[K], A[M]), K < M, of **distinct** integers in A. And for every pair we need compute () and store the result(A[K]2+A[M]2) in array B(). It will take O(n2) + O(n2) + O()= O(n2) time.

Then we sort array B. because the size of b is (n2-n) < n2 we use n2 for further prove for convenient. We use merge sort can sort in O() in worst case. because the size of n should be n2 in our case. It become

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 .

So worst case should take =

1. Frist, we create and initialization array B and size of which to store the all result for pair. O() = O()

Then each element in A, we need go through all pairs (A[K], A[M]), K < M, of distinct integers in A. And for every pair we need compute () and store the result(A[K]2+A[M]2) in array B(). It will take O(n2) + O(n2) + O()= O(n2) 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(n2).