Q1

Written by Xiao Hu
Z5223731

Answer:

a. We go through all of $\binom{n}{2} = \frac{n(n-1)}{2}$ pairs (A[k], A[m]; k < m) in array A and compute the sums of $A[k]^2$ and $A[m]^2$ then put them into a new array B of size $\frac{n(n-1)}{2}$. we sort the array by using merge sort to check if there are two numbers in array B are equal to see the number which can be written as a sum of squares of two distinct numbers exists or not. The only special case is the two same numbers are adjacent during the last division in merge sort and it takes O ($\log n$) time. However, this algorithm takes O ($n^2 \log n$) in the worst case due to the size of new array B is $\frac{n(n-1)}{2}$.

b. We take a similar approach as in (a), but using a hash table to check if there are two numbers are equal.

   We hash all elements of B and check if the corresponding slot that you try to hash in is already filled or not. It will take at most O (n^2) time.