Homework Assignment #3
Thu, Feb 13, 2020 5:30 PM
Written by *Jiahong Zhai* 1005877561
Read by *Jiabao Shen 1004297548*

Question 3

a.

Main idea: Create an empty hash table. Add the square of all element in X into hash table as keys. Loop over Y, for each y in Y calculate the value of $d^2 - y^2$. Check if the result is a key in the hash table. If yes, return "Yes". Otherwise, check the next element in Y. if no such Y has been found, then return "No".

```
Check(X, Y):
HT = new empty HashTable
for x in X:
    x² insert to HT as key
for y in Y:
    search key d² – y² in HT
    if found:
        return "Yes"
return "No"
```

b.

We assume that for all x in X, x^2 is Simple Uniform Hashing. Which is to say that the probability of any x^2 hashed to any slot in the Hash Table is the same. We also assume that each the hash table use chaining to solve collision, and for each newly added element, its added to the head of the slot. Which is to say that each insert operation takes constant time. Additionally, we assume that each Mathematical calculation takes constant time. The first loop takes $O(n)$ time, since it need to loop through a list with n elements. The second loop takes $O(n)$ time. Because, the expected time complexity for each search operation is $O(1)$, and the expected repeat time for the second loop is $O(n)$. Other operations in the program take $O(1)$ time. So, in total. Its expected time complexity is $O(n)$.

c.

In the worst case, for all x in X, x^2 is hashed to the same slot in hash table, and there is no such y in Y^2 that exists an x in X, $x^2 + y^2 = d^2$. All assumptions in b stays the same. The only different part from b is the second loop. Each search operation will take $O(n)$ times, because it has to loop through the only non-empty slot in hash table, which contains n elements. And the second loop repeats n times. So, the time complexity of the second loop is $O(n^2)$. the upper bound time complexity of the entire program is $O(n^2 + n + 1) = O(n^2)$. In terms of lower bound, take an arbitrary X with n ($n>100$) distinct positive integer, insert all x^2 as key to one slot in the hash table, the program will take more than $0.1n^2$ time. So, the lower bound is $\Omega(n^2)$. So, the worst-case time complexity is $\Theta(n^2)$.