question1

June 19, 2021

[1]:
```
from sage.misc.converting_dict import KeyConvertingDict
```

0.1 Utility Function

The findMaxCount function takes a list and checks if there are any two elements such that their difference is a square and returns the pairs.

[2]:
```
def findMaxCount(array):
    count = KeyConvertingDict(int)
    for i in array:
        count[i]=0
    for i in range(len(array)):
        for j in range(i+1,len(array)):
            a = array[i]
            b = array[j]
            if a!=b and ZZ(abs(a-b)).is_square():
                count[a]+=1
                count[b]+=1
    counted = []
    for key in count.keys():
        if count[key]>0:
            counted.append((count[key], -1*key))
    return sorted(counted)[::-1]
```

[3]:
```
# Creating the subset A
A = list(range(1, 225+1, 1))
```

[4]:
```
# Finding the squares less than 225
squares = [i*i for i in range(1, int(sqrt(255))+1, 1) ]
```

0.2 Logic

Iterate through the array, for every element i, remove $i + k^2$ from the array, where $k \in squares$

Consider a random set $A \equiv \{a_0, a_1, ..., a_n\}$. As we are only worried about the difference of any two elements, we can condiser the set $A' \equiv \{1, a_1 - a_0 + 1, ..., a_n - a_0 + 1\}$. Applying the above

algorithm will therefore yeild the longest subset of A with no two elements whose difference is a perfect square.

[5]:
```
B = A.copy()
for i in B:
    for j in squares:
        if B.count(i+j):
            B.remove(i+j)
C = A.copy()[::-1]
for i in C:
    for j in squares:
        if C.count(i-j):
            C.remove(i-j)
```

[6]:
```
print("The Maximum length of Subset B of A is ", len(B))
```

```
The Maximum length of Subset B of A is  41
```

[7]:
```
print("The Subset B is ")
print(B)
```

```
The Subset B is
[1, 3, 6, 8, 11, 13, 16, 18, 21, 23, 35, 40, 45, 53, 58, 63, 66, 68, 73, 86, 96,
110, 120, 125, 128, 131, 133, 138, 143, 148, 151, 171, 178, 181, 183, 188, 193,
198, 205, 211, 216]
```

[8]:
```
# Testing if B is a valid subset
findMaxCount(B)
```

[8]: []

[ ]:
```

```