Ishaan and Sticks

Ishaan has a craving for sticks. He has N sticks. He observes that some of his sticks are of the same length, and thus he can make squares out of those.
He wants to know how big a square he can make using those sticks as sides. Since the number of sticks is large, he can't do that manually. Can you tell him the maximum area of the biggest square that can be formed?
Also, calculate how many such squares can be made using the sticks.

Example 1:

```
â€‹Input : arr[ ] = {2, 2, 2, 9, 2, 2, 2, 2, 2}
Output : 4 2
Explanation:
2 squares of side 2 are formed.
return maximum area and number of square.
```

â€‹Example 2:

```
Input : arr[ ] = {5, 3, 2, 3, 6, 3, 3}
Output :  9 1
```

Your Task:
This is a function problem. The input is already taken care of by the driver code. You only need to complete the function **square()** that takes an array **(arr)**, sizeOfArray **(n)** and return the array of the maximum area of the largest square that can be formed and the number of squares that can be formed if there is no possible square return **-1**. The driver code takes care of the printing.

Expected Time Complexity: O(N).
Expected Auxiliary Space: O(M), where M is the maximum value of an array.

Constraints :

1 ≤ N ≤ 10^5

1 ≤ arr_i ≤ 10^3

```
import collections
def square (arr, n) :
    #Complete the function
    freq = collections.Counter(arr)
    freq = sorted(freq.items(),key=lambda x: (-x[1],-x[0]))
    maxArea = 0
    number = 0
    for key,count in freq:
        if count>=4:
            if key**2 > maxArea:
                maxArea = key**2
                number = count//4
    if maxArea==0:
        return [-1]
    else:
        return maxArea,number
```