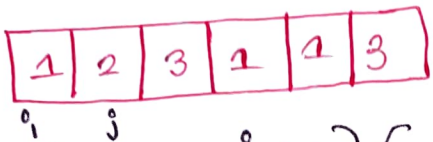
LeetCode 1512. Good Pairs

Problem: Given an array of integers `nums` return the number of good pairs. A pair is called good if (nums[i] == nums[j]) and i < j.

Example,
nums = [1, 2, 3, 1, 1, 3]
O/P : 4 Good pairs

Brute Force




```
for (i = 0; i < n; i++) {
  for (j = i + 1; j < n; j++) {
    if (nums[i] == nums[j]) {
      count++;
    }
  }
}
return count;
}
```

My Solution (Pattern Identification + Math).

- If the array had [1, 1] ∴ Number of pairs = 1.

Similarly, if nums = [1, 1, 1] ∴ Number of pairs = 3 (1+2)
if nums = [1, 1, 1, 1] ∴ Number of pairs = 6 (3+3)
if nums = [1, 1, 1, 1, 1] ∴ Number of pairs = 10 (6+4)

So, by observation, we can see that if we know the count, we can code the above pattern.

N of Pairs for [1, 1] = 1 + 0 = 1
for [1, 1, 1] = 1 + 2 = 3
for [1, 1, 1, 1] = 3 + 3 = 6
for [1, 1, 1, 1, 1] = 6 + 4 = 10
for [1, 1, 1, 1, 1, 1] = 10 + 5 = 15
for [1, 1, 1, 1, 1, 1, 1] = 15 + 6 = 21

res += count.get(num)
count.put(num, count.get(num)+1)

number of occurences

∴ we need to code such a solution, which keeps a track of current number / count of pairs and adds + 1 to that count and finally adds it with the previous result.

Algorithm :

① Iterate through each element. If the element was not present in the map, put it in the map with a default count of 1 (as it is being added in the map for the first ever time.)

② If the element was already there, then perform the following :

i) Add its number of occurences in a "result" variable.

ii) Then, update its value (number of occurences) in the map.

```
res += count.get(num);
count.put(num, count.get(num)+1);
```