

[每日 LeetCode] 532. K-diff Pairs in an Array

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1552139373825>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Description:

Given an array of integers and an integer **k**, you need to find the number of **unique** k-diff pair in the array. Here **k-diff** pair is defined as an integer pair (i, j), where **i** and **j** are both number in the array and their **absolute difference** is **k**.

Example 1:

Input: [3, 1, 4, 1, 5], k = 2

Output: 2

Explanation: There are two 2-diff pairs in the array, (1, 3) and (3, 5).

Although we have two 1s in the input, we should only return the number of **unique** pairs.

Example 2:

Input: [1, 2, 3, 4, 5], k = 1

Output: 4

Explanation: There are four 1-diff pairs in the array, (1, 2), (2, 3), (3, 4) and (4, 5).

Example 3:

Input: [1, 3, 1, 5, 4], k = 0

Output: 1

Explanation: There is one 0-diff pair in the array, (1, 1).

Note:

1. The pairs (i, j) and (j, i) count as the same pair.
2. The length of the array won't exceed 10,000.
3. All the integers in the given input belong to the range: [-1e7, 1e7].

思路：题目意思是给出一组数组，要求查找绝对值差值为k的组合有多少个，可以使用unordered_map映射，直接查找某个值-k是否存在，value存储数组的值出现的次数，因为当k=0的时候，需要判断该值是否有不少于出现两次，除此之外我们就只用判断k值是否存在即可。特别注意unordered_map中count方法，只返回1（存在）或0（不存在）。

C++代码

```
class Solution {
public:
    int findPairs(vector<int>& nums, int k) {
        if (k < 0)
            return 0;
        unordered_map<int, int> umap;
        for (int num : nums)
            umap[num]++;

        int count = 0;
```

```
for (auto u : umap){
    if (k == 0){
        if (u.second >=2 )
            count++;
    }
    else{
        count += umap.count(u.first - k);
    }
}
return count;
};
```

运行时间: 42ms

运行内存: 12.5M