

# LeetCode 第 454 号问题：四数相加 II

本文首发于公众号「图解面试算法」，是 [图解 LeetCode](#) 系列文章之一。

同步博客：<https://www.algomooc.com>

题目来源于 LeetCode 上第 454 号问题：四数相加 II。题目难度为 Medium，目前通过率为 50.8%。

## 题目描述

给定四个包含整数的数组列表 A, B, C, D, 计算有多少个元组 (i, j, k, l), 使得  $A[i] + B[j] + C[k] + D[l] = 0$ 。

为了使问题简单化，所有的 A, B, C, D 具有相同的长度 N，且  $0 \leq N \leq 500$ 。所有整数的范围在 -228 到 228 - 1 之间，最终结果不会超过  $2^{31} - 1$ 。

例如:

输入:

A = [ 1, 2]

B = [-2, -1]

C = [-1, 2]

D = [ 0, 2]

输出:

2

解释:

两个元组如下:

1. (0, 0, 0, 1) ->  $A[0] + B[0] + C[0] + D[1] = 1 + (-2) + (-1) + 2 = 0$

2. (1, 1, 0, 0) ->  $A[1] + B[1] + C[0] + D[0] = 2 + (-1) + (-1) + 0 = 0$

## 题目解析

与 [Two Sum](#) 类似，需要用哈希表来解决问题。

- 把 A 和 B 的两两之和都求出来，在哈希表中建立两数之和与其出现次数之间的映射
- 遍历 C 和 D 中任意两个数之和，只要看哈希表存不存在这两数之和的相反数就行了

## 动画描述

## 代码实现

```
// 454. 4Sum II
// https://leetcode.com/problems/4sum-ii/description/
// 时间复杂度: O(n^2)
// 空间复杂度: O(n^2)
class Solution {
public:
    int fourSumCount(vector<int>& A, vector<int>& B, vector<int>& C, vector<int>& D) {
```

```
unordered_map<int,int> hashtable;
for(int i = 0 ; i < A.size() ; i ++){
    for(int j = 0 ; j < B.size() ; j ++){
        hashtable[A[i]+B[j]] += 1;
    }
}

int res = 0;
for(int i = 0 ; i < C.size() ; i ++){
    for(int j = 0 ; j < D.size() ; j ++){
        if(hashtable.find(-C[i]-D[j]) != hashtable.end()){
            res += hashtable[-C[i]-D[j]];
        }
    }
}

return res;
}
};
```