454. 4Sum II

Description Hints Submissions Discuss Solution

Pick One

Given four lists A, B, C, D of integer values, compute how many tuples `(i, j, k, l)` there are such that `A[i] + B[j] + C[k] + D[l]` is zero.

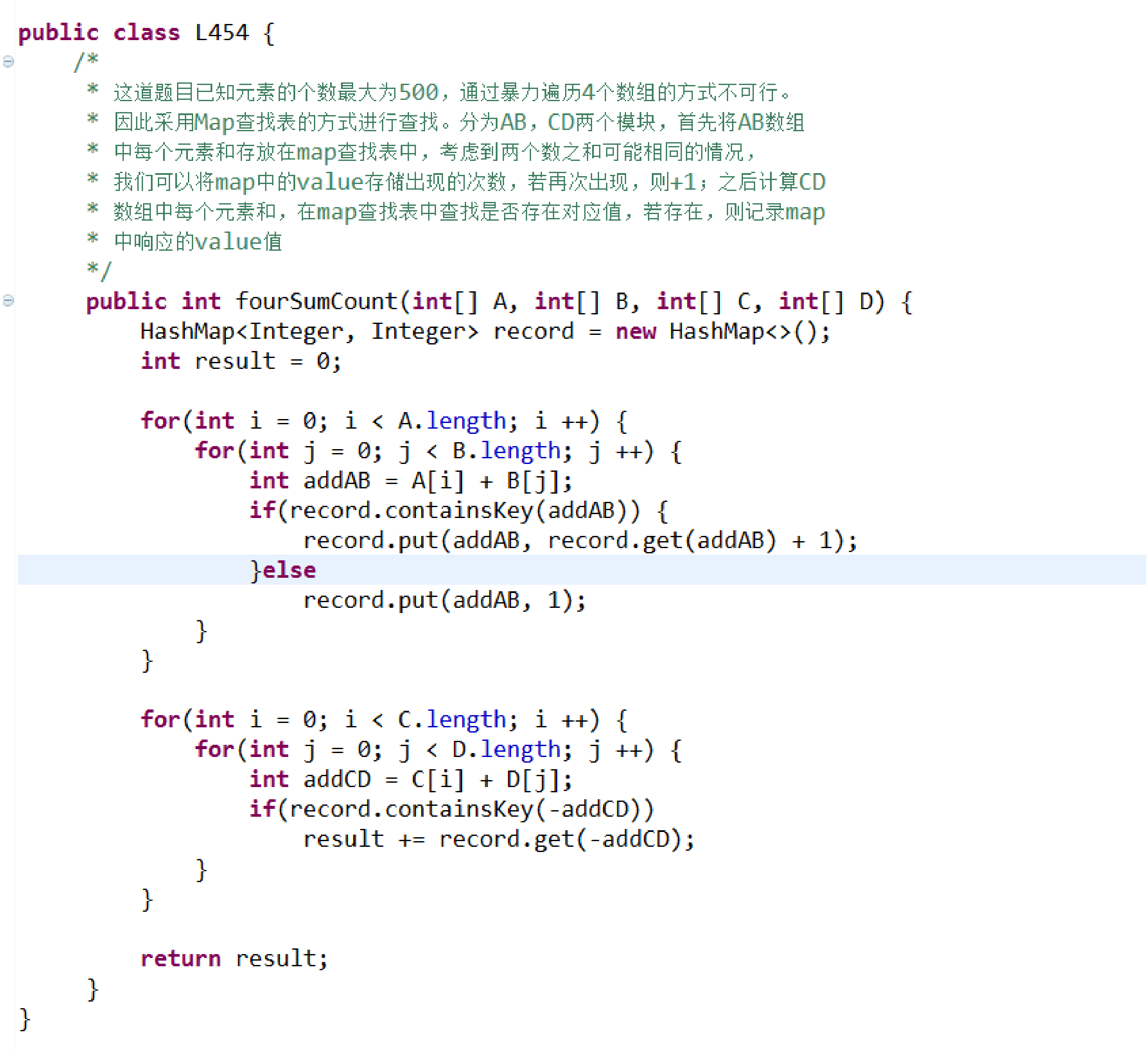
To make problem a bit easier, all A, B, C, D have same length of N where 0 ≤ N ≤ 500. All integers are in the range of -2^{28} to $2^{28} - 1$ and the result is guaranteed to be at most $2^{31} - 1$.

Example:

```
Input:
A = [ 1, 2]
B = [-2,-1]
C = [-1, 2]
D = [ 0, 2]

Output:
2

Explanation:
The two tuples are:
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
```



```
public class L454 {
    /*
     * 这道题目已知元素的个数最大为500，通过暴力遍历4个数组的方式不可行。
     * 因此采用Map查找表的方式进行查找。分为AB，CD两个模块，首先将AB数组
     * 中每个元素和存放在map查找表中，考虑到两个数之和可能相同的情况，
     * 我们可以将map中的value存储出现的次数，若再次出现，则+1；之后计算CD
     * 数组中每个元素和，在map查找表中查找是否存在对应值，若存在，则记录map
     * 中响应的value值
     */
    public int fourSumCount(int[] A, int[] B, int[] C, int[] D) {
        HashMap<Integer, Integer> record = new HashMap<>();
        int result = 0;

        for(int i = 0; i < A.length; i ++) {
            for(int j = 0; j < B.length; j ++) {
                int addAB = A[i] + B[j];
                if(record.containsKey(addAB)) {
                    record.put(addAB, record.get(addAB) + 1);
                }else
                    record.put(addAB, 1);
            }
        }

        for(int i = 0; i < C.length; i ++) {
            for(int j = 0; j < D.length; j ++) {
                int addCD = C[i] + D[j];
                if(record.containsKey(-addCD))
                    result += record.get(-addCD);
            }
        }

        return result;
    }
}
```