# 题目

给定两个整数数组a和b，计算具有最小差绝对值的一对数值（每个数组中取一个值），并返回该对数值的差

示例：

输入：{1, 3, 15, 11, 2}, {23, 127, 235, 19, 8}

输出：3，即数值对(11, 8)

提示：

1 <= a.length, b.length <= 100000

-2147483648 <= a[i], b[i] <= 2147483647

正确结果在区间 [0, 2147483647] 内

# 分析

## 方法一：双指针

这个问题可以通过先将两个数组排序，然后使用双指针来逐步找到具有最小差绝对值的一对数值。具体步骤如下：

1、将数组a和b分别排序。

2、使用两个指针pa和pb分别指向数组a和b的起始位置。

3、初始化最小差值为整型最大值INT\_MAX。

4、不断移动指针pa和pb，比较当前两个指针位置的值的差的绝对值与最小差值的大小关系，更新最小差值和结果。

5、如果a[pa]小于b[pb]，则移动指针pa；否则移动指针pb。

6、当任意一个指针到达数组末尾时，停止循环。

下面是具体的实现：

class Solution {

public:

int smallestDifference(vector<int>& a, vector<int>& b) {

sort(a.begin(), a.end());

sort(b.begin(), b.end());

long long minDiff = LLONG\_MAX;

int pa = 0, pb = 0;

while (pa < a.size() && pb < b.size()) {

long long diff = abs((long long)a[pa] - (long long)b[pb]);

minDiff = min(minDiff, diff);

if (a[pa] < b[pb]) {

pa++;

} else {

pb++;

}

}

return (int)minDiff;

}

};

在这个实现中，我们使用了长整型long long来存储差值，以防止溢出。然后，我们使用双指针来逐步找到具有最小差绝对值的一对数值。