# 题目

给你两个下标从 0 开始长度为 n 的整数排列 A 和 B 。

A 和 B 的 前缀公共数组 定义为数组 C ，其中 C[i] 是数组 A 和 B 到下标为 i 之前公共元素的数目。

请你返回 A 和 B 的 前缀公共数组 。

如果一个长度为 n 的数组包含 1 到 n 的元素恰好一次，我们称这个数组是一个长度为 n 的 排列 。

示例 1：

输入：A = [1,3,2,4], B = [3,1,2,4]

输出：[0,2,3,4]

解释：i = 0：没有公共元素，所以 C[0] = 0 。

i = 1：1 和 3 是两个数组的前缀公共元素，所以 C[1] = 2 。

i = 2：1，2 和 3 是两个数组的前缀公共元素，所以 C[2] = 3 。

i = 3：1，2，3 和 4 是两个数组的前缀公共元素，所以 C[3] = 4 。

示例 2：

输入：A = [2,3,1], B = [3,1,2]

输出：[0,1,3]

解释：i = 0：没有公共元素，所以 C[0] = 0 。

i = 1：只有 3 是公共元素，所以 C[1] = 1 。

i = 2：1，2 和 3 是两个数组的前缀公共元素，所以 C[2] = 3 。

提示：

1 <= A.length == B.length == n <= 50

1 <= A[i], B[i] <= n

题目保证 A 和 B 两个数组都是 n 个元素的排列。

# 分析

class Solution {

public:

vector<int> findThePrefixCommonArray(vector<int>& A, vector<int>& B) {

// 以数组A为基准进行检查

int n = A.size();

// cnt[x]表示元素x到目前为止一共出现了几次

int cnt[n+1];

memset(cnt, 0, sizeof(cnt));

vector<int> ans;

// now标识目前枚举到的前缀中，一共有几个公共的元素

int now = 0;

for (int i = 0; i < n; i++) {

// 检查A[i],B[i]是否出现了两次

if (++cnt[A[i]] == 2) now++;

if (++cnt[B[i]] == 2) now++;

ans.push\_back(now);

}

return ans;

}

};