

## 通过翻转子数组使两个数组相等

给你两个长度相同的整数数组 `target` 和 `arr` 。

每一步中，你可以选择 `arr` 的任意 **非空子数组** 并将它翻转。你可以执行此过程任意次。

如果你能让 `arr` 变得与 `target` 相同，返回 `True`；否则，返回 `False` 。

### 示例 1：

输入：`target = [1,2,3,4]`，`arr = [2,4,1,3]`

输出：`true`

解释：你可以按照如下步骤使 `arr` 变成 `target`：

1- 翻转子数组 `[2,4,1]`，`arr` 变成 `[1,4,2,3]`

2- 翻转子数组 `[4,2]`，`arr` 变成 `[1,2,4,3]`

3- 翻转子数组 `[4,3]`，`arr` 变成 `[1,2,3,4]`

上述方法并不是唯一的，还存在多种将 `arr` 变成 `target` 的方法。

### 示例 2：

输入：`target = [7]`，`arr = [7]`

输出：`true`

解释：`arr` 不需要做任何翻转已经与 `target` 相等。

### 示例 3：

输入：`target = [1,12]`，`arr = [12,1]`

输出：`true`

### 示例 4：

输入：`target = [3,7,9]`，`arr = [3,7,11]`

输出：`false`

解释：`arr` 没有数字 9，所以无论如何也无法变成 `target`。

### 示例 5：

输入：`target = [1,1,1,1,1]`，`arr = [1,1,1,1,1]`

输出: true

### 解题思路

若任意翻转总会相等说明 二者元素个数及元素值相同

```
class Solution {
public:
    bool canBeEqual(vector<int>& target, vector<int>& arr) {
        sort(target.begin(),target.end());
        sort(arr.begin(),arr.end());
        for(int i=0;i<target.size();i++)
        {
            if(target[i]!=arr[i])
                return false;
        }
        return true;
    }
};
```