

给定两个大小相等的数组 **A** 和 **B**，**A** 相对于 **B** 的**优势**可以用满足  $A[i] > B[i]$  的索引 **i** 的数目来描述。

返回 **A** 的**任意**排列，使其相对于 **B** 的优势最大化。

### 示例 1:

输入:  $A = [2, 7, 11, 15]$ ,  $B = [1, 10, 4, 11]$

输出:  $[2, 11, 7, 15]$

### 示例 2:

输入:  $A = [12, 24, 8, 32]$ ,  $B = [13, 25, 32, 11]$

输出:  $[24, 32, 8, 12]$

### 提示:

1.  $1 \leq A.length = B.length \leq 10000$
2.  $0 \leq A[i] \leq 10^9$
3.  $0 \leq B[i] \leq 10^9$

### 解题思路:

两个无序数组的比较总是很麻烦的，我们如果把他们变成有序数组，那么是不是比较起来就很方便了一——核心理想。

1，两个数组排序，注意排序之前要保留一个数组的当前顺序。

2，对排序后的数组进行优势洗牌，我这里由于数组从小到大排序，比较的时候只需要逐个比较，不符合优势的直接放到最后一位，即把不符合优势的最小数去对应另一个数组的最大数，怎么说呢，有点像田忌赛马。

3，我们对有序数组进行优势洗牌后，还好还原成原始顺序，这时候，通过之前保留的一个数组顺序和他对应的排序数组的映射关系可以找出洗牌后数组的原始顺序。

```
class Solution {
public:
    vector<int> advantageCount(vector<int>& A, vector<int>& B) {
        vector<int> res(A.size(), 0);
        vector<int> temp(A.size(), 0);
        vector<int> C(B);
```

```

sort(A.begin(), A.end());
sort(C.begin(), C.end());
int n = A.size();
for (int i = 0, j=0, K=0; i < A.size(); i++, j++)
{
    if (A[i] <= C[j])
    {
        temp[--n] = A[i];
        j--;
    }
    else
    {
        temp[K++] = A[i];
    }
}

```

```

for (int i = 0; i < A.size(); i++)
{
    for (int j = 0; j < A.size(); j++)
    {
        if (B[i] == C[j])
        {
            res[i] = temp[j];
            C[j] = -1;
            break;
        }
    }
}

```

```

return res;

```

```

}

```

```

};

```