

给你输入两个长度相等的数组nums1和nums2，请你重新组织nums1中元素的位置，使得nums1的「优势」最大化。

如果nums1[i] > nums2[i]，就是说nums1在索引i上对nums2[i]有「优势」。优势最大化也就是说让你重新组织nums1，尽可能多的让nums[i] > nums2[i]。

分析：

1. 对此类排序，首先需要知道nums1，nums2的最值比较，如果nums1的最大值比不过nums2的最值，则用最小值对付；
2. 如果能够胜过，则进行下一轮对比。

```
class Solution {
public:
    std::vector<int> advantageCount(std::vector<int>& nums1,
                                   std::vector<int>& nums2) {
        int len = nums1.size();
        for (int i = 0; i < nums2.size(); ++i) {
            maxpq.push(std::make_pair(i, nums2[i]));
        }

        std::vector<int> res(len, 0);

        sort(nums1.begin(), nums1.end());
        int left = 0, right = len - 1; // 分别对应nums1中的最大值和最小值

        while (!maxpq.empty()) {
            std::pair<int, int> tmp = maxpq.top();
            maxpq.pop();
            int index = tmp.first, value = tmp.second;
            if (nums1[right] > value) {
                res[index] = nums1[right--];
            } else {
                res[index] = nums1[left++];
            }
        }

        return res;
    }

private:
    struct greater {
        bool operator()(std::pair<int, int> a, std::pair<int, int> b) {
            return b.second >= a.second;
        }
    };

    std::priority_queue<std::pair<int, int>,
                        std::vector<std::pair<int, int>>,
                        greater>
```

```
};    maxpq;
```