1331. Rank Transform of an Array

Given an array of integers `arr`, replace each element with its rank.

The rank represents how large the element is. The rank has the following rules:

- Rank is an integer starting from 1.
- The larger the element, the larger the rank. If two elements are equal, their rank must be the same.
- Rank should be as small as possible.

Example 1:

```
Input: arr = [40,10,20,30]
Output: [4,1,2,3]
Explanation: 40 is the largest element. 10 is the smallest. 20 is the second
smallest. 30 is the third smallest.
```

Example 2:

```
Input: arr = [100,100,100]
Output: [1,1,1]
Explanation: Same elements share the same rank.
```

Example 3:

```
Input: arr = [37,12,28,9,100,56,80,5,12]
Output: [5,3,4,2,8,6,7,1,3]
```

Constraints:

- `0 <= arr.length <= ` 10^5
- -10^9 `<= arr[i] <= ` 10^9

1331. Rank Transform of an Array

```
/*
    Sorting+hash map
    Time complexity: O(nlogn)
    Space complexity: O(n)
*/
class Solution{
    public:
        std::vector<int> arrayRankTransform(std::vector<int>& arr){
            int n=arr.size();

            if(n==0) return {};

            std::vector<int> saved_arr=arr;

            std::sort(arr.begin(),arr.end());

            std::unordered_map<int,int> ranks;

            int rank=1;
            ranks[arr[0]]=1;

            for(int i=1;i<n;++i){
                if(arr[i-1]!=arr[i]) rank++;
                ranks[arr[i]]=rank;
            }

            std::vector<int> ans;
            for(auto& e: saved_arr){
                ans.push_back(ranks[e]);
            }

            return ans;
        }
};
```