

相对名次

给出 **N** 名运动员的成绩，找出他们的相对名次并授予前三名对应的奖牌。前三名运动员将会被分别授予 “金牌”，“银牌” 和“铜牌” (“Gold Medal", "Silver Medal", "Bronze Medal")。

(注：分数越高的选手，排名越靠前。)

示例 1:

输入: [5, 4, 3, 2, 1]

输出: ["Gold Medal", "Silver Medal", "Bronze Medal", "4", "5"]

解释: 前三名运动员的成绩为前三高的，因此将会分别被授予 “金牌”，“银牌”和“铜牌” ("Gold Medal", "Silver Medal" and "Bronze Medal").

余下的两名运动员，我们只需要通过他们的成绩计算将其相对名次即可。

```
/*class Solution {
public:
    string trans(int i) {
        if (i == 0) return "Gold Medal";
        if (i == 1) return "Silver Medal";
        if (i == 2) return "Bronze Medal";
        return to_string(i + 1);
    }

    vector<string> findRelativeRanks(vector<int>& score) {
        vector<string> res;
        if(score.size()==0)
        {
            return {};
        }
        sort(score.begin(),score.end());
        for(int i=score.size();i>0;i--)
        {
            res.push_back(trans(score.size()-i));
        }
        return res;
    }
};*/

class Solution {
public:
    vector<string> findRelativeRanks(vector<int>& nums) {
        //通过 hashmap 进行映射 实现值和索引的映射关系
        unordered_map<int,string> map;
```

```
//首先保存 nums 原有顺序
vector<int> tmp=nums;
//实现逆序排序 从大到小
sort(nums.rbegin(),nums.rend());
//索引为 0 1 2 在 map 中映射第一大 第二大 第三大值
for(int i=0;i<nums.size();i++)
{
    if(i==0)
    {
        map[nums[i]]="Gold Medal";
    }
    if(i==1)
    {
        map[nums[i]]="Silver Medal";
    }
    if(i==2)
    {
        map[nums[i]]="Bronze Medal";
    }
    if(i>=3)
    {
        map[nums[i]]=to_string(i + 1);
    }
}
//通过在原数组中遍历值在 map 中映射
vector<string> res;
for(int i=0;i<nums.size();i++)
{
    res.push_back(map[tmp[i]]);
}
return res;
}
};
```