

离散化（相同元素相等）

```
#include <bits/stdc++.h>
using namespace std;
#define MAXN 100005

int main(){
    //n 原数组大小 num 原数组中的元素 lsh 离散化的数组 cnt 离散化后的数组大小
    int lsh[MAXN],cnt,num[MAXN] , n;
    cin >> n;
    for(int i=1; i<=n; i++){
        scanf("%d",&num[i]);
        lsh[i]=num[i];
    }
    sort(lsh+1,lsh+n+1);
    cnt=unique(lsh+1,lsh+n+1)-lsh-1;
    for(int i=1;i<=n;i++) num[i]=lower_bound(lsh+1,lsh+cnt+1,num[i])-lsh;
    for(int i=1;i<=n;i++) cout << num[i] << " ";
    return 0;
}
```

离散化（相同元素不相等）

```
#include <bits/stdc++.h>
using namespace std;
#define MAXN 100005

struct Node {
    int data,id;
    bool operator < (const Node &a) const {
        return data < a.data;
    }
};
Node num[MAXN];
int rank[MAXN],n;
int main(){
    cin >> n;
    for(int i=1; i<=n; i++) {
        scanf("%d",&num[i].data);
        num[i].id = i;
    }
    sort(num+1 , num+n+1);
    for(int i=1; i<=n; i++) rank[num[i].id] = i;
    for(int i=1; i<=n; i++) cout << rank[i] << " ";
    return 0;
}
```

