排序

sort简介

sort函数包含在头文件`<algorithm>`中,在sort函数使用前需要#include`<algorithm>`或使用万能头文件。
sort是C++标准库中的一个函数模板，用于对指定范围内的元素进行排序。
sort算法使用的是快速排序(QuickSort)或者类似快速排序的改进算法。具有较好的平均时间复杂度，一般为

$$O(nlogn)$$

sort用法

1

```
sort(起始位置， 结束地址的下一位， *比较函数);
```

*处的比较函数为空的话，默认小于号`<`（升序）

```cpp
int a[1000];

int n;
//读取数组大小
cin >> n;

//读取元素
for(int i = 1;i <= n; ++ i)
    cin >> a[i];

//对数组进行排序
sort(a + 1, a + n + 1);// [1,n+1)左闭右开区间

//输出
for(int i = 1; i <= n; ++ i)
    cout << a[i] << ' ';
```

2

```
sort(起始地址， 结束地址的下一位， *比较函数);
```

这里的起始地址指迭代器begin()

```cpp
// 初始化v
vector<int> v = {5, 1, 3, 9, 11};

// 对数组进行排序
sort(v.begin(), v.end());

// 输出
for (int i = 0; i < v.size(); ++ i)
    cout << v[i] << ' ';
//也可以这么写:
//for (auto i : v)cout << i << ' ';
```

关于vector相关的拓展知识，请看后文。

3. 自定义比较函数

sort默认使用小于号进行排序，如果自定义比较规则，可以传入第三个参数，可以是函数或lambda表达式。

```
bool cmp(const int &u, const int &v)
{
    return u > v;
}
int main()
{
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    //初始化v
    vector<int> v = {5, 1, 3, 9, 11};

    //对数组进行排序，降序排列
    sort(v.begin(), v.end(), cmp);

    //输出
    for (int i = 0; i < (int)v.size(); ++ i)
        cout << v[i] << ' ';
}
```

使用lambda表达式（匿名函数）：

```
//初始化v
vector<int> v = {5, 1, 3, 9, 11};

//对数组进行排序，降序排列
sort(v.begin, v.end(), [](const int &u, const int &v)
{
    return u > v;
})

//输出
for (int i = 0;i < v; ++ i)cout << v[i] << ' ';
```

拓展知识

1. C++ 数组 之 vector

vector中的数据类型T可以表示任何数据类型，比如int, string, class, vector（构建多维数组）等等，就像一个可以放下任何东西的容器，所以vector也常被称作为**容器**，c++中的不同种类的容器拥有很多相同的操作，因此string的很多操作方法可以直接用在vector上。

表1 vector对象的定义和初始化方式

类型	解释
vector<T> v1	v1是一个元素类型为T的空vector
vector<T> v2(v1)	使用v2中所有元素初始化v1
vector<T> V2 = V1	使用v2中所有元素初始化v1
vector<T>	