

# sort()函数详解

辉小歌 于 2021-03-09 20:37:47 发布

## 目录

- 前言
- 使用sort
- 实现比较函数cmp
- 结构体数组的排序
- 容器的排序

## 前言

顾名思义，sort()就是用来排序的函数，它根据具体情况使用不同的排序方法，效率较高。  
一般来说，不推荐使用C语言中的 **qsort** 函数，原因是qsort用起来比较繁琐，涉及很多指针的操作。  
而且sort在实现中避免了经典 **快速排序** 中可能出现的会导致实际复杂度退化到 $O(n^2)$ 的极端情况。  
话不多说了，下面我们来了解一下sort()函数的使用。

## 使用sort

### 需要的头文件

```
1 | #include<algorithm>
```

### 需要的其他东西:

```
1 | using namespace std;
```

### 使用的方式如下:

```
1 | sort(首元素地址(必填), 尾元素地址的下一个地址(必填), 比较函数(非必填));
```

比较函数可以根据自身需要填写，如果不写比较函数，则默认对前面给出的区间进行递增排序。

### 例子:对int型和double型数组排序

```
1 | #include<cstdio>
2 | #include<algorithm>
3 | using namespace std;
4 | #define N 5
5 | int main(void)
6 | {
7 |     int a[N]={12,3,13,45,4};
8 |     double b[N]={2.1,333.12,137.56,35.013,4.78};
9 |     sort(a,a+5);//全部排序
10 |    for(int i=0;i<N;i++)
11 |    {
12 |        printf("%d ",a[i]);
13 |    }
14 |    printf("\n");
15 |    sort(b,b+3);//排前三个数
16 |    for(int i=0;i<N;i++)
17 |    {
18 |        printf("%lf ",b[i]);
19 |    }
20 |    return 0;
21 | }
```

选择C:\Users\26846\Desktop\test\test\Untitled1.exe

```
3 4 12 13 45
2.100000 137.560000 333.120000 35.013000 4.780000
-----
Process exited after 0.08237 seconds with return value 0
请按任意键继续. . .
```

[https://blog.csdn.net/qq\\_46527915](https://blog.csdn.net/qq_46527915)

对char型数组排序：

Untitled1.cpp

```
1 #include<cstdio>
2 #include<algorithm>
3 using namespace std;
4 int main(void)
5 {
6     char a[]={'h','e','l','l','o'};
7     sort(a,a+5); //全部排序
8     for(int i=0;i<5;i++)
9     {
10         printf("%c ",a[i]);
11     }
12     printf("\n");
13     return 0;
14 }
```

C:\Users\26846\Desktop\test\test\Untitled1.exe

e h l l o

Process exited after 0.08659 seconds with return value 0  
请按任意键继续. . .

[https://blog.csdn.net/qq\\_46527915](https://blog.csdn.net/qq_46527915)

- 1 其实你会发现一个规律。
- 2 例如：你要排前三个数 `sort(a,a+3);`
- 3 你要排前六个数 `sort(a,a+6);`
- 4 你要排前n个数 `sort(a,a+n);`

我们应该知道，如果需要对序列进行排序，那么序列中的元素一定要有可比性，因此需要制定排序规则来建立这种可比性。特别是像结构体，本身并没有大小关系，需要人为制订比较的规则。  
sort的第三个可选参数就是compare函数(一般写作cmp函数)，用来实现这个规则。

## 实现比较函数cmp

如果想要从大到小排序，则使用比较函数cmp来“告诉”sort何时要交换元素(让元素的大小比较关系反过来)。

id1.cpp

```
#include<cstdio>
#include<algorithm>
using namespace std;
bool cmp(int a,int b)
{
    return a>b; //可以理解为: 当a>b时把a放在b前面
}
int main(void)
{
    int a[]={12,45,67,1,3};
    sort(a,a+5,cmp); //全部排序
    for(int i=0;i<5;i++)
    {
        printf("%d ",a[i]);
    }
    printf("\n");
    return 0;
}
```

选择C:\Users\26846\Desktop\test\test\Untitled1.exe

67 45 12 3 1

Process exited after 0.08012 seconds with return value 0  
请按任意键继续. . .

[https://blog.csdn.net/qq\\_46527915](https://blog.csdn.net/qq_46527915)

led1.cpp

```
#include<stdio>
#include<algorithm>
using namespace std;
bool cmp(double a,double b)
{
    return a>b;//可以理解为:当a>b时把a放在b前面
}
int main(void)
{
    double a[]={12.45,1233.4,45.56,76.3,12.3};
    sort(a,a+5,cmp);//全部排序
    for(int i=0;i<5;i++)
    {
        printf("%lf ",a[i]);
    }
    printf("\n");
    return 0;
}
```

C:\Users\26846\Desktop\test\test\Untitled1.exe

1233.400000 76.300000 45.560000 12.450000 12.300000
-----
Process exited after 0.08314 seconds with return value 0
请按任意键继续. . .

https://blog.csdn.net/qq\_46527915

Untitled1.cpp

```
1 #include<stdio>
2 #include<algorithm>
3 using namespace std;
4 bool cmp(char a,char b)
5 {
6     return a>b;//可以理解为:当a>b时把a放在b
7 }
8 int main(void)
9 {
10     char a[]={'h','e','l','l','o'};
11     sort(a,a+5,cmp);//全部排序
12     for(int i=0;i<5;i++)
13     {
14         printf("%c ",a[i]);
15     }
16     printf("\n");
17     return 0;
18 }
```

选择C:\Users\26846\Desktop\test\test\Untitled1.exe

o l l h e
-----
Process exited after 0.07714 seconds with return value 0
请按任意键继续. . .

https://blog.csdn.net/qq\_46527915

## 结构体数组的排序

现在定义了如下的结构体:

```
1 struct student
2 {
3     int x,y;
4 };
```

如果想将结构体数组按照x从大到小排序,可以这样写cmp函数

```
1 bool cmp(student a,student b)
2 {
3     return a.x>b.x;//按x值从大到小对结构体排序
4 }
```

结构体数组在排序时,必须写cmp函数来规定规则。

例子如下:

```

1 #include<cstdio>
2 #include<algorithm>
3 using namespace std;
4 struct student
5 {
6     int x,y;
7 };
8 bool cmp(student a,student b)
9 {
10     return a.x>b.x;//按x值从大到小对结构体排序
11 }
12 int main(void)
13 {
14     struct student a[5];
15     a[0].x=2;
16     a[0].y=2;
17     a[1].x=1;
18     a[1].y=3;
19     a[2].x=3;
20     a[2].y=1;
21     sort(a,a+3,cmp);//全部排序
22     for(int i=0;i<3;i++)
23     {
24         printf("%d %d\n",a[i].x,a[i].y);
25     }
26     return 0;
27 }
28

```

C:\Users\26846\Desktop\test\test\Untitled1.exe

```

3 1
2 2
1 3

```

Process exited after 0.09771 seconds with return value 0  
请按任意键继续. . .

[https://blog.csdn.net/qq\\_46527915](https://blog.csdn.net/qq_46527915)

如果想要先按x从大到小排序(即一级排序), 但当x相等的情况下, 按照y的大小从小到大排序(即二级排序)。那么cmp的写法是:

```

1 bool cmp(student a,student b)
2 {
3     if(a.x==b.x)//x值相等时按y从小到大排
4     {
5         return a.y<b.y;
6     }
7     return a.x>b.x;//按x值从大到小对结构体排序
8 }
9 或
10 bool cmp(student a,student b)
11 {
12     if(a.x != b.x)//x值不相等时按x从大到小排
13     return a.x>b.x;
14     else
15     return a.y<b.y;//x相等时按y值从小到大对结构体排序
16 }

```

```

1 #include<stdio>
2 #include<algorithm>
3 using namespace std;
4 struct student
5 {
6     int x,y;
7 };
8 bool cmp(student a,student b)
9 {
10     if(a.x==b.x)//x值相等时按y从小到大排
11     {
12         return a.y<b.y;
13     }
14     return a.x>b.x;//按x值从大到小对结构体排序
15 }
16 int main(void)
17 {
18     struct student a[5];
19     a[0].x=2;
20     a[0].y=2;
21     a[1].x=2;
22     a[1].y=3;
23     a[2].x=2;
24     a[2].y=1;
25     sort(a,a+3,cmp);//全部排序
26     for(int i=0;i<3;i++)
27     {
28         printf("%d %d\n",a[i].x,a[i].y);
29     }
30     return 0;
31 }

```

C:\Users\26846\Desktop\test\test\Untitled1.exe

```

2 1
2 2
2 3

```

Process exited after 0.08534 seconds with return value 0  
请按任意键继续. . .

[https://blog.csdn.net/qq\\_46527915](https://blog.csdn.net/qq_46527915)

## 容器的排序

在STL标准容器中，只有vector、string、deque是可以使用sort的。

这是因为像set、map这种容器是用红黑树实现的(了解即可)，元素本身有序，故不允许使用sort排序。

例子: vector:

```

1 #include<stdio>
2 #include<algorithm>
3 #include<vector>
4 using namespace std;
5 bool cmp(int a,int b)
6 {
7     return a>b;
8 }
9 int main(void)
10 {
11     vector<int> vi;
12     vi.push_back(3);
13     vi.push_back(1);
14     vi.push_back(2);
15     sort(vi.begin(),vi.end(),cmp);
16     for(int i=0;i<3;i++)
17     {
18         printf("%d\n",vi[i]);
19     }
20     return 0;
21 }

```

C:\Users\26846\Desktop\test\test\Untitled1.exe

```

3
2
1

```

Process exited after 0.09001 seconds with return value 0  
请按任意键继续. . .

[https://blog.csdn.net/qq\\_46527915](https://blog.csdn.net/qq_46527915)

例: string:

Intitled1.cpp

```
1 #include<cstdio>
2 #include<algorithm>
3 #include<string>
4 #include<iostream>
5 using namespace std;
6 int main(void)
7 {
8     string a[3]={"bbbb","cc","aaa"};
9     sort(a,a+3); // 将string型数组按字典序从小到大输出
10    for(int i=0;i<3;i++)
11    {
12        cout<<a[i]<<endl;
13    }
14    return 0;
15 }
```

C:\Users\26846\Desktop\test\test\Untitled1

aaa  
bbbb  
cc

Process exited after 0.1327 seconds  
请按任意键继续. . .

[https://blog.csdn.net/qq\\_46527915](https://blog.csdn.net/qq_46527915)

Intitled1.cpp

```
1 #include<cstdio>
2 #include<algorithm>
3 #include<string>
4 #include<iostream>
5 using namespace std;
6 bool cmp(string str1,string str2)
7 {
8     return str1.length()<str2.length(); // 将string的长度从小到大排序
9 }
10 int main(void)
11 {
12     string a[3]={"bbbb","cc","aaa"};
13     sort(a,a+3,cmp);
14     for(int i=0;i<3;i++)
15     {
16         cout<<a[i]<<endl;
17     }
18     return 0;
19 }
```

选择C:\Users\26846\Desktop\test\test\Untitled1.exe

cc  
aaa  
bbbb

Process exited after 0.119 seconds with return val  
请按任意键继续. . .

[https://blog.csdn.net/qq\\_46527915](https://blog.csdn.net/qq_46527915)