

个人资料



zjufirefly



访问： 6421次

积分： 464

等级： 2

排名： 千里之外

原创： 42篇 转载： 4篇

译文： 0篇 评论： 1条

文章搜索

文章分类

C++ (14)

其他 (5)

链接 (13)

linux (13)

shell (7)

操作 (11)

文章存档

2014年12月 (8)

2014年11月 (7)

2014年10月 (8)

2014年09月 (6)

2014年08月 (4)

展开

阅读排行

git操作命令 (327)

blktrace (287)

mysql常用命令 (280)

python (270)

cgroups (262)

linux下如何配置openvpn (260)

2015年4月微软MVP申请 《京东技术解密》有奖试读，礼品大放送 “我的2014”年度征文活动火爆开启 CSDN 2014博客之星

如何为排序，查找，set，map提供自定义比较功能

分类： C++ 2014-05-04 22:12 77人阅读 评论(0) 收藏 编辑 删除

set按照operator<重载操作符比较规则从小到大排序

```
set<type> setValue;  
set<type,less<type>> setValue;
```

set按照operator<重载操作符比较规则从大到小排序

```
set<type,greater<type>> setValue;
```

set按照自定义比较规则从大到小排序

```
set<type, compareType> setValue;  
compareType为实现bool operater(const type& v1, const type& v2)的类。  
TODO：程序范例。
```

map按照operator<重载操作符比较规则从小到大排序

```
map<keyType, valueType> setValue;  
map<keyType, valueType,less<keyType>> setValue;
```

map按照operator<重载操作符比较规则从大到小排序

```
map<keyType, valueType,greater<keyType>> setValue;
```

map按照自定义比较规则从大到小排序

```
map<keyType, valueType,compareKeyType> setValue;  
compareKeyType为实现bool operater(const keyType& v1, const keyType& v2)的类。  
TODO：程序范例。
```

模板函数min按照自定义比较规则返回最小值

```
template<class T, class Compare>  
inline const T& min(const T& a, const T& b, Compare) {  
    return comp(a, b) ? a : b;  
}  
Compare为函数或者仿函数
```

模板函数max按照自定义比较规则返回最小值

```
template<class T, class Compare>  
inline const T& max(const T& a, const T& b, Compare) {  
    return comp(b, a) ? a : b;  
}  
Compare为函数或者仿函数
```

模板函数max_element按照operator<重载操作符比较规则返回最大值

vim-ctags-taglist-netrw	(249)
性能测试命令字段解释	(213)
正则表达式	(211)
SystemTap	(195)

评论排行	
为什么需要auto_ptr_ref	(1)
valgrind内存检查	(0)
cgroups	(0)
linux下如何配置openvpn	(0)
git操作命令	(0)
thrift	(0)
Apache开源软件	(0)
vim-ctags-taglist-netrw	(0)
vim常用配置	(0)
shell按行读取字符串，并	(0)

推荐文章	
* 挣扎与彷徨——我的2014	
* 校招回忆录——小米篇	
* Android UI-自定义日历控件	
* 30岁程序员回顾人生、展望未来	
* 2014年终总结，我决定要实现的三个目标	
* Android 启动问题——黑屏 死机解决方法	

最新评论	
为什么需要auto_ptr_ref	
zjufirefly: explicit auto_ptr_ref(Tp1* __p): _M_ptr(__p){}...	

[cpp]

```
01. template<class _FwdIt> inline
02.     _FwdIt _Max_element(_FwdIt _First, _FwdIt _Last)
03. { // find largest element, using operator<
04.     _DEBUG_RANGE(_First, _Last);
05.     _FwdIt _Found = _First;
06.     if (_First != _Last)
07.         for (; ++_First != _Last; )
08.             if (_DEBUG_LT(*_Found, *_First))
09.                 _Found = *_First;
10.     return (_Found);
11. }
```

模板函数max_element按照自定义比较规则返回最大值

[cpp]

```
01. template<class _FwdIt,
02.         class _Pr> inline
03.     _FwdIt _Max_element(_FwdIt _First, _FwdIt _Last, _Pr _Pred)
04. { // find largest element, using _Pred
05.     _DEBUG_RANGE(_First, _Last);
06.     _DEBUG_POINTER(_Pred);
07.     _FwdIt _Found = _First;
08.     if (_First != _Last)
09.         for (; ++_First != _Last; )
10.             if (_DEBUG_LT_PRED(_Pred, *_Found, *_First))
11.                 _Found = *_First;
12.     return (_Found);
13. }
```

[cpp]

```
01. #define _DEBUG_LT_PRED(pred, x, y) _Debug_lt_pred(pred, x, y, __FILE__, __LINE__)
```

[cpp]

```
01. template<class _Pr, class _Ty1, class _Ty2> inline
02.     bool __CLRCALL_OR_CDECL _Debug_lt_pred(_Pr _Pred, const _Ty1& _Left, const _Ty2& _Rig
03.     const wchar_t *_Where, unsigned int _Line)
04. { // test if _Pred(_Left, _Right) and _Pred is strict weak ordering
05.     if (!_Pred(_Left, _Right))
06.         return (false);
07.     else if (_Pred(_Right, _Left))
08.         _DEBUG_ERROR2("invalid operator<", _Where, _Line);
09.     return (true);
10. }
```

上一篇 为什么需要auto_ptr_ref

下一篇 为何迭代器前置式递增比后置式递增效率要高

顶 0

踩 0

猜你在找

服务器设计

C ++ 编程思想卷二 笔记

面试资料一

MFC学习总结67个技巧

android linux 基础知识总结

最佳vim技巧

Sphinx 099Coreseek 32 参考手册

一个程序员的Java和C++学习之路整理

top-10-algorithms-for-coding-interview 笔记

android linux 最全的基础知识总结

1 期货贵金属

2 博客搬家工具

3 微型消防车

4 电液推杆

5 778游戏下载

6 北斗手机报价

7 海尔全时云会议

8 紫薇苗圃

9 磁座钻

10 一号棋牌

11 新疆翻译

12 家饰加盟

13 广西河粉机

14 外贸代加工

15 深林舞会

16 水晶灯价格

17 最新银杏树价格

18 省立医院西院

19 腰间盘膨出

20 盐雾试验设备

查看评论

暂无评论

发表评论

用户名: zjufirefly

评论内容:

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

apttech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap