

www.cnblogs.com

Darren

C++ 及算法

QQ留言

留言簿(3)

给我留言

查看公开留言

查看私人留言

随笔分类

动态规划(13)

数据结构(11)

搜索(9)

图论(10)

未分类(6)

ACMers

搜索

积分与排名

积分 - 52718

排名 - 188

最新随笔

1. 换个博客, 重新开始学习。。。

2. pku 1691 Painting A Board 状态压缩DP

3. HDU 1255

4. PKU 1151

5. 2009年ACM-ICPC亚洲区预选赛共设十五个赛区如下(按现场赛日期排序)

6. acmer必看的26个对acm态度

7. ZJU 3228 Searching the String (AC 自动机)

8. Pku 3169 Layout

9. Pku 1986 Distance Queries

10. Pku 1276 Cash Machine

C++ 博客 首页 新随笔 联系 管理

priority_queue用法

priority_queue 调用 STL里面的 make_heap(), pop_heap(), push_heap() 算法实现, 也算是堆的另外一种形式。

先写一个用 STL 里面堆算法实现的与真正的STL里面的 priority_queue 用法相似的 priority_queue, 以加深对 priority_queue 的理解

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

class priority_queue

{

private:

vector<int> data;

public:

void push(int t){

data.push_back(t);

push_heap(data.begin(), data.end());

}

void pop(){

pop_heap(data.begin(), data.end());

data.pop_back();

}

int top() { return data.front(); }

int size() { return data.size(); }

bool empty() { return data.empty(); }

http://www.cppblog.com/Darren/archive/2009/06/09/87224.html[2012/8/28 22:05:44]

最新评论 XML



1. re: Size Balanced Tree C++模板实现

ORZ!!!!!!!!!!学习了!

--Indeed

```
};

int main()
{
    priority_queue test;
    test.push( 3 );
    test.push( 5 );
    test.push( 2 );
    test.push( 4 );

    while( !test.empty() ){
        cout << test.top() << endl;
        test.pop(); }

    return 0;
}
```

STL里面的 `priority_queue` 写法与此相似，只是增加了模板及相关的迭代器什么的。

`priority_queue` 对于基本类型的使用方法相对简单。

他的模板声明带有三个参数，`priority_queue<Type, Container, Functional>`

`Type` 为数据类型，`Container` 为保存数据的容器，`Functional` 为元素比较方式。

`Container` 必须是用数组实现的容器，比如 `vector`, `deque` 但不能用 `list`。

STL里面默认用的是 `vector`。比较方式默认用 `operator<`，所以如果你把后面俩个

参数缺省的话，优先队列就是大顶堆，队头元素最大。

看例子

```
#include <iostream>
#include <queue>

using namespace std;

int main() {
    priority_queue<int> q;

    for( int i= 0; i< 10; ++i ) q.push( rand() );
    while( !q.empty() ){
        cout << q.top() << endl;
        q.pop();
    }
}
```

```

|   getchar();
|   return 0;
| }

```

如果要用到小顶堆，则一般要把模板的三个参数都带进去。

STL里面定义了一个仿函数 `greater<>`，对于基本类型可以用这个仿函数声明小顶堆

例子：

```

▢ #include <iostream>
  #include <queue>

  using namespace std;

▢ int main() {
  |   priority_queue<int, vector<int>, greater<int> > q;
  |
  |   for( int i= 0; i< 10; ++i ) q.push( rand() );
  ▢ while( !q.empty() ){
  |   |   cout << q.top() << endl;
  |   |   q.pop();
  |   }
  |
  |   getchar();
  |   return 0;
  | }

```

对于自定义类型，则必须自己重载 `operator<` 或者自己写仿函数
先看看例子：

```

▢ #include <iostream>
  #include <queue>

  using namespace std;

▢ struct Node{
  |   int x, y;
  |   Node( int a= 0, int b= 0 ):
  ▢       x(a), y(b) {}
  | }

▢ bool operator<( Node a, Node b ){
  |   if( a.x== b.x ) return a.y> b.y;
  |   return a.x> b.x;

```

```

    }

    int main() {
        priority_queue<Node> q;

        for( int i= 0; i< 10; ++i )
            q.push( Node( rand(), rand() ) );

        while( !q.empty() ){
            cout << q.top().x << ' ' << q.top().y << endl;
            q.pop();
        }

        getchar();
        return 0;
    }

```

自定义类型重载 `operator<` 后，声明对象时就可以只带一个模板参数。但此时不能像基本类型这样声明

```
priority_queue<Node, vector<Node>, greater<Node> >;
```

原因是 `greater<Node>` 没有定义，如果想用这种方法定义则可以按如下方式

例子：

```

#include <iostream>
#include <queue>

using namespace std;

struct Node{
    int x, y;
    Node( int a= 0, int b= 0 ):
        x(a), y(b) {}
};

struct cmp{
    bool operator() ( Node a, Node b ){
        if( a.x== b.x ) return a.y> b.y;

        return a.x> b.x; }
};

int main() {
    priority_queue<Node, vector<Node>, cmp> q;

    for( int i= 0; i< 10; ++i )
        q.push( Node( rand(), rand() ) );
}

```

```
while( !q.empty() ){
    cout << q.top().x << ' ' << q.top().y << endl;
    q.pop();
}

getchar();
return 0;
}
```

posted on 2009-06-09 18:00 [Darren](#) 阅读(4457) [评论\(5\)](#) [编辑](#) [收藏](#) [引用](#)

评论:

re: priority_queue用法 2009-06-24 17:26 | [superlong](#)
前来学习C++ 东西不错 [回复](#) [更多评论](#)

re: priority_queue用法 2009-06-26 18:30 | [gnaggnoyil](#)
骚啊.....写Pascal还得自己写..... [回复](#) [更多评论](#)

re: priority_queue用法 2009-10-26 18:27 | [huangzhifei](#)
好东西，牛B
Orz
..... [回复](#) [更多评论](#)

re: priority_queue用法 2011-05-21 10:56 | [I流](#)
不错哦 [回复](#) [更多评论](#)

re: priority_queue用法 2011-12-20 22:05 | [zz_lizhen](#)
好文章！对我很有帮助！学习了。 [回复](#) [更多评论](#)

[刷新评论列表](#)

[国家职业资格证书.NET/Java免费培训\(上海\)](#)

[博客园](#) [博问](#) [IT新闻](#) [C++ 程序员招聘](#)

标题

姓名

[主页](#)

验证码

*

5040

内容(提交失败后,可以通过“恢复上次提交”恢复刚刚提交的内容)

Remember Me?

[登录](#) [使用高级评论](#) [新用户注册](#) [返回首页](#) [恢复上次提交](#)

[使用Ctrl+Enter键可以直接提交]



IT新闻:

- [任天堂29日公布未来规划 避谈Wii U](#)
- [分析称iPhone 5美销量将突破5千万 远超4S](#)
- [腾讯联手12321打造首个立体手机安全防护网](#)
- [Facebook聘迪斯尼音乐厅设计师设计总部大楼](#)
- [夏普宣布在日本裁员2000人 遣散费3.4亿美元](#)

博客园首页随笔:

- [timestamp\(数据库中的数据类型\)](#)
- [JavaScript 高级程序设计 \(第3版\) 学习笔记6——初识对象](#)
- [Node.js能让Javascript写后端, 为啥不让Python写前端?](#)
- [Win7,VS2010与GIT结合](#)
- [\(6\)Django框架学习-Views,URLconf的advanced用法篇](#)

知识库:

- [JavaScript 项目优化总结](#)
- [TDD中的任务分解](#)
- [IE 浏览器的创新](#)

- [我和Google的故事](#)
- [依赖的故事之一](#)

网站导航: [博客园](#) [IT新闻](#) [BlogJava](#) [知识库](#) [程序员招聘](#) [管理](#)

Powered by: [博客园](#) 模板提供: [沪江博客](#) Copyright ©2012 Darren