

STL-队列简介

By 大肥羊

#include <queue>

- queue就是普通的队列，队列声明方式：
`queue <元素类型> 变量名`
- 比如 `queue <int> q;` 表示你声明了一个队列q，队列里的元素是int类型的
- 优点：不用手动维护队首和队尾指针，不易出错，动态分配空间，减少段错误
- 用途：普通BFS、最短路、格子题、模拟题

简单的bfs框架

```
queue <node> q;  
/*node代表一个结构体，比如表示位置的两个int  
变量x,y，以及走的步数step*/  
while(!q.empty()) //队列非空  
{  
    node ss=q.front(); //得到队首的元素  
    q.pop(); //抛出队首的元素  
    /*进行更新操作得到新的状态tt*/  
    q.push(tt); //将新的状态压入队列（队首）  
}
```

#include <priority_queue>

- `priority_queue <int> q;`
- 优先队列、最大堆、最小堆
- 抛出队首元素用的是`q.top()`,不是`q.front()`
- 默认抛出队列中最大的元素, 抛出最小可改成:
`priority_queue<int, vector<int>, greater<int> > q;`
- 优点: 简单易用, 减少代码量
- 用途: 步长大于1的bfs最短路、特定dp优化

重载运算符

```
struct node
{
    int x,y;
    friend bool operator < (node n1,node n2) //重载的是小于号
    {
        if(n1.x != n2.x)
            return n1.x > n2.x;
        else return n1.y > n2.y;
    }
};
priority_queue <node> q;
//优先抛出x最小的node， x相同则抛出y最小的node
//注意node里面一定要重载运算符， 否则用优先队列会编译错误
```

Deque

- 双端队列

`q.push_back(x)`; 往队尾push元素

`q.push_front(x)`; 往队首push元素

`q.pop_back()`; 抛出队尾元素

`q.pop_front()`; 抛出队首元素

注意

- 三种队列的push函数都要带一个参数x表示你push进去的元素
- 从队列pop元素是不需要参数的

参考例题

- [ZOJ3632](#):

很简单的dp，但需要线段树或者堆优化将复杂度降到 $n\log n$ ，使用优先队列的STL非常方便

- [ZOJ3531](#):

规则比较复杂的格子题模拟

- [BBQ](#):

7月集训的大鸡腿模拟题，用queue可以避免很多脑残错误

- [Dot Dot Dot](#)、[Fetus](#):

7月集训的格子题，AC数很少，但本质都是简单题，大家要多练习

参考链接

- <http://www.cplusplus.com/reference/stl/queue/>
- http://www.cplusplus.com/reference/stl/priority_queue/
- <http://www.cplusplus.com/reference/deque/deque/>