

04-H

栈与队列

队列接口与实现

邓俊辉

deng@tsinghua.edu.cn

也有个先来后到！甚么官人的伴当要换座头！老爷不换！

操作与接口

❖ 队列 (queue) 也是受限的序列

❖ 只能在队尾插入 (查询) :

- enqueue() / rear()

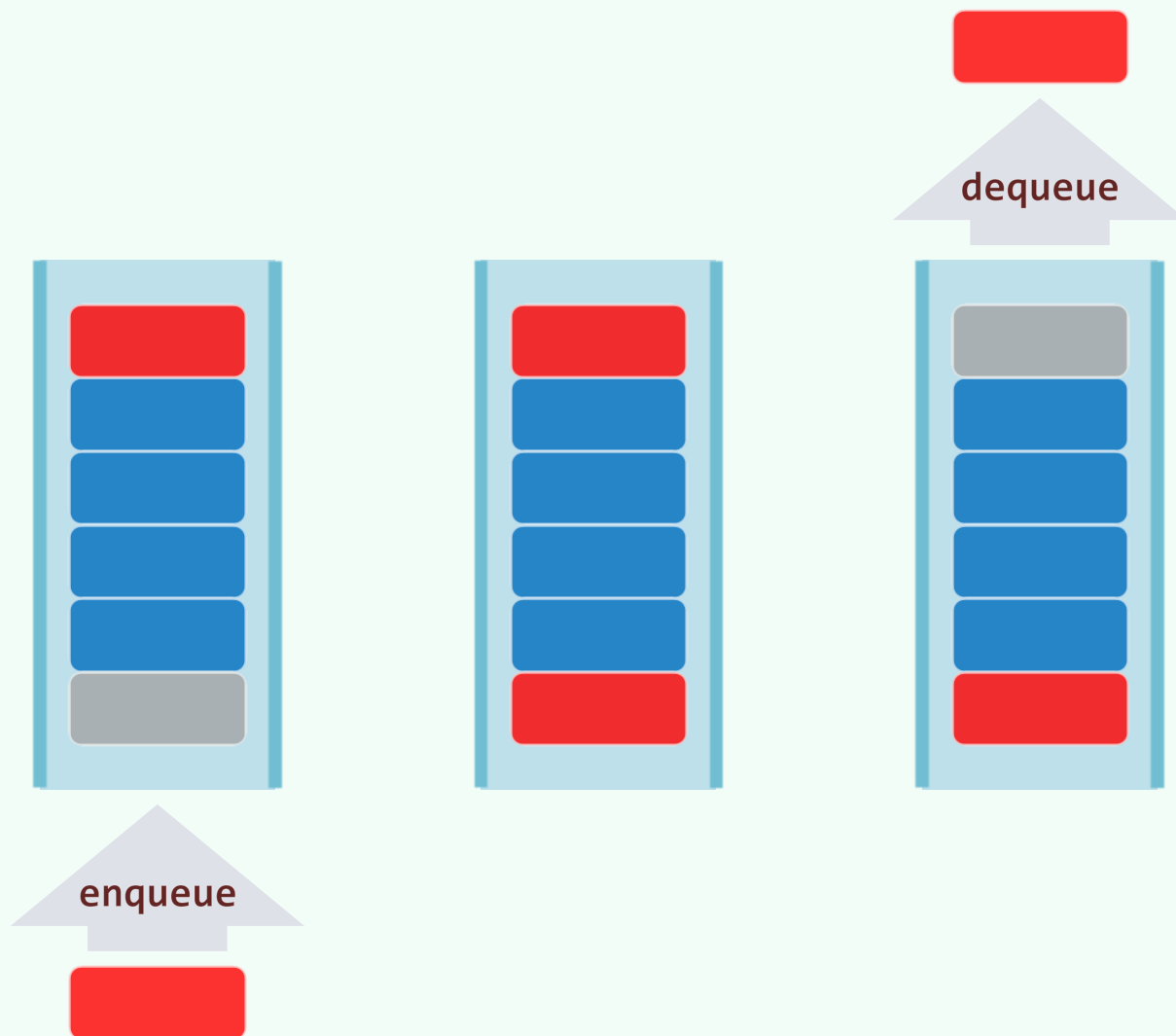
❖ 只能在队头删除 (查询) :

- dequeue() / front()

❖ 先进先出 (FIFO)

后进后出 (LILO)

❖ 扩展接口: getMax()...



实例

| 操作 | >-- 队列 --> | | | | 输出 |
|------------|------------|---|---|--|-------|
| Queue() | | | | | |
| empty() | | | | | true |
| enqueue(E) | E | | | | |
| enqueue(C) | C | E | | | |
| dequeue() | C | | | | E |
| enqueue(G) | G | C | | | |
| enqueue(C) | C | G | C | | |
| front() | C | G | C | | C |
| empty() | C | G | C | | false |

| 操作 | >-- 队列 --> | | | | | | 输出 |
|------------|------------|---|---|---|---|---|-------|
| enqueue(K) | K | C | G | C | | | |
| size() | K | C | G | C | | | 4 |
| enqueue(F) | F | K | C | G | C | | |
| empty() | F | K | C | G | C | | false |
| enqueue(G) | G | F | K | C | G | C | |
| dequeue() | G | F | K | C | G | | C |
| dequeue() | G | F | K | C | | | G |
| front() | G | F | K | C | | | C |
| size() | G | F | K | C | | | 4 |

实现：既然属于**线性序列**，故可直接基于向量或列表**派生**

```
template <typename T> class Queue: public List<T> { //原有接口一概沿用
```

```
public:
```

```
void enqueue( T const & e ) { insertLast( e ); } //入队
```

```
T dequeue() { return remove( first() ); } //出队
```

```
T & front() { return first()->data; } //队首
```

```
}; //以列表首/末端为队列头/尾——颠倒过来呢？
```

❖ 确认：如此实现的队列接口，均只需 $\mathcal{O}(1)$ 时间

❖ 课后：基于向量，派生出队列模板类——如此实现的各接口，效率如何？