

04-H 队列

#数据结构邓神

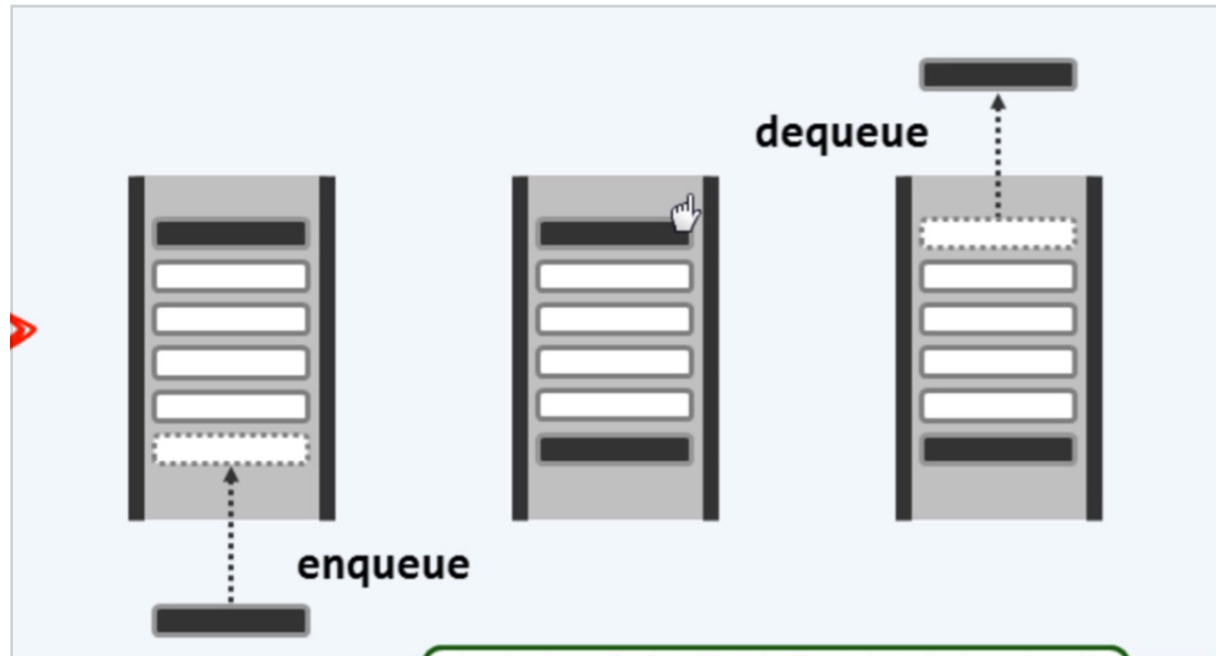
操作和接口

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()`



与栈的特性完全对称

FIFO LILO 先进先出，后进后出

实例

操作实例

操作	输出	队列（右侧为队头）			
Queue()					
empty()	true				
enqueue(5)		5			
enqueue(3)		3	5		
dequeue()	5	3			
enqueue(7)		7	3		
enqueue(3)		3	7	3	
front()	3	3	7	3	
empty()	false	3	7	3	

操作	输出	队列（右侧为队头）				
enqueue(11)		11	3	7	3	
size()	4	11	3	7	3	
enqueue(6)		6	11	3	7	3
empty()	false	6	11	3	7	3
enqueue(7)		7	6	11	3	7
dequeue()	3	7	6	11	3	7
dequeue()	7	7	6	11	3	
front()	3	7	6	11	3	
size()	4	7	6	11	3	

实现 模版类 (借助原来的List实现)

```
template <typename T> class Queue : public list<T> {
public:
    void enqueue(T const & e){
        insertAsLast(e);
    }
    T dequeue() {
        return remove(first());
    }
    T & front() {
        return first()->deta;
    }
};
```

如此实现的队列接口，所有的操作都只需要常数时间 $O(1)$