



CSDN学院 IT实战派

# 图解数据结构和算法

栈和队列

讲师：Samuel

# | 本章概述

■ 无处不在的栈和队列



队尾

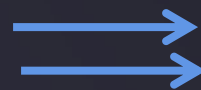
队首

先进先出FIFO

后进先出LIFO

好看  
这么  
心云  
我

栈顶  
栈底



# | Content

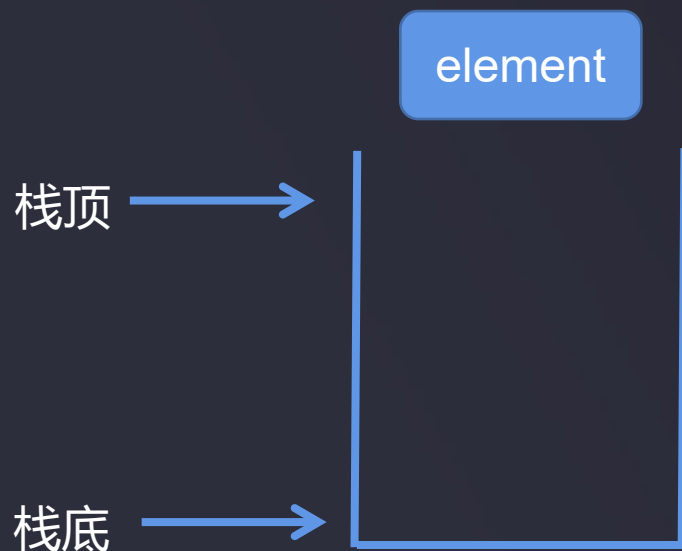
---

- 1 栈
- 2 队列
- 3 复杂度分析

# 栈

## 1 栈的特征

- ◆ 栈 (stack) 又名堆栈, 它是一种运算受限的线性表
- ◆ 限定仅在表尾进行插入和删除操作, 这一端被称为栈顶, 相对地, 把另一端称为栈底
- ◆ 向一个栈插入新元素又称作进栈、入栈或压栈, 它是把新元素放到栈顶元素的上面, 使之成为新的栈顶
- ◆ 元素从一个栈删除元素又称作出栈或退栈, 它是把栈顶元素删除掉, 使其相邻的元素成为新的栈顶元素
- ◆ 栈是一种后进先出的数据结构, Last In First Out(LIFO)



# | 栈

## 2 栈的操作

- ◆ 面向接口编程，用户关心所需要的调用的方法即可
- ◆ 底层有多重实现方式，在本课程中使用Array的方式来实现一个栈
- ◆ 和Array对比，栈的操作就是数组中相应操作的一个约束操作
- ◆ 栈的时间复杂度是 $O(1)$ 级别的

```
public interface Stack<E> {  
    void push(E e);  
    E pop();  
    E peek();  
    int getSize();  
    boolean isEmpty();  
}
```

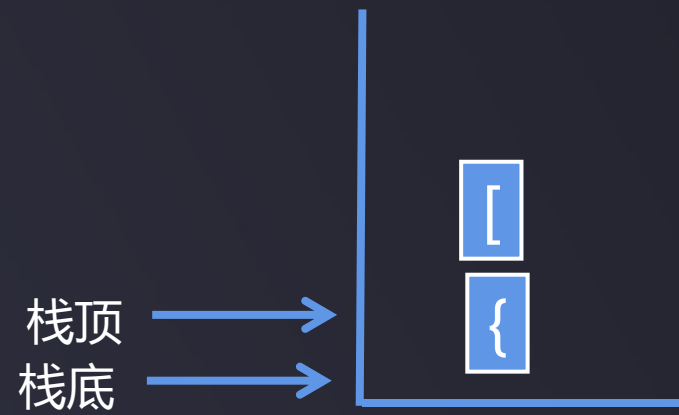
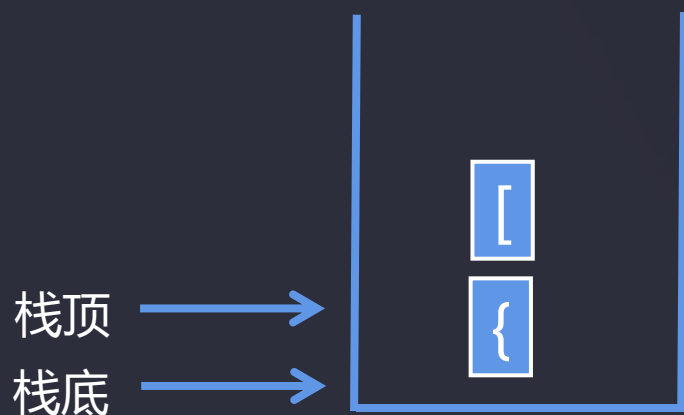
# 栈

## 3 栈的应用

- ◆ 给定一个只包括 '(', ')', '{', '}', '[', ']' 的字符串，判断字符串是否有效
- ◆ 有效字符串需满足：左括号必须用相同类型的右括号闭合。左括号必须以正确的顺序闭合
- ◆ 注意空字符串可被认为是有效字符串
- ◆ 思路：左括号就入栈，右括号就和栈顶匹配，如果匹配成功就出栈

{ [ ] }

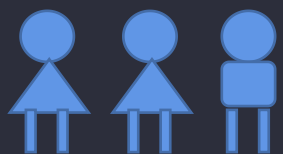
{ [ } ]



# 队列

## 1 队列的特征

- ◆ 队列是一种特殊的线性数据结构
- ◆ 只允许在队首进行删除操作，而在队尾进行插入操作
- ◆ 队列的特点是先进先出First In First Out (FIFO)



队尾

队首

# | 队列

## 2 队列的定义

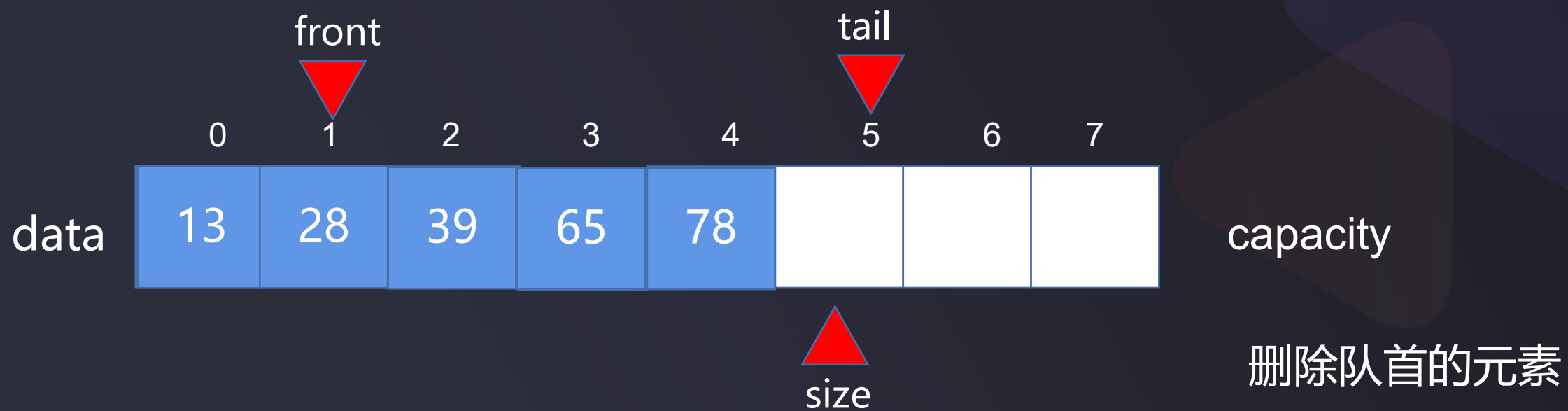
- ◆ 面向接口编程，用户关心所需要的调用的方法即可
- ◆ 底层有多重实现方式，在本课程中使用Array的方式来实现一个队列
- ◆ 和Array对比，队列的操作就是数组中相应操作的一个约束操作

```
public interface Queue<E> {  
    void enqueue(E e);  
    E dequeue();  
    E getFront();  
    int getSize();  
    boolean isEmpty();  
}
```



# 队列

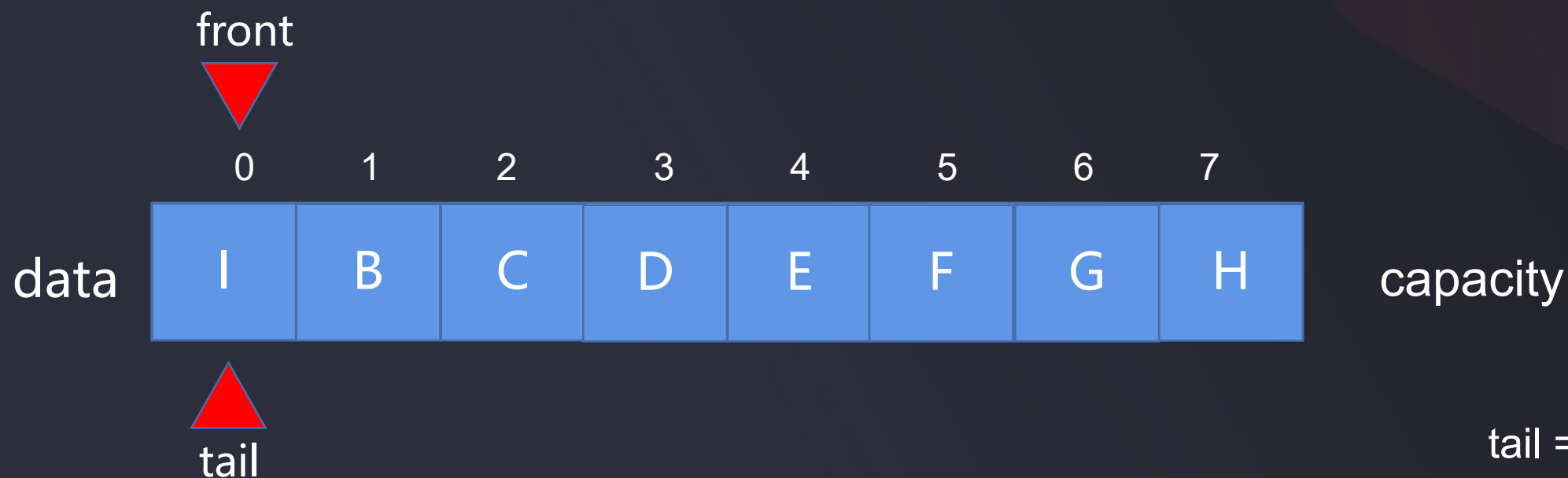
## 3 循环队列



# 队列

## 3 循环队列

- ◆ 循环队列的底层依然是数组，不过增加了指向头和尾的指针了
- ◆  $\text{front} = \text{tail}$  时队列为空
- ◆  $(\text{tail} + 1) \% c == \text{front}$  判断队为满，此时 capacity 预留一个空间不能使用



# | 时间复杂度分析

## 1 队列的复杂度

	ArrayQueue<E>	LoopQueue<E>
void enqueue(E e)	$O(1)$	$O(1)$
E dequeue()	$O(n)$	$O(1)$
E getFront()	$O(1)$	$O(1)$
int getSize()	$O(1)$	$O(1)$
boolean isEmpty()	$O(1)$	$O(1)$

**EDU**

CSDN学院 IT实战派

下节课再见，记得关注公众号

