



CSDN学院 IT实战派

图解数据结构和算法

链表

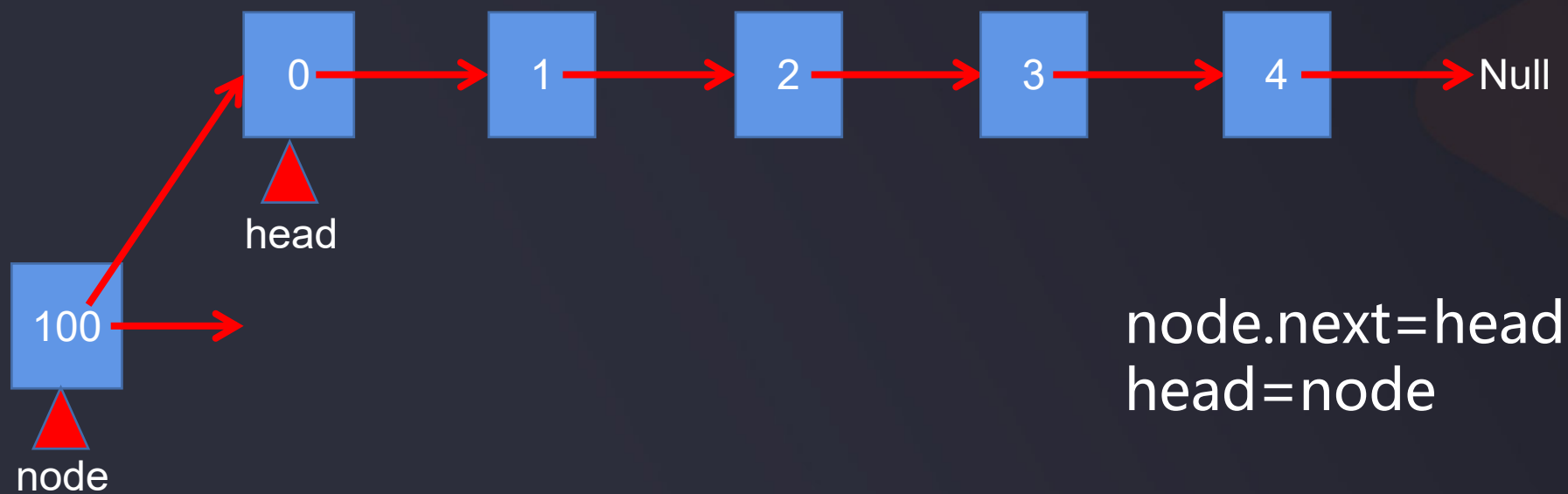
讲师：Samuel

| 本章概述

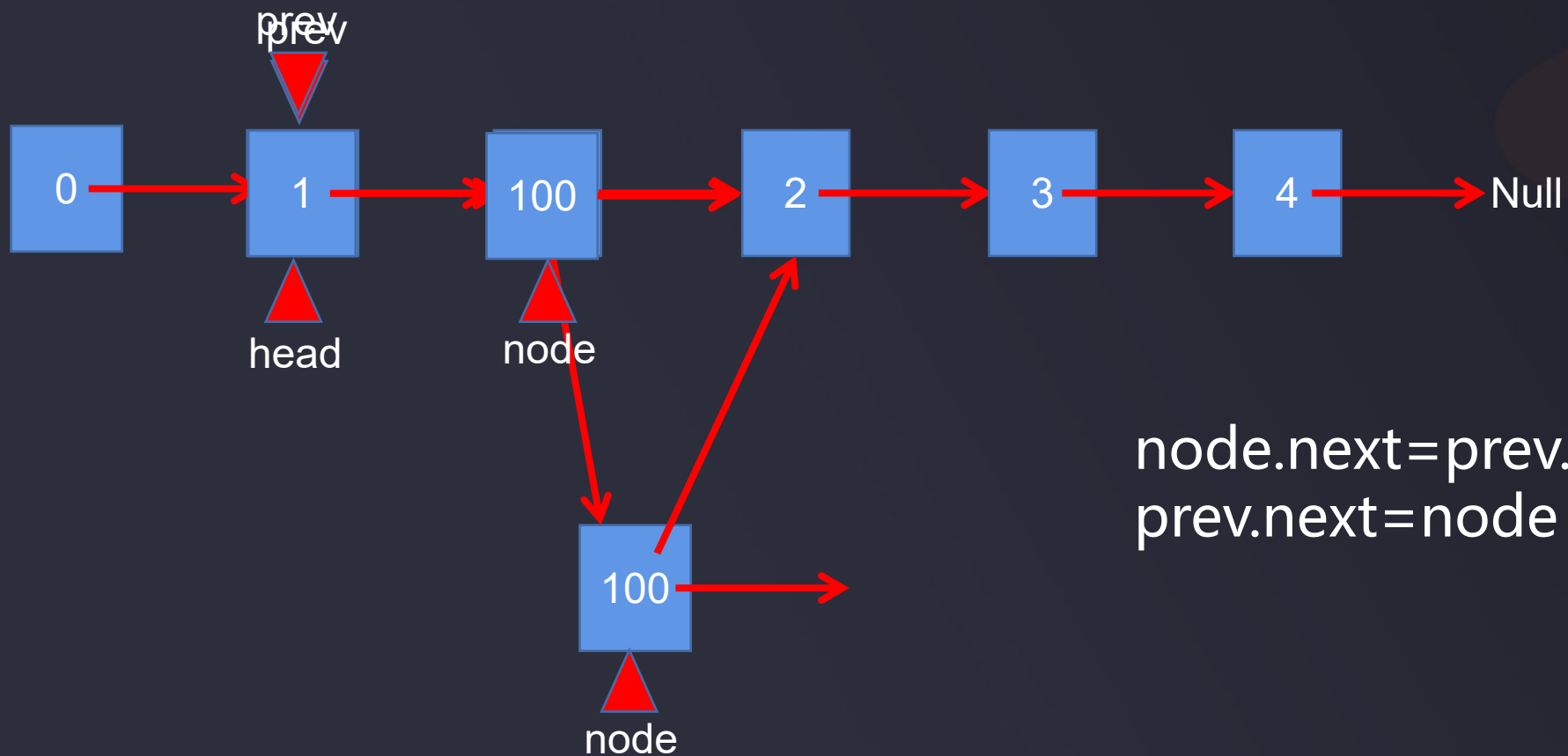
◆ 链表 (Linked list) 是一种真正的动态的数据结构



| 本章概述

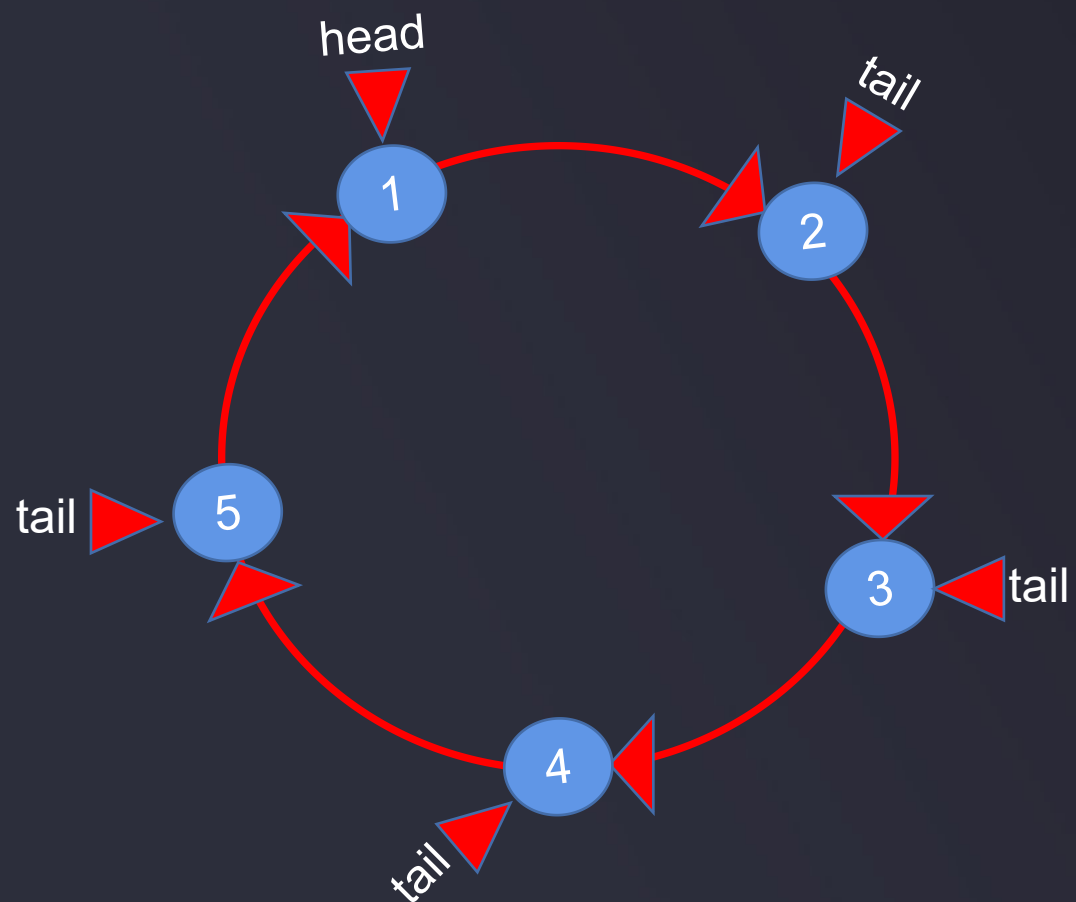


本章概述

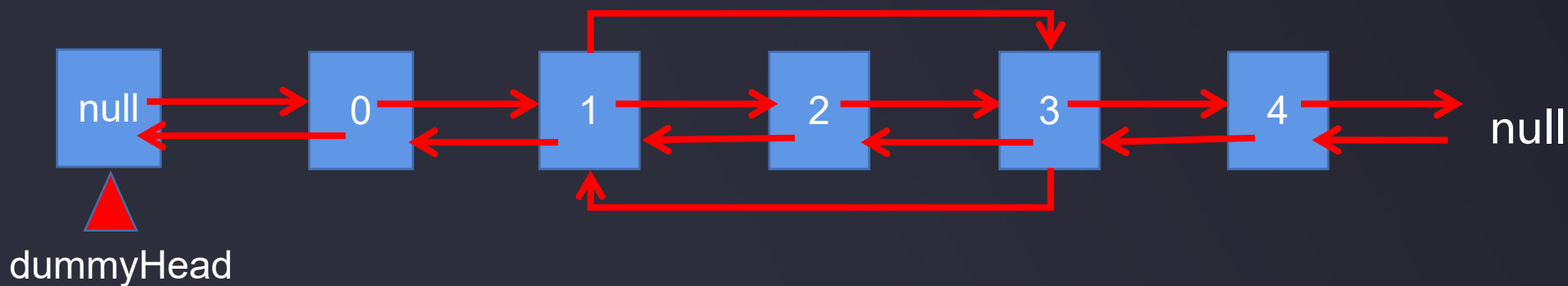
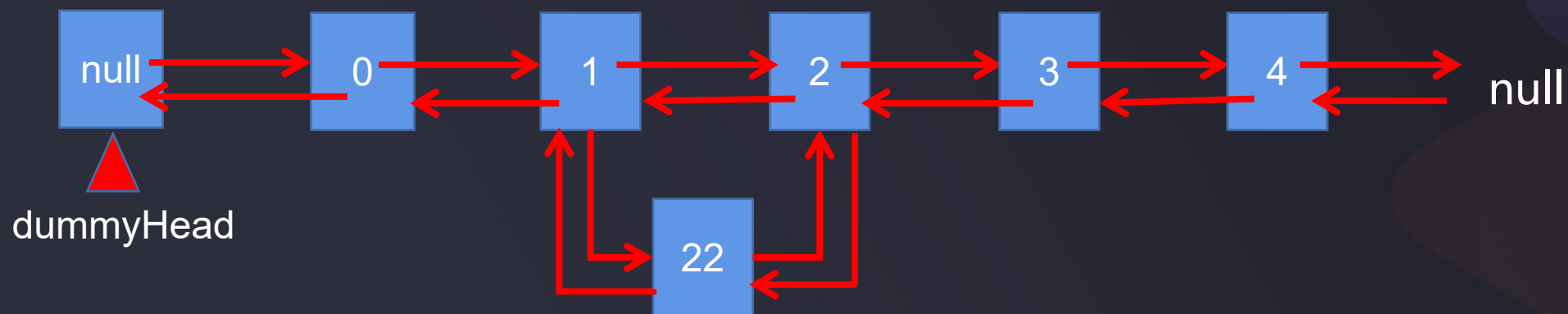


`node.next=prev.next`
`prev.next=node`

| 本章概述



本章概述



| Content

- 1 链表特征和定义
- 2 链表的基本操作
- 3 环形链表和约瑟夫问题

链表

1 链表的特征

- ◆ 链表 (Linked list) 是一种真正的动态的数据结构
- ◆ 链表是一种线性表，但是并不会按线性的顺序存储数据，而是在每一个节点里存到下一个节点的指针
- ◆ 使用链表结构可以克服数组需要预先知道数据大小的缺点，但增加了结点的指针域，空间开销比较大
- ◆ 链表允许插入和移除链表上任意位置上的节点，但是不允许随机存取
- ◆ 链表有很多种不同的类型：单向链表，双向链表以及循环链表



| 链表

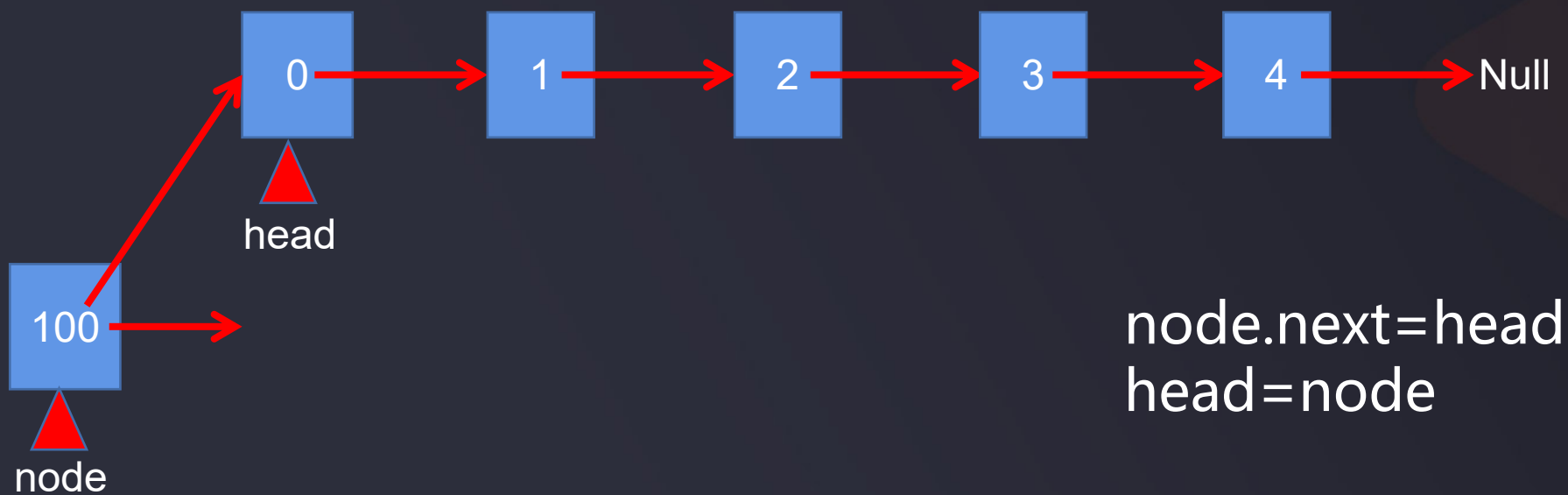
2 单链表的定义

- ◆ 链表中的数据是以结点来表示的
- ◆ 每个结点的构成：元素 + 指针
- ◆ 元素就是存储数据的存储单元，指针就是连接每个结点的地址数据
- ◆ 以“结点的序列”表示线性表称作线性链表（单链表），单链表是链式存取的结构

```
public class SingleLinkedList<E> {  
    private class Node{  
        public E e;  
        public Node next;  
    }  
}
```

链表

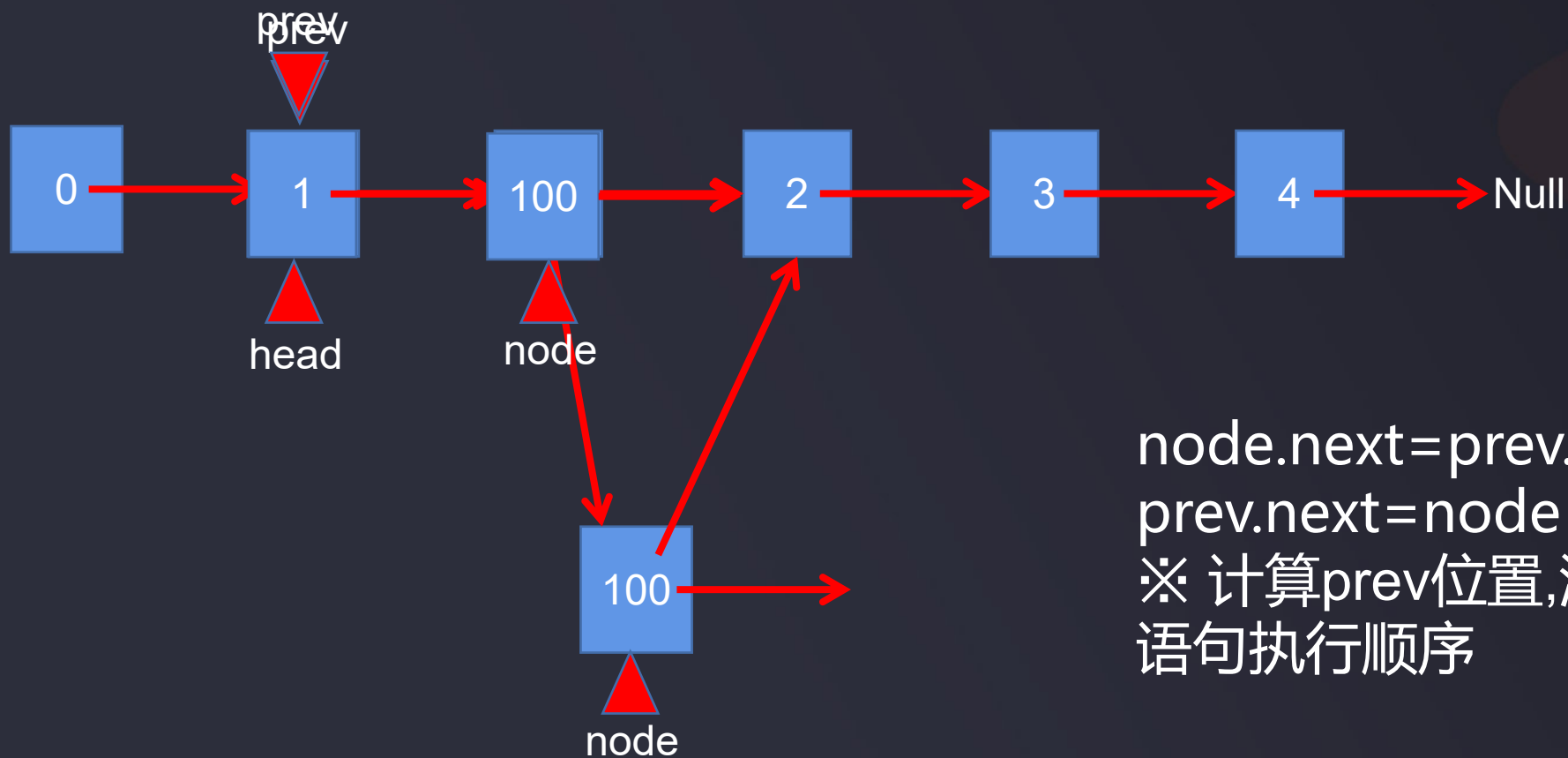
3 单链表的头添加元素



链表

4 单链表的中间添加元素

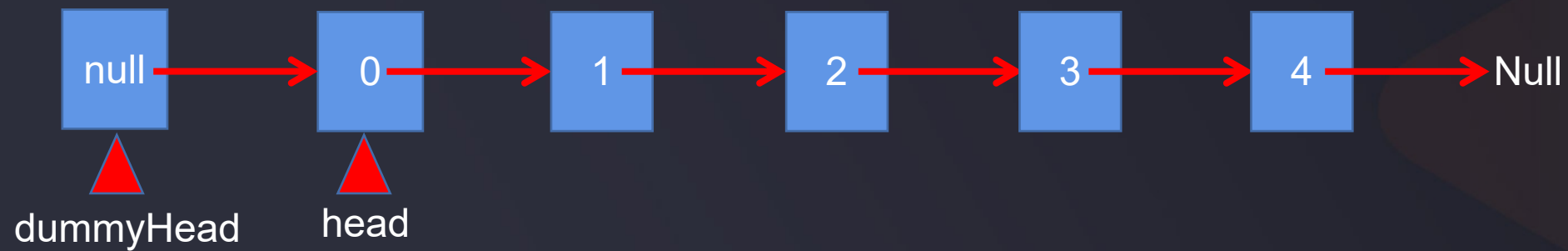
在索引2的位置插入100



`node.next=prev.next`
`prev.next=node`
※ 计算prev位置,注意索引为0的情况
语句执行顺序

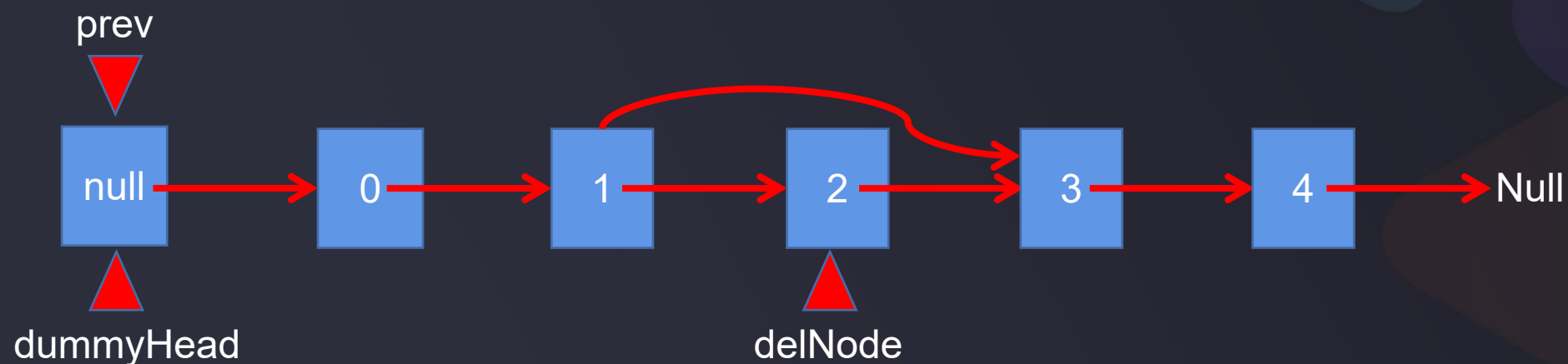
链表

5 单链表的虚拟头结点



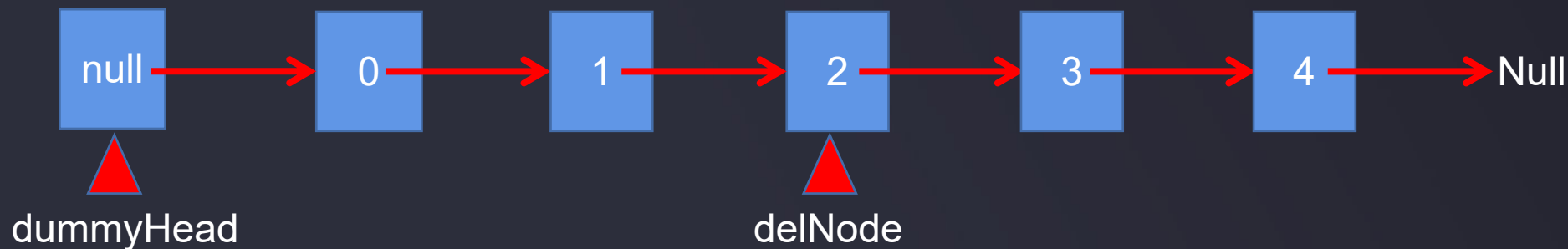
链表

6 单链表的删除节点



删除索引为2的节点

$\text{prev.next} = \text{delNode.next}$
 $\text{delNode} = \text{null}$
※ 不能使用 $\text{delNode} = \text{delNode.next}$



链表

7 单链表的时间复杂度

增

$O(n)$

删

$O(n)$

查

$O(n)$

改

$O(n)$

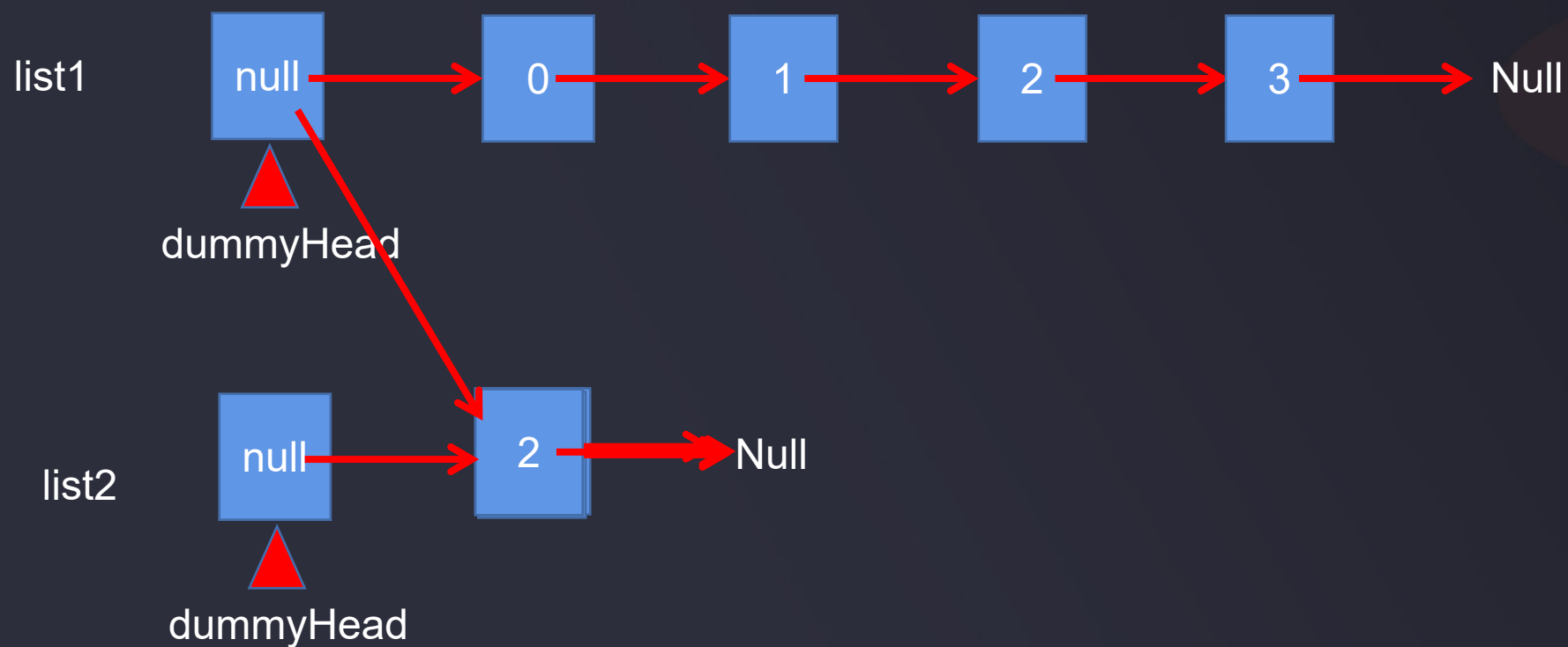
} 如果只对链表头进行操作是 $O(1)$

→ 只查链表头的元素(1)

链表

8 链表的应用

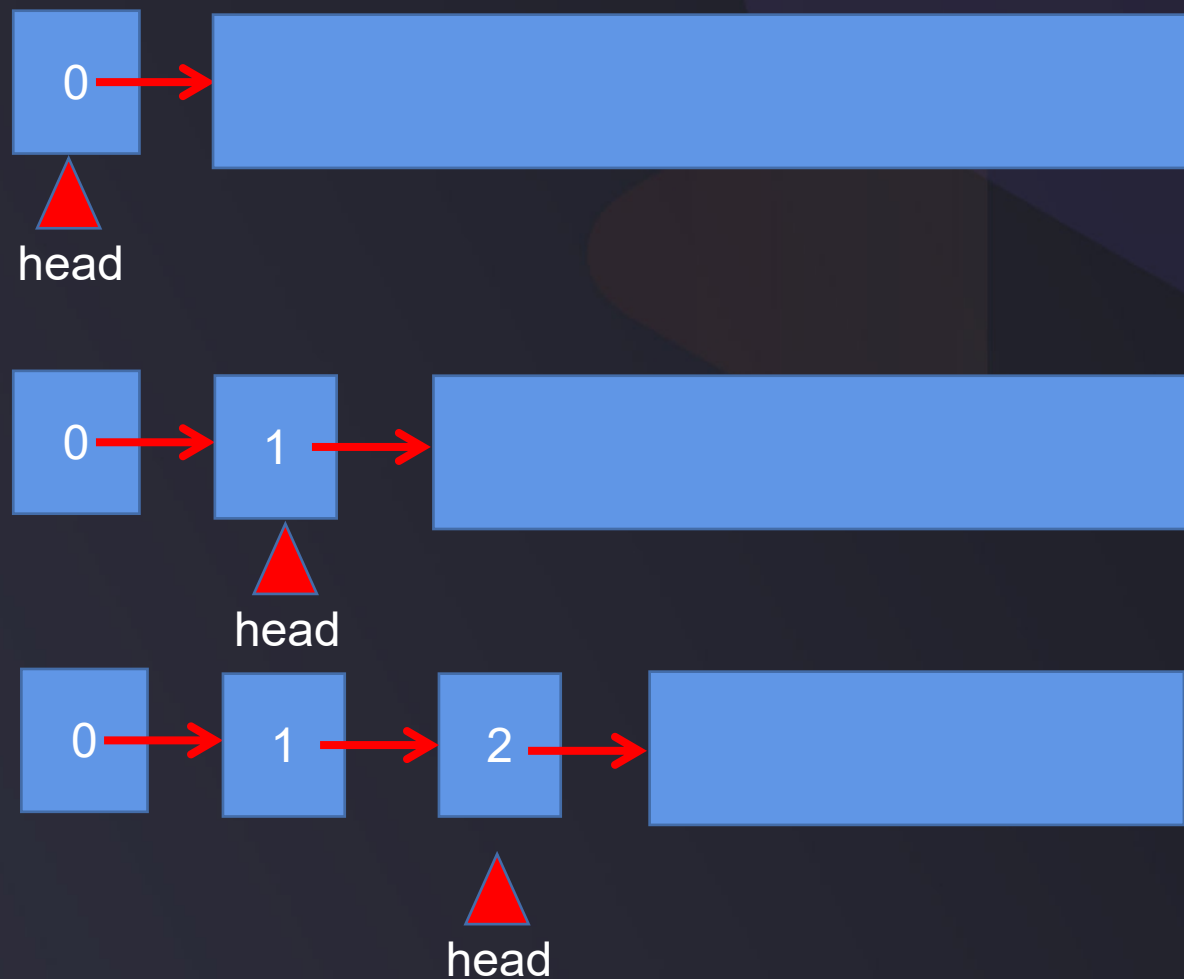
◆ 翻转单链表----腾讯



链表

9 链表和递归

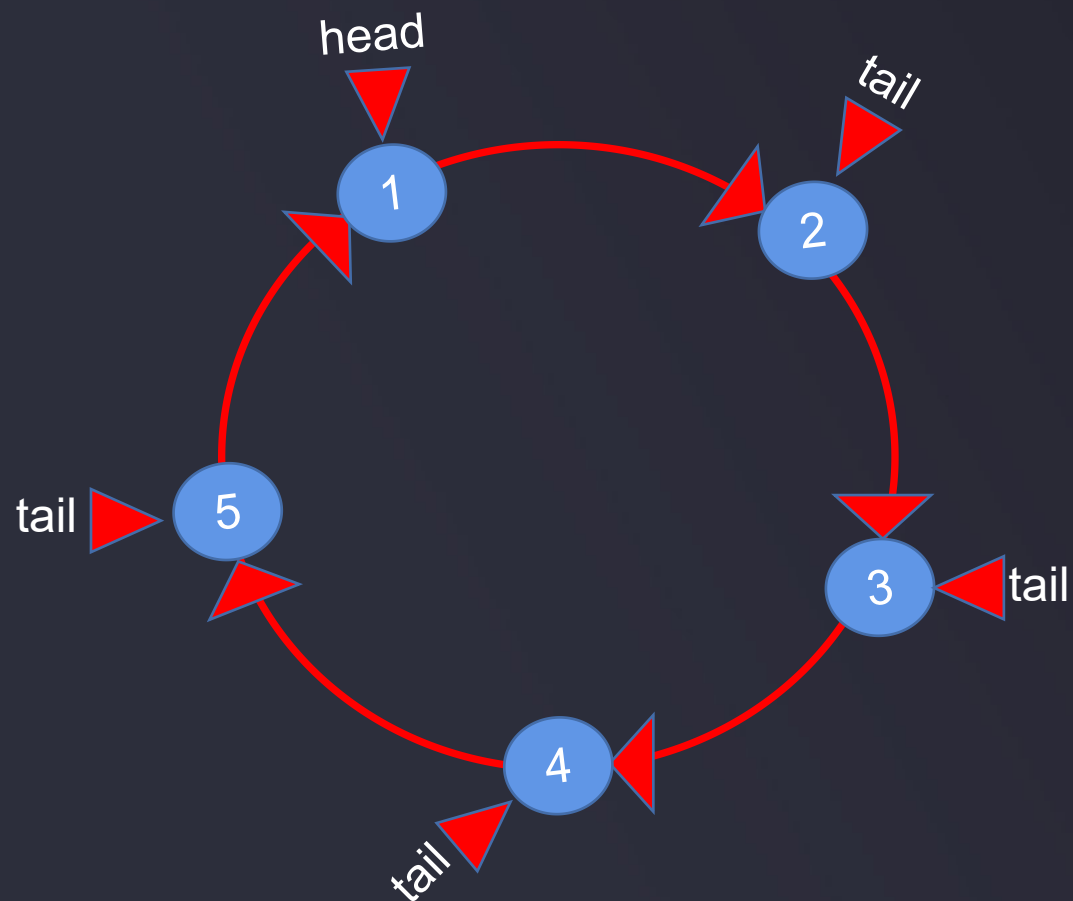
- ◆ 删除链表中等于给定值 val 的所有节点
- ◆ 输入: $1 \rightarrow 2 \rightarrow 6 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$, $val = 6$
- ◆ 输出: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$
- ◆ 解题方式一: 使用while循环
- ◆ 解题方式二: 递归



链表

10 环形链表

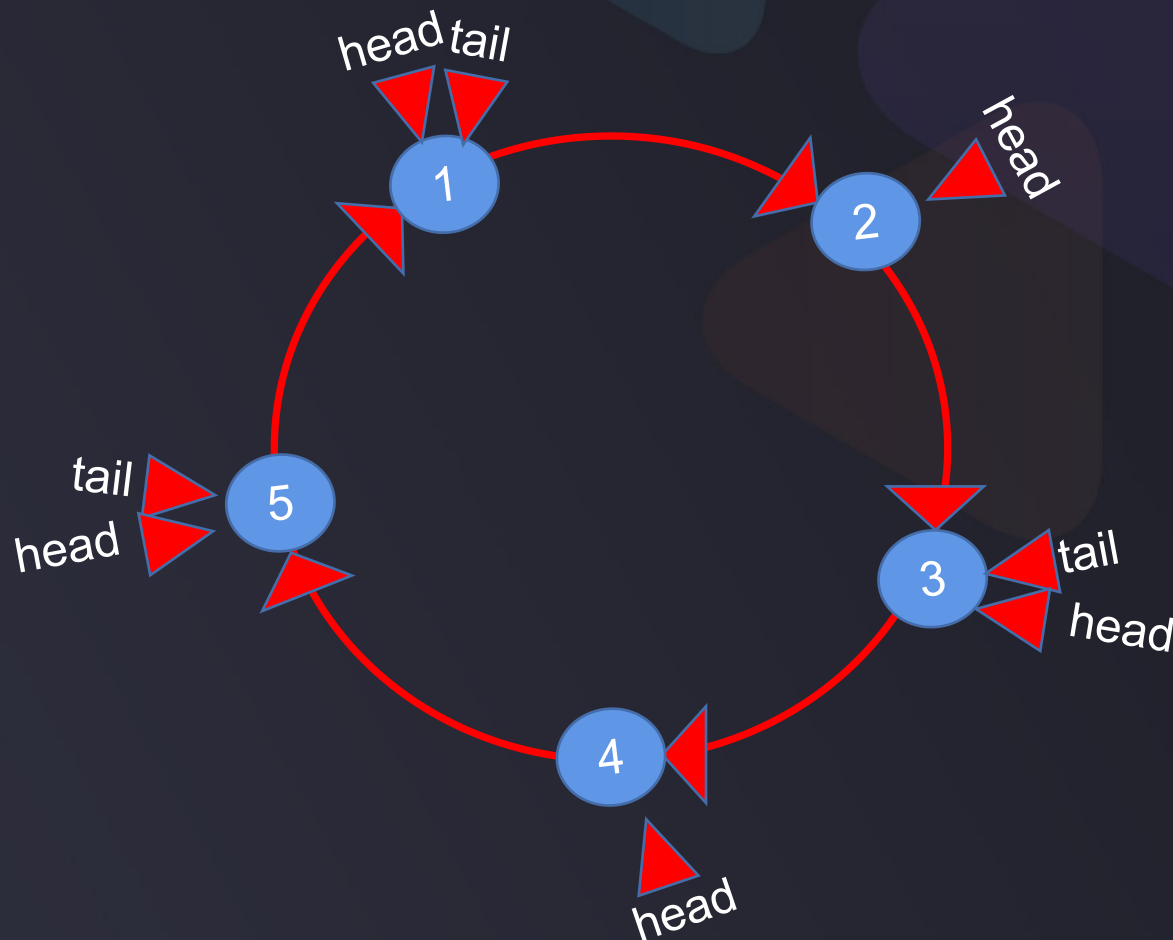
◆ 将单链表最后一个节点指向头节点，那么这个单链表就是一个环形链表



链表

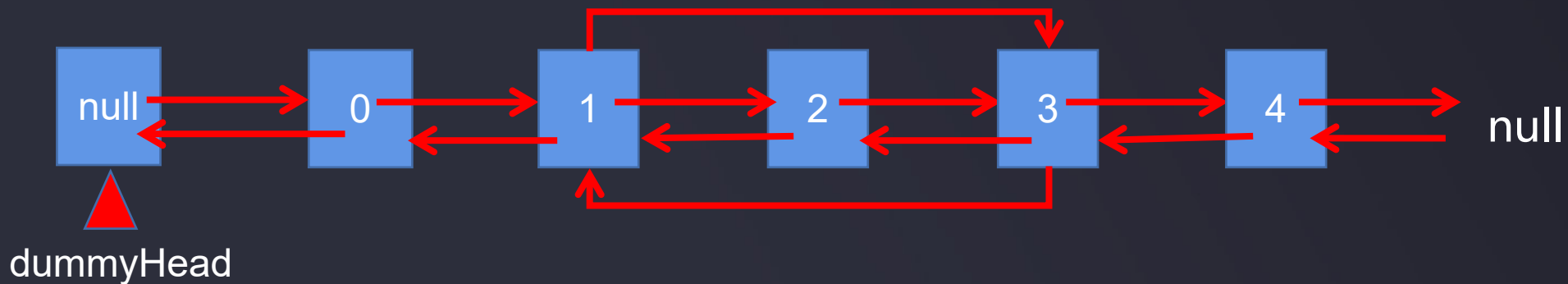
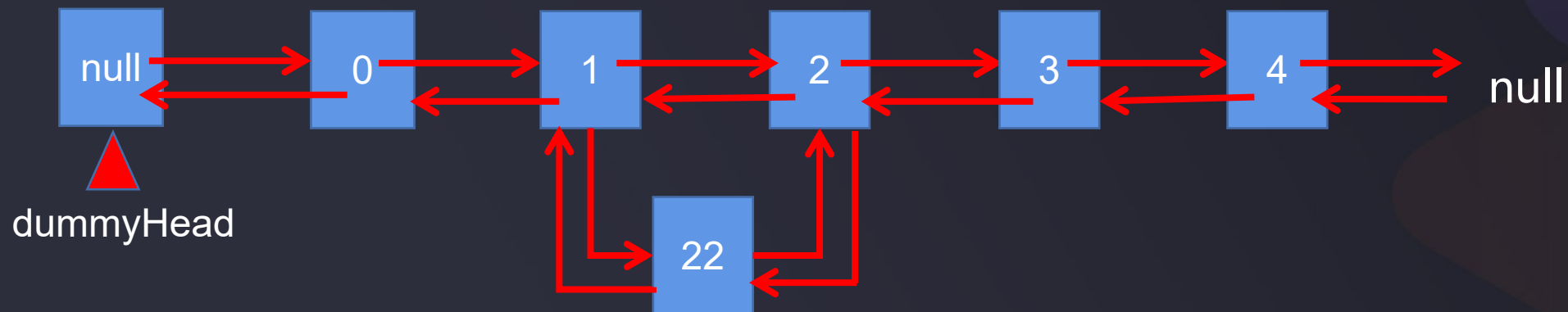
10 约瑟夫问题

- ◆ 一堆猴子都有编号，编号是1, 2, 3 ...n
这n个猴子按照1-n的顺序围坐一圈
- ◆ 从第k个开始数，每数到第m个,该猴子就要离开此圈
- ◆ 下一次从k+1个开始，直到圈中只剩下最后一只猴子，则该猴子为大王
- ◆ 2->4->1->5->3



链表

11 双链表



EDU

CSDN学院 IT实战派

下节课再见，记得关注公众号

