

# Data Structure & Algorithm

## Data Structure

### Singly Linked List (part 1)

# 1. Nội dung

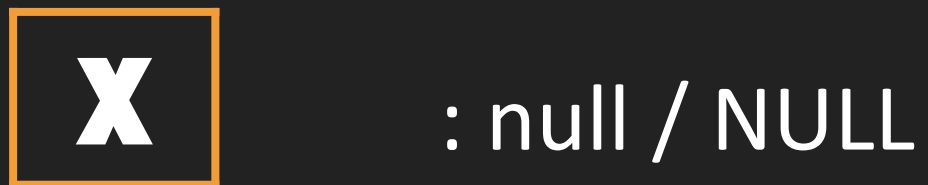
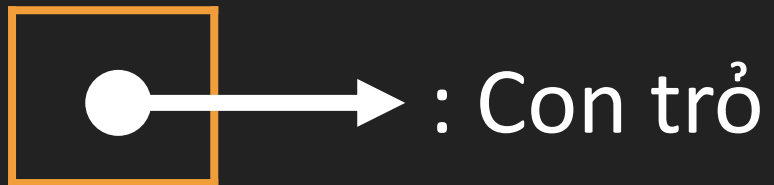
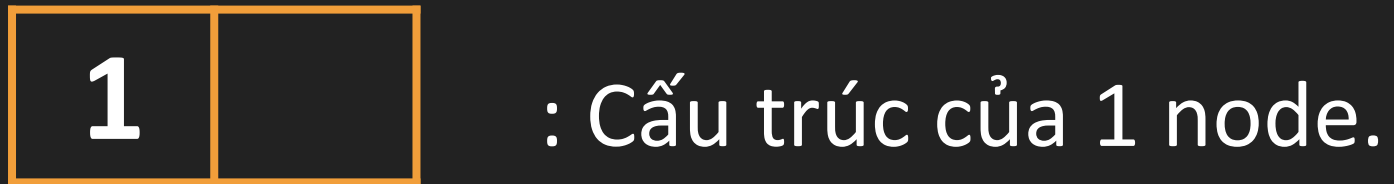
1. Định nghĩa Singly Linked List.
2. Ưu nhược điểm của Linked List.
3. Thao tác cơ bản: Thêm, Xoá.

# 1. Yêu cầu

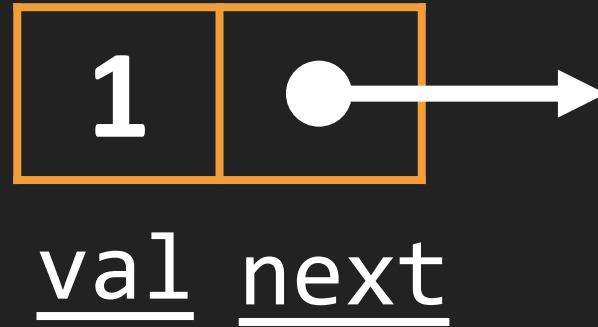
**1.** Biết sử dụng **struct** / **class** trong **C/C++** hoặc **class** trong **Java**.

**2.** Phân biệt được **reference type** và **value/primitive type** trong **Java** hoặc biết cách sử dụng **con trỏ** trong **C/C++**.

## 2. Định nghĩa



## 2. Định nghĩa



### C++

```
struct SinglyListNode
{
    int val;
    SinglyListNode *next;
    SinglyListNode(int x) : val(x),
                           next(NULL) {}
};
```

### Java

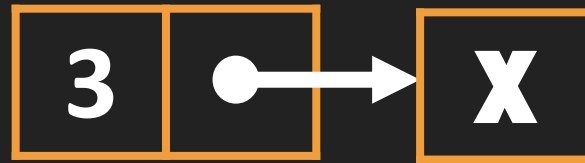
```
public class SinglyListNode
{
    int val;
    SinglyListNode next;
    SinglyListNode(int x) { val = x; }
}
```

### 3. Đặc điểm của Singly Linked List



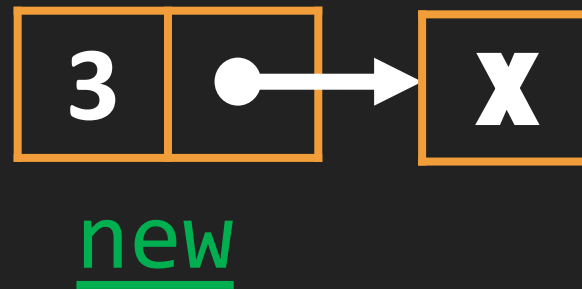
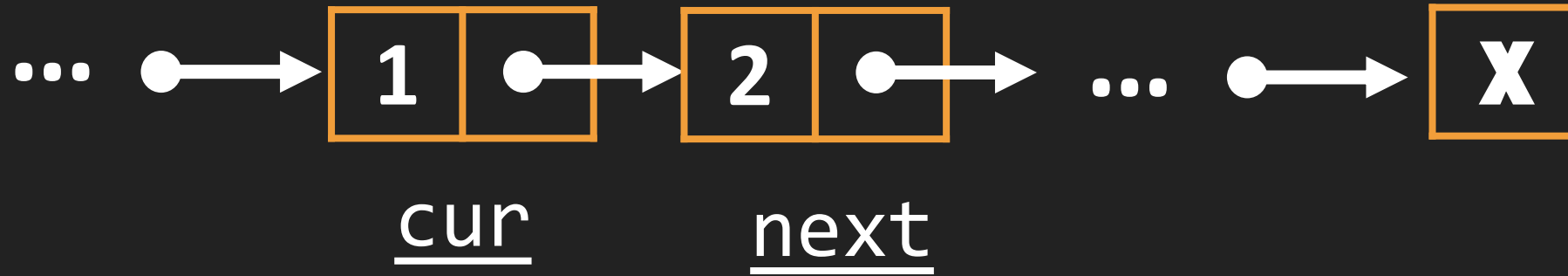
1. Chỉ cần lưu node đầu tiên (**head**) là có thể truy xuất cả mảng.
2. Khả năng truy xuất 1 phần tử của mảng:  $O(n)$
3. Khả năng thêm và xoá 1 phần tử :  $O(1)^*$   
(\*: Vị trí đó phải biết trước)

## 4. Thao tác thêm phần tử vào giữa list.



new

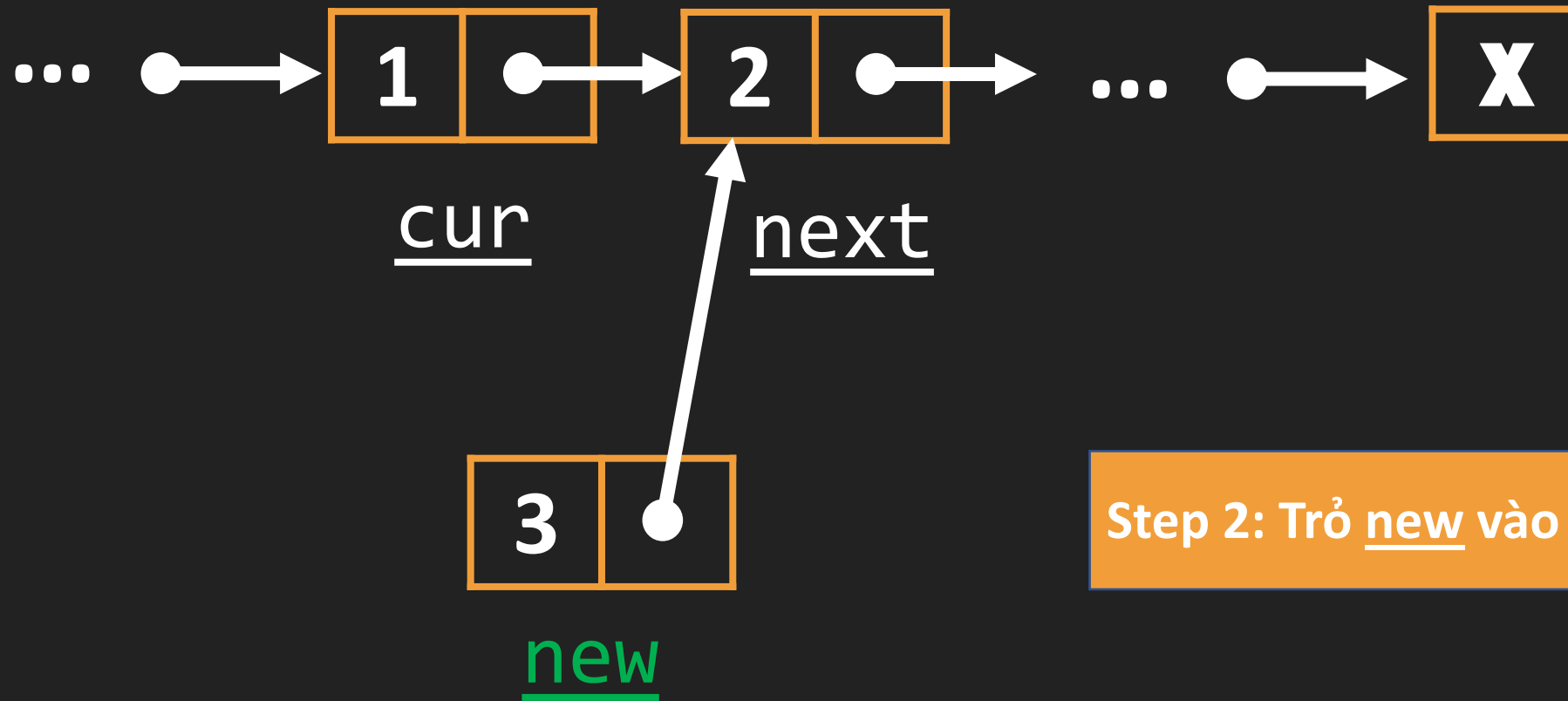
## 4. Thao tác thêm phần tử vào giữa list.



Step 1: Tìm cur và next.

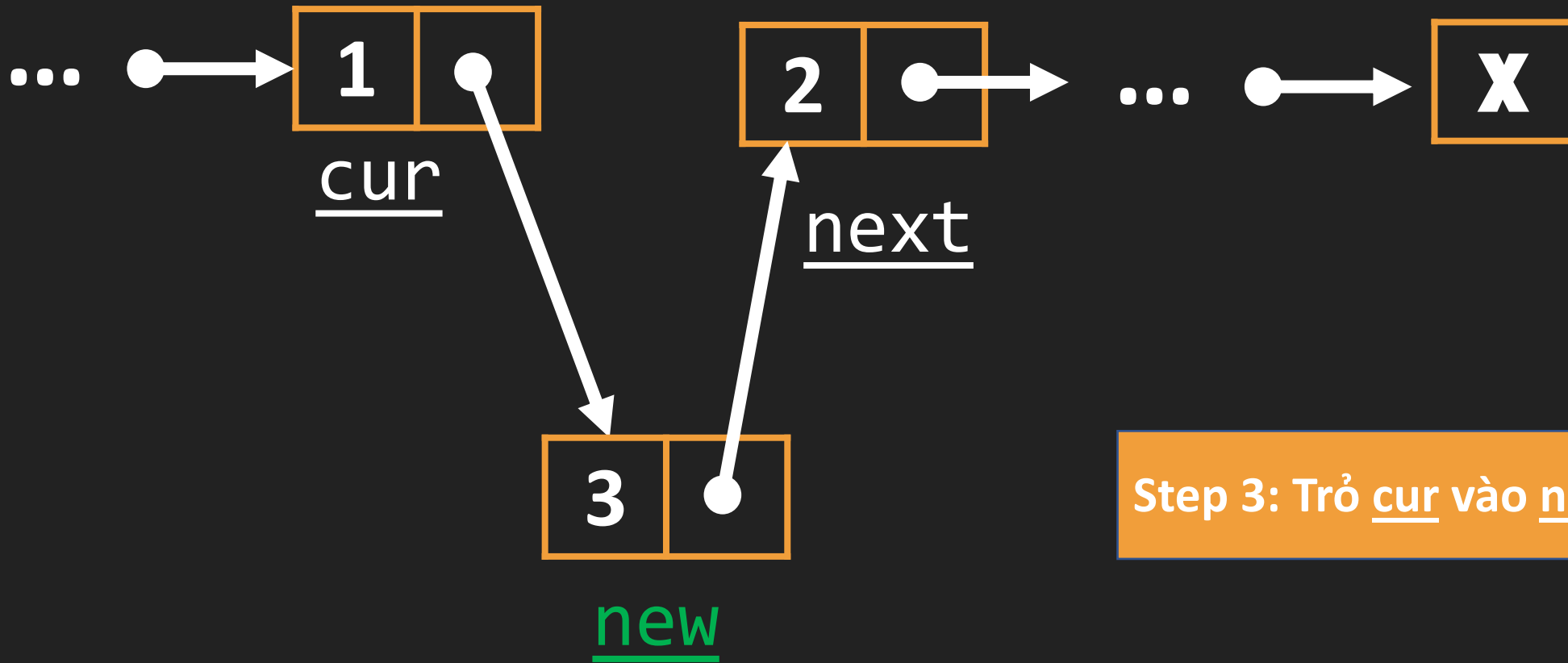


## 4. Thao tác thêm phần tử vào giữa list.



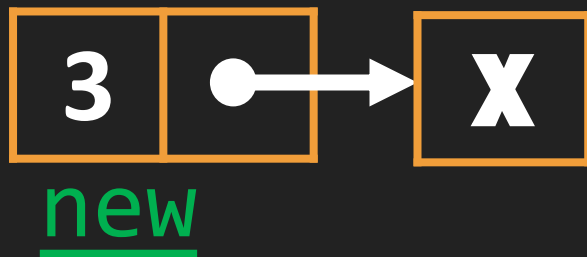
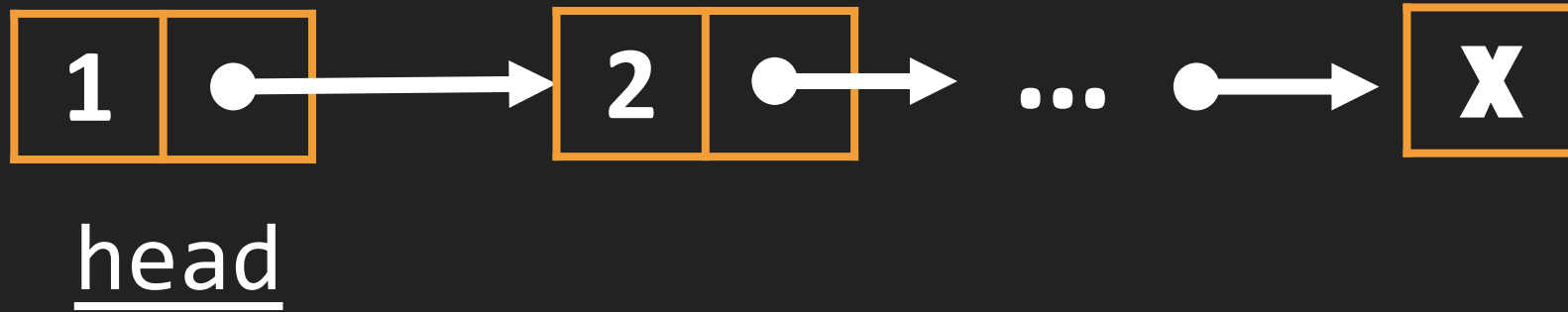
Step 2: Trỏ new vào next.

## 4. Thao tác thêm phần tử vào giữa list.

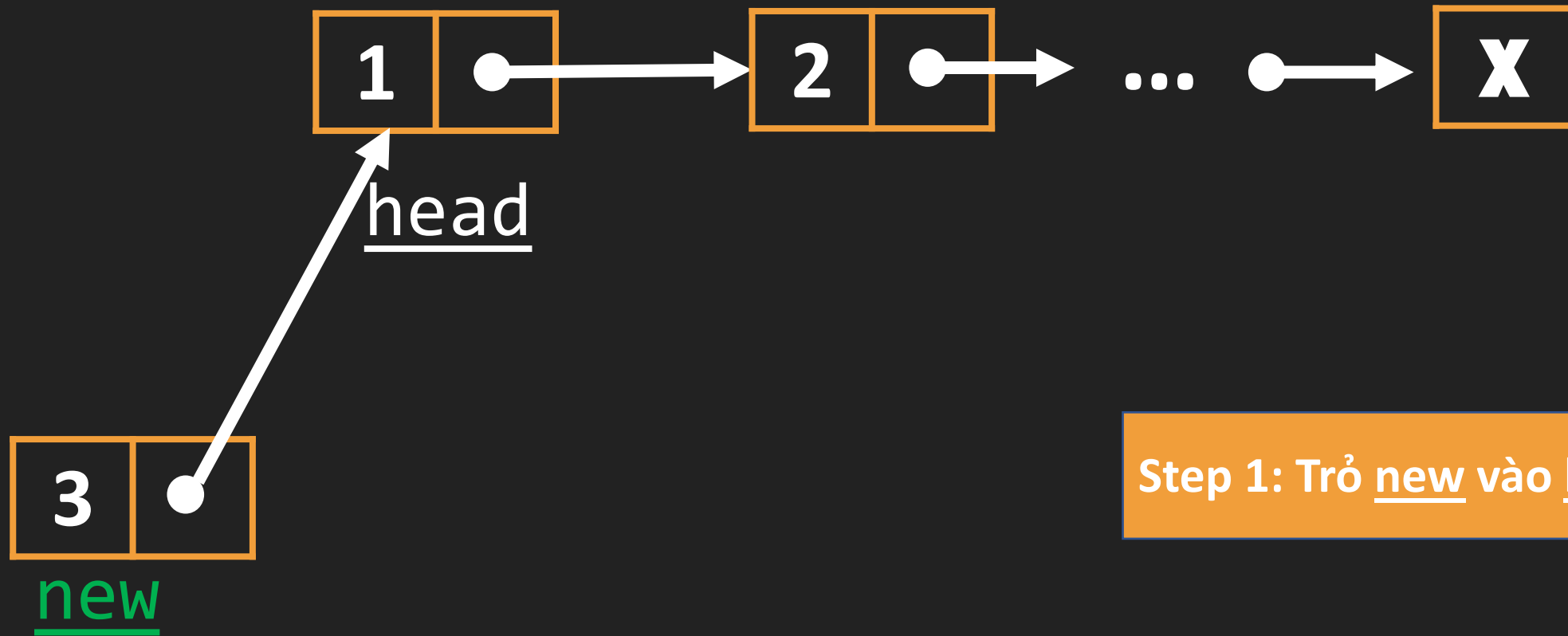


Step 3: Trỏ cur vào new.

## 5. Thao tác thêm phần tử vào đầu list.

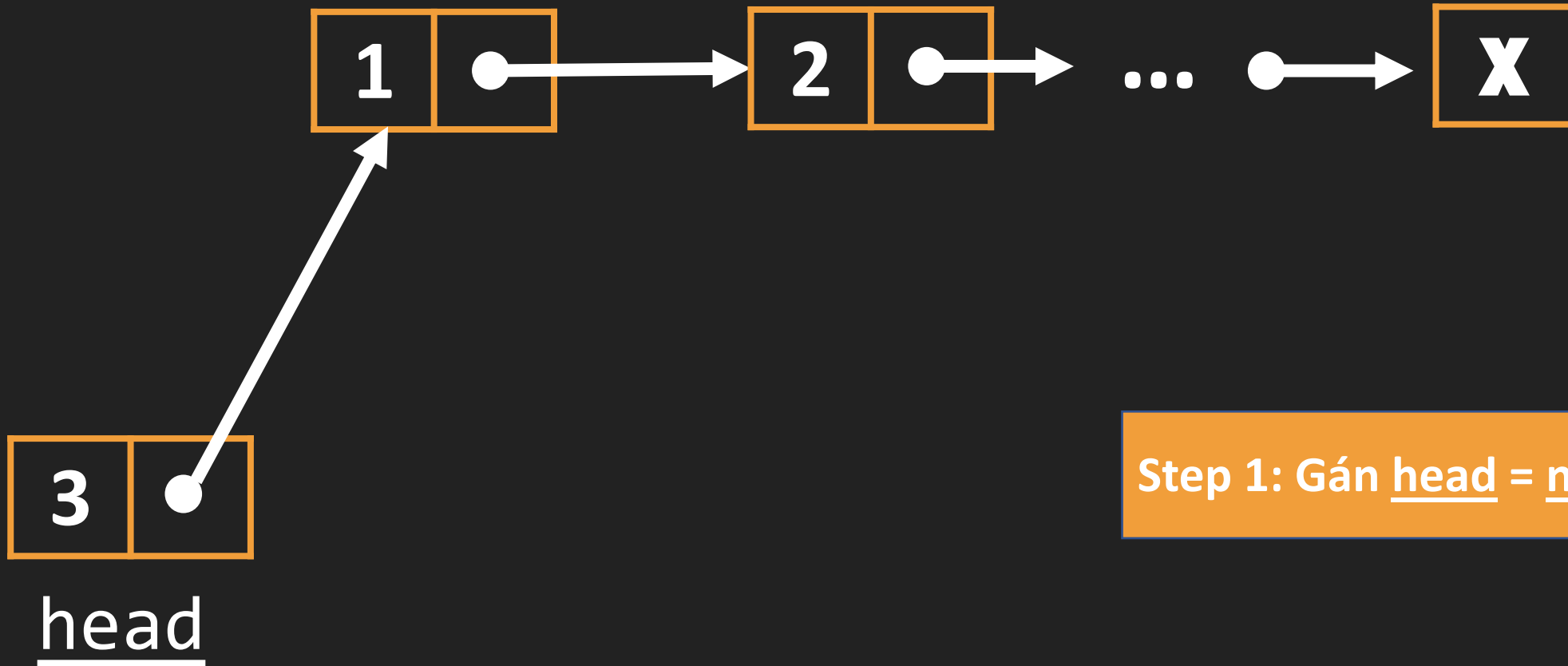


## 5. Thao tác thêm phần tử vào đầu list.



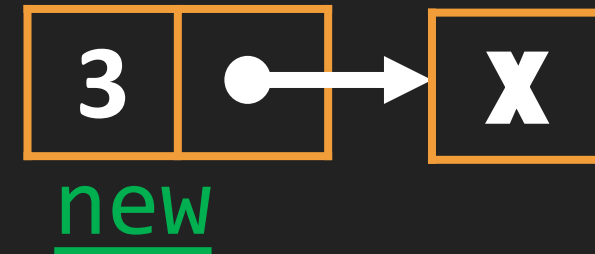
Step 1: Trỏ new vào head.

## 5. Thao tác thêm phần tử vào đầu list.

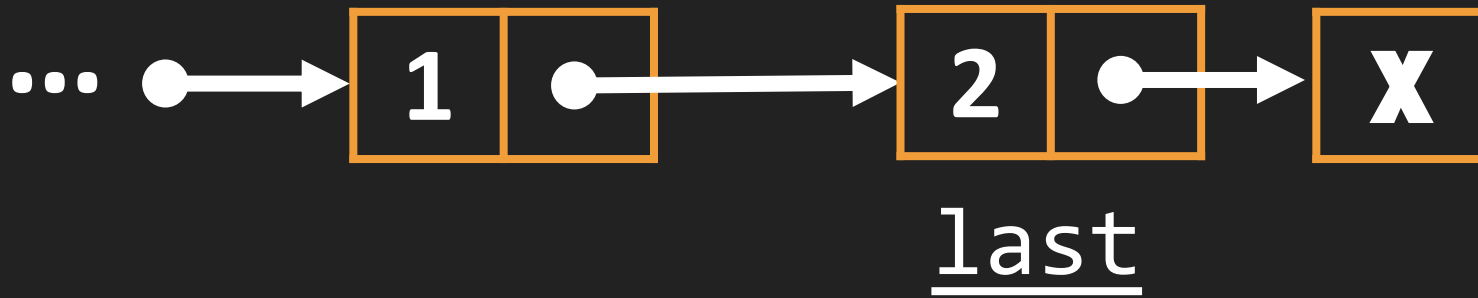


Step 1: Gán head = new.

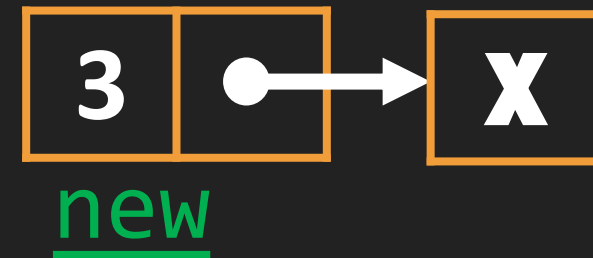
## 6. Thao tác thêm phần tử vào cuối list.



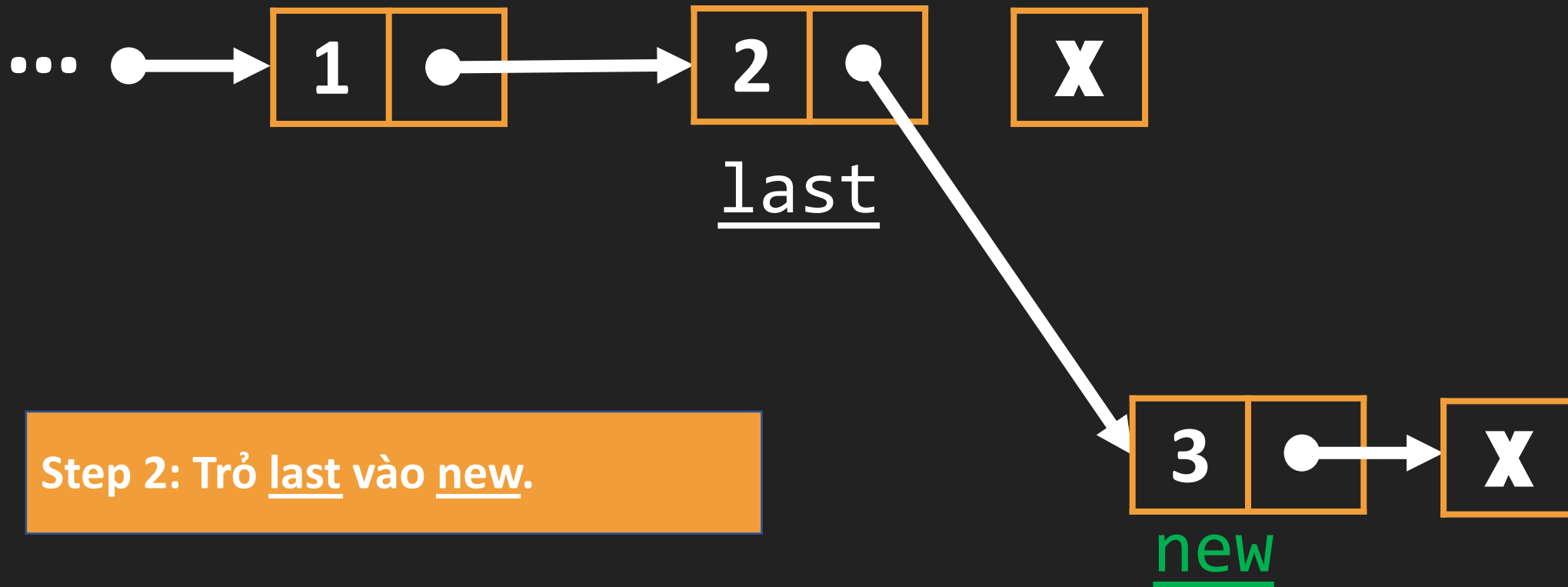
## 6. Thao tác thêm phần tử vào cuối list.



Step 1: Xác định last.

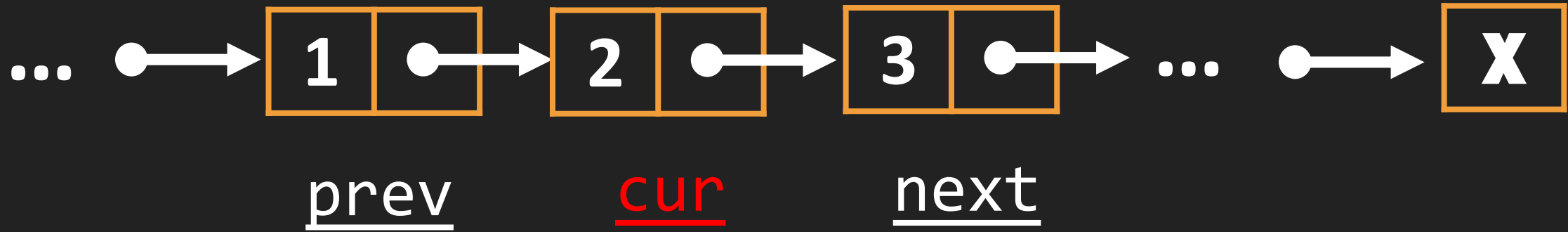


## 6. Thao tác thêm phần tử vào cuối list.

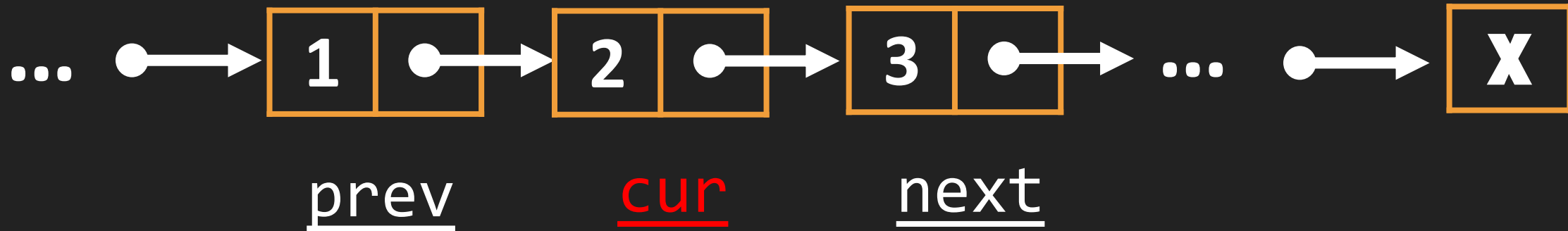




## 7. Xoá phần tử ở giữa list.

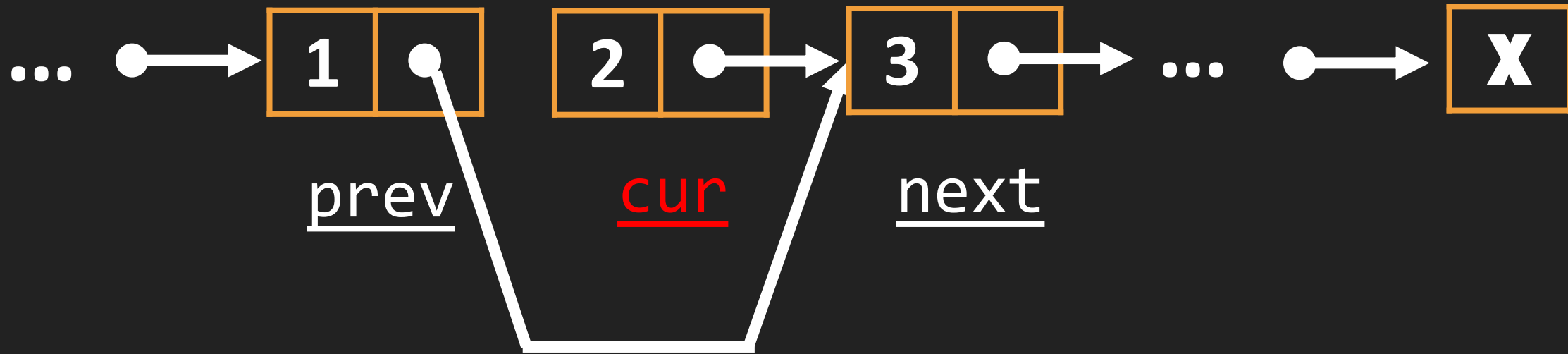


## 7. Xoá phần tử ở giữa list.



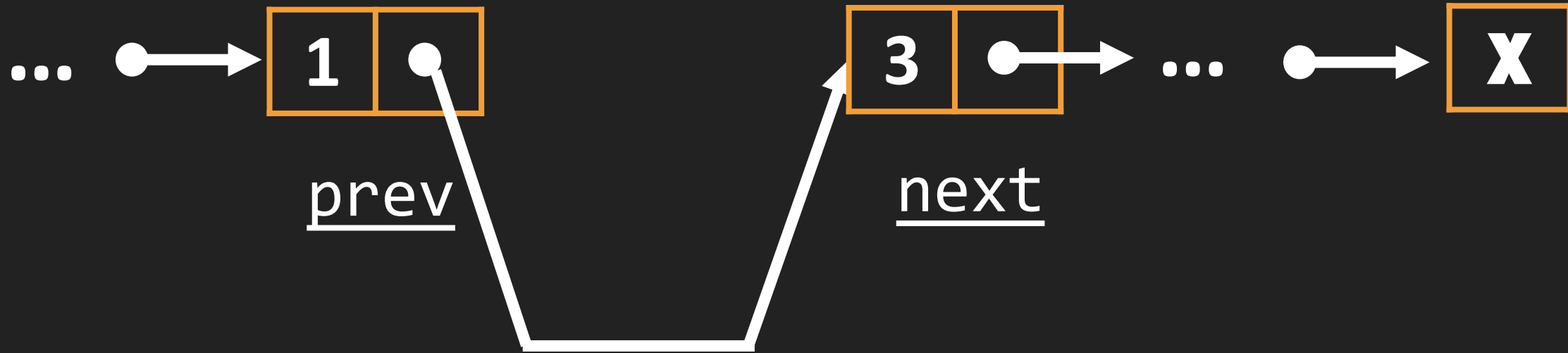
Step 1: Xác định prev, cur và next.

## 7. Xoá phần tử ở giữa list.



Step 2: Trỏ prev vào next.

## 7. Xoá phần tử ở giữa list.



Step 3: Giải phóng vùng nhớ dành cho cur. (C/C++)

## 8. Xoá phần tử ở đầu list.



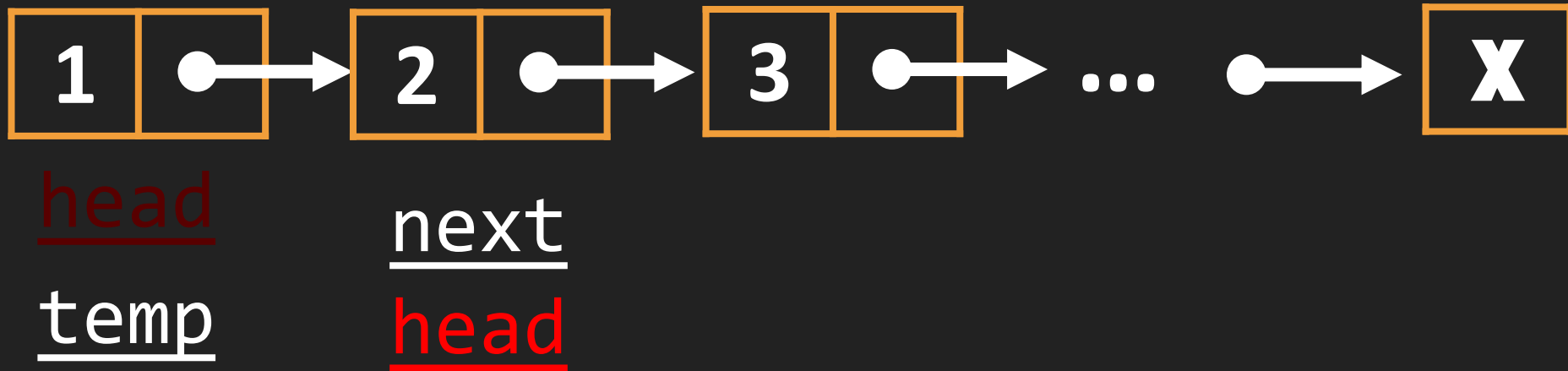
head

## 8. Xoá phần tử ở đầu list.



Step 1: Xác định next.

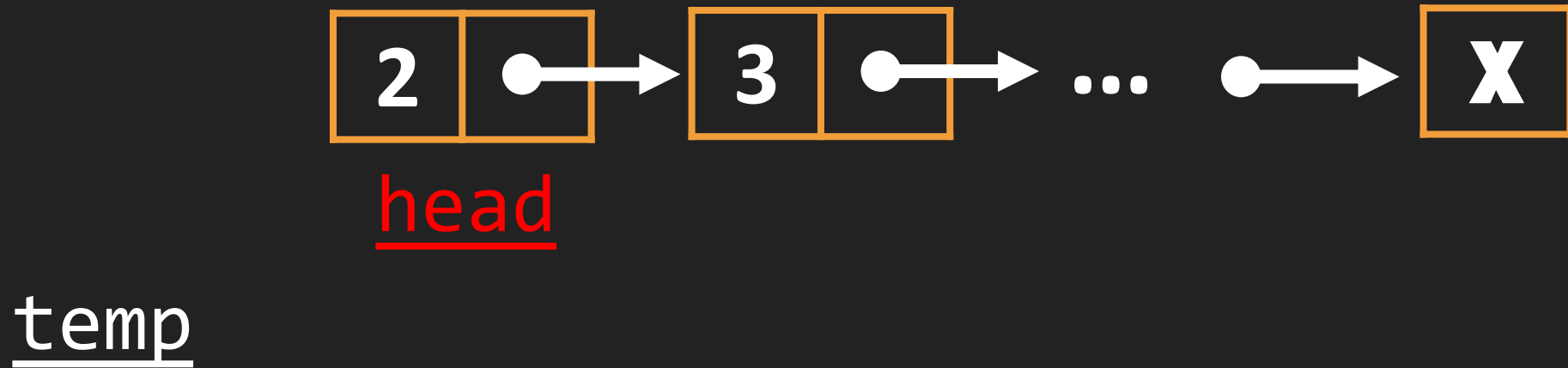
## 8. Xoá phần tử ở đầu list.



Step 2:

- Trỏ temp vào head. (C/C++).
- Trỏ head vào next.

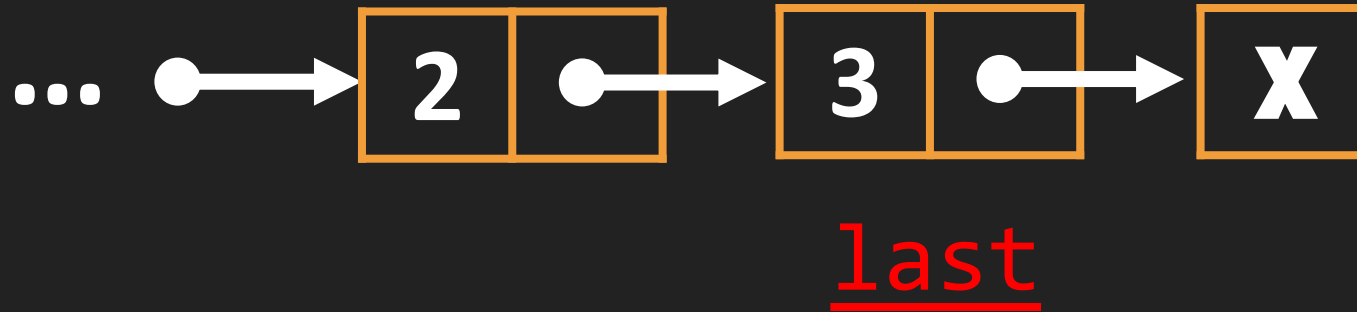
## 8. Xoá phần tử ở đầu list.



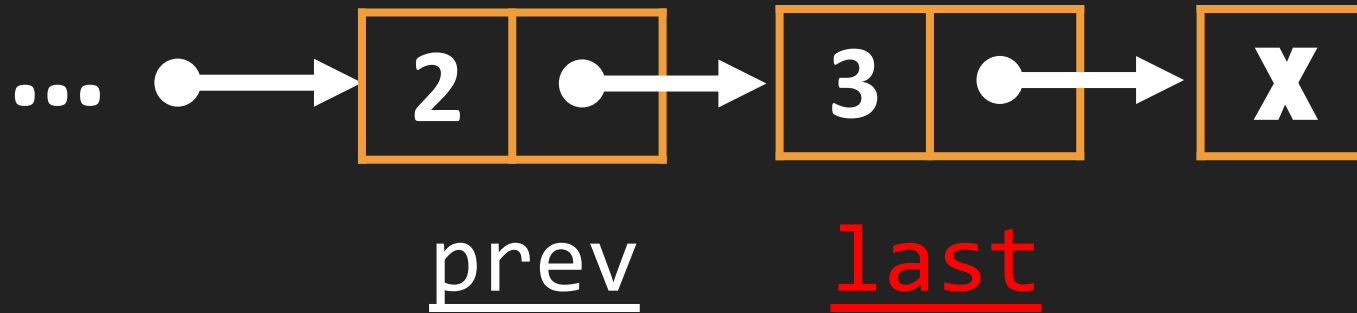
Step 3: Giải phóng vùng nhớ của temp. (C/C++)



## 8. Xoá phần tử ở cuối list.

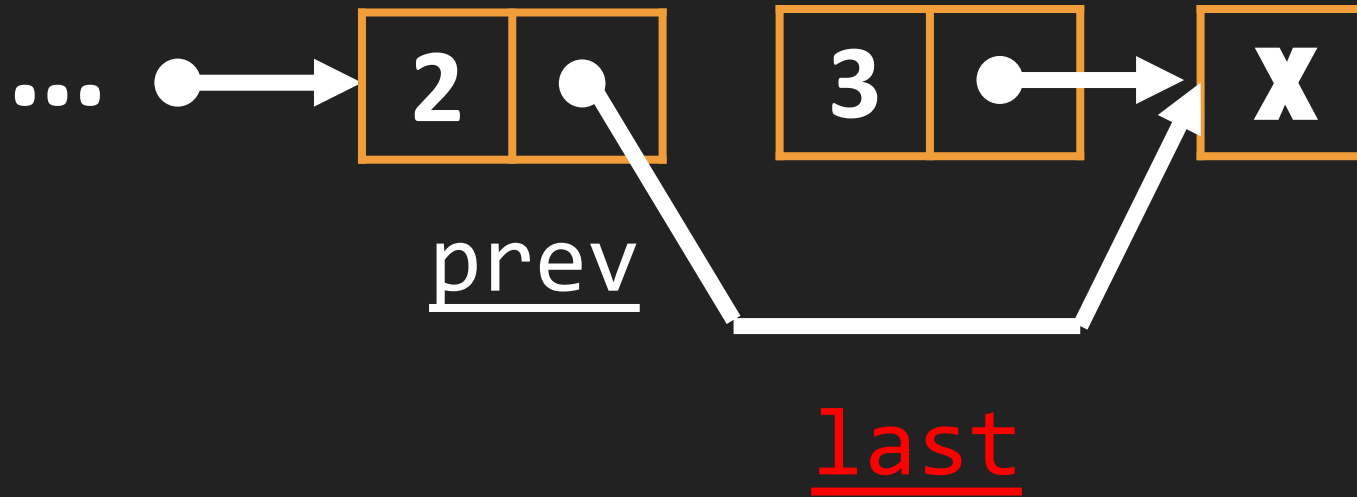


## 8. Xoá phần tử ở cuối list.



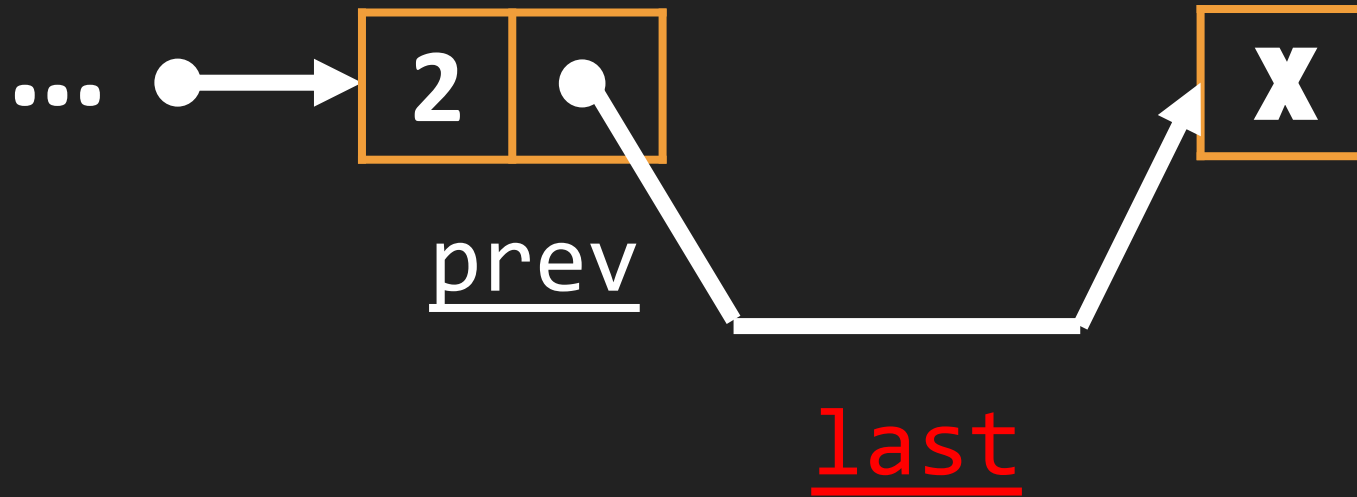
Step 1: Xác định prev và last

## 8. Xoá phần tử ở cuối list.



Step 2: Trỏ prev về NULL

## 8. Xoá phần tử ở cuối list.



Step 3: Giải phóng vùng nhớ cho last. (C/C++)

# 9. Implement Singly Linked List.



Daily **LeetCode** Challenge

16

**707.** Design Linked List

Tags: #linkedlist

- The Brown Box -

[hoangvancong.com](https://hoangvancong.com)

# Data Structure & Algorithm

