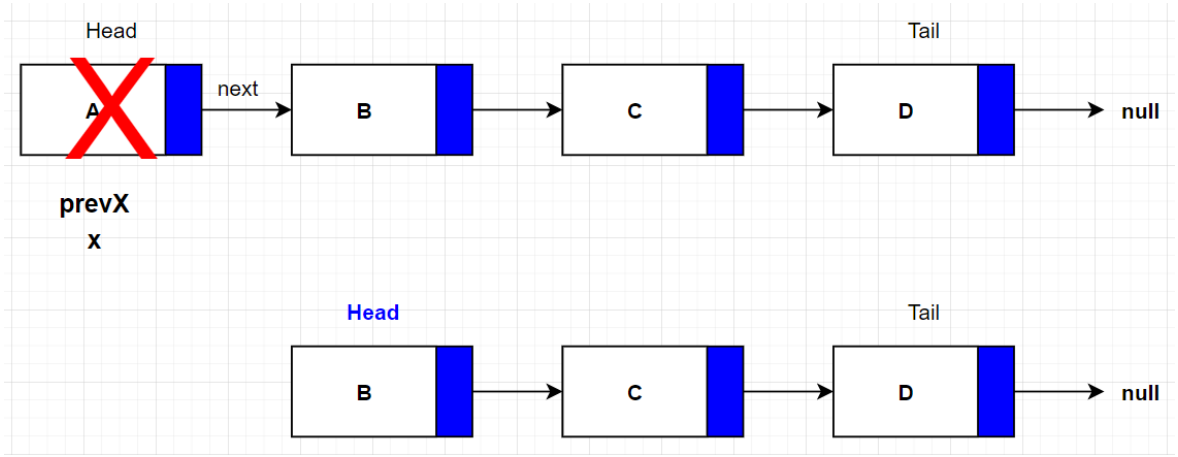
Bài 3.5: Xóa một node khỏi DSLK đơn

- ✓ Các bước thực hiện
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Các bước thực hiện

- Gọi node cần xóa là x, node liền trước x là prevX:
- Bước 1. Tìm node x và prevX.
- Bước 2. Xóa node x nếu x khác null:
  - B2.1. Nếu x là node head:
    - Lưu lại node head vào biến r;
    - Gán head = x->next;
    - Xóa node r;
  - B2.2. Nếu x là node tail:
    - Lưu lại node tail vào biến r;
    - Gán prevX->next = null;
    - Gán tail = prevX;
    - Xóa node r;
  - B2.3. Nếu x khác head, tail:
    - Gán prevX->next = x->next;
    - Xóa node x;
- Bước 3. Thông báo trạng thái xóa node thành công hay thất bại.

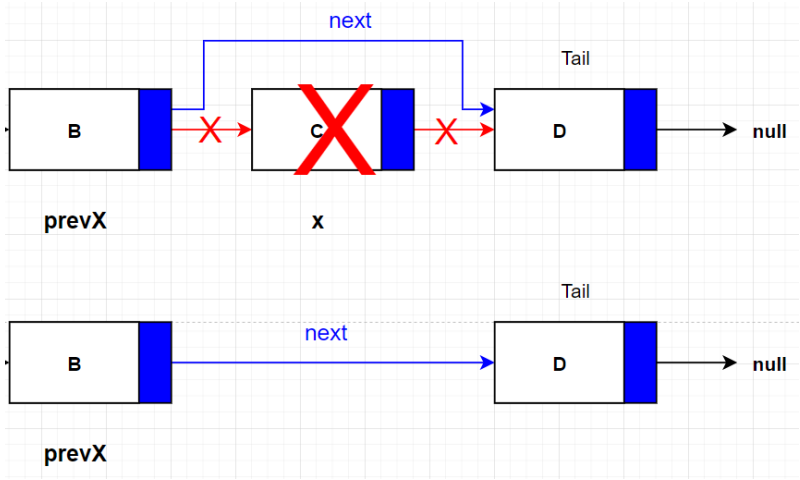
Xóa node head



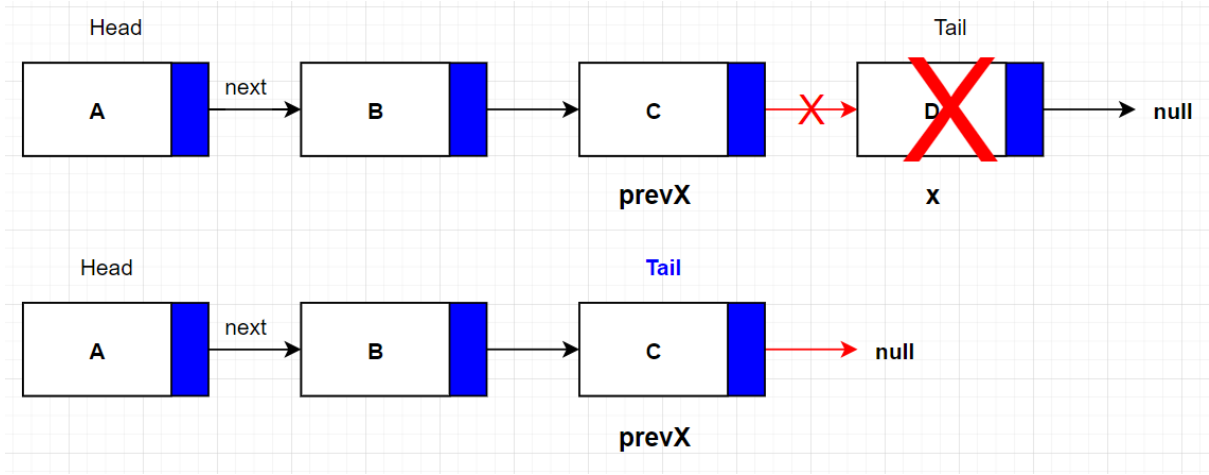

```
bool removeHead() {
    Node<T>* r = head; // lưu lại head
    head = head->next; // cập nhật head mới
    delete r; // xóa head
    return true; // xóa thành công
}
```

Xoá node bất kì

```
bool remove(T data) {
    Node<T>* x = head;
    Node<T>* prevX = head;
    while (x != nullptr) {
        if (x->data == data) {
            break;
        }
        prevX = x;
        x = x->next;
    }
    if (x == head) {
        return removeHead();
    }
    else if (x == tail) {
        return removeTail(prevX);
    }
    else if(x != nullptr) {
        prevX->next = x->next;
        delete x;
        return true; // xóa thành công
    }
    else {
        return false; // xóa thất bại
    }
}
```



Xóa node tail




```
bool removeTail(Node<T>* prevX) {
    Node<T>* r = tail; // lưu lại tail
    prevX->next = nullptr; // cập nhật next của prevX
    tail = prevX; // cập nhật lại tail
    delete r; // xóa node r
    return true; // xóa thành công
}
```

Nội dung tiếp theo

Xóa node khỏi danh sách liên kết đôi