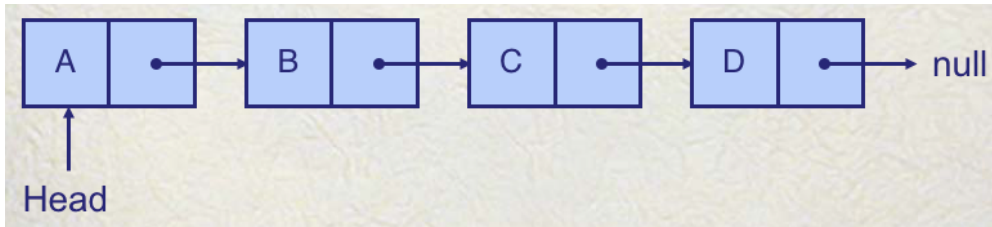


# Cấu trúc danh sách liên kết

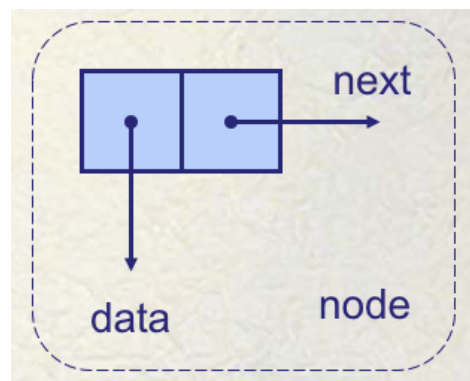
## 1. Giới thiệu

Danh sách liên kết (linked list) được dùng để lưu giữ các phần tử có cùng kiểu giá trị. Bên dưới là ví dụ minh họa sử dụng danh sách liên kết gồm 4 phần tử để lưu giữ dãy A, B, C, D. Phần tử đầu tiên của danh sách liên kết gọi là "head".



Trong khi các phần tử trong mảng được lưu giữ liên tiếp trong bộ nhớ, thì các phần tử trong danh sách liên kết có thể nằm ở các vị trí khác nhau trong bộ nhớ. Mỗi phần tử (node) trong danh sách liên kết gồm hai thành phần:

- "data": Lưu giữ dữ liệu.
- "next": Con trỏ liên kết chỉ đến phần tử tiếp theo trong danh sách liên kết.



## 2. Duyệt trên danh sách liên kết

Bắt đầu từ phần tử đầu tiên "head", di chuyển đến các phần tử khác trên danh sách liên kết dựa vào biến con trỏ liên kết "next" như bên dưới.

```
Method move (Node head):  
    Node current ← head;  
    while (current != NULL) do  
        current ← current → next;
```

## 3. Truy cập phần tử trong danh sách liên kết

Để truy cập phần tử thứ **p** trong danh sách liên kết, chúng ta phải di chuyển từ phần tử đầu tiên (head) cho đến phần tử thứ **p** như bên dưới.

```

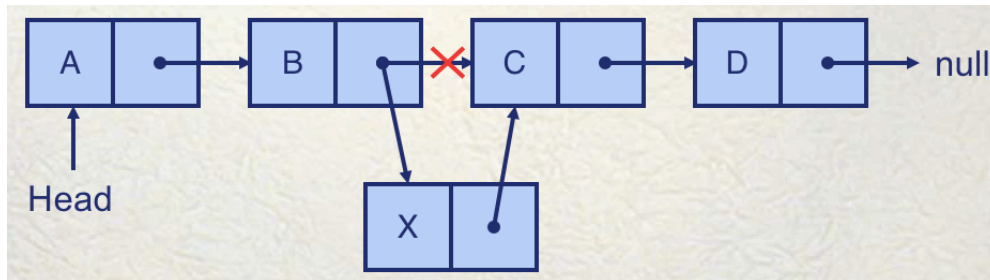
Method element (head, p):
    current ← head;
    index ← 0;
    while (index < p) do
        index ← index + 1;
        current ← current → next;
    return current;

```

#### 4. Chèn một phần tử vào danh sách liên kết

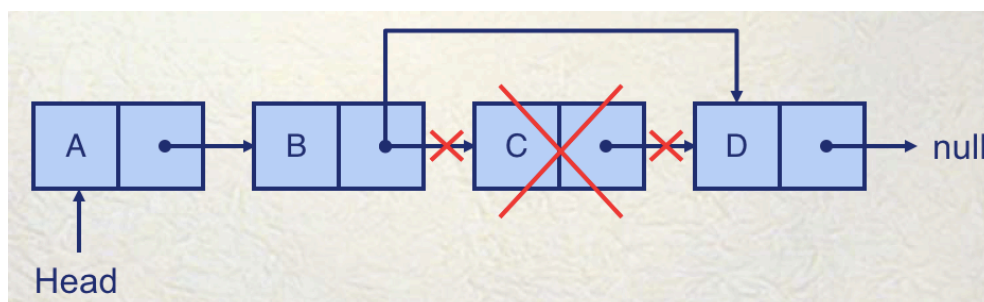
Để chèn một phần tử có giá trị  $X$  vào vị trí  $p$  trong danh sách liên kết, chúng ta tiến hành hai bước cơ bản sau:

- Tạo một phần tử mới chứa giá trị  $X$ .
- Gán giá trị con trỏ liên kết "next" của phần tử thứ  $(p-1)$  và phần tử mới như bên dưới.



#### 5. Xóa một phần tử khỏi danh sách liên kết

Để xóa phần tử tại vị trí  $p$  khỏi danh sách liên kết, chúng ta thay đổi giá trị con trỏ liên kết "next" của phần tử thứ  $(p-1)$  như hình dưới. Sau đó xóa phần tử thứ  $p$  ra khỏi bộ nhớ!

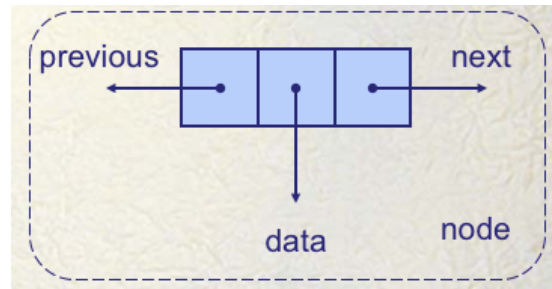


#### 6. Danh sách liên kết đôi

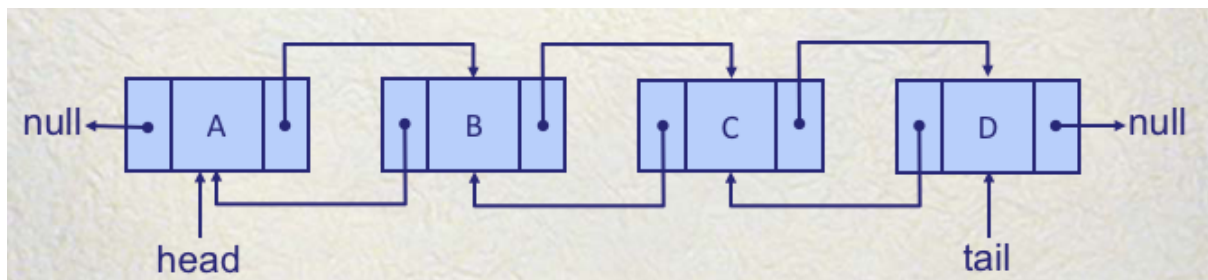
Danh sách liên kết được trình bày ở trên được gọi là danh sách liên kết đơn, bởi vì mỗi phần tử trong danh sách liên kết đơn có đúng một biến con trỏ liên kết "next" chỉ đến phần tử tiếp theo trong danh sách. Việc duyệt trên danh sách liên kết đơn chỉ thực hiện được theo một chiều từ đầu danh sách đến cuối danh sách và không thực hiện được theo chiều ngược lại.

Danh sách liên kết đôi là mở rộng của danh sách liên kết đơn, trong đó mỗi phần tử có hai con trỏ liên kết:

- “previous” trỏ đến phần tử phía trước trong danh sách.
- “next” trỏ đến phần tử tiếp theo trong danh sách.



Bên dưới là ví dụ minh họa sử dụng danh sách liên kết đôi để lưu giữ dãy A, B, C, D. Phần tử đầu tiên của danh sách liên kết đôi gọi là “head”, phần tử cuối cùng gọi là “tail”.



Danh sách liên kết đôi thường được dùng trong các bài toán cần duyệt trên danh sách theo cả hai chiều. Nó cũng hay được dùng trong các bài toán mà các phép toán chèn và xóa các phần tử trong danh sách thường diễn ra ở một trong hai đầu của danh sách bởi vì các phép toán này được thực hiện một cách nhanh chóng.

## 7. So sánh cấu trúc danh sách liên kết và cấu trúc mảng

- Các phần tử trong mảng được lưu giữ liên tục trong bộ nhớ, cho nên việc truy cập và thay đổi giá trị một phần tử bất kì trong mảng rất nhanh. Các phần tử trong danh sách liên kết được lưu giữ tại các vị trí khác nhau trong bộ nhớ máy tính, cho nên việc truy cập và thay đổi giá trị một phần tử bất kì tốn nhiều thời gian.
- Kích thước của mảng phải được xác định và khai báo trước khi sử dụng, điều này dẫn đến khó khăn và dư thừa bộ nhớ trong các bài toán mà số lượng các phần tử không được xác định trước. Khi sử dụng danh sách liên kết, chúng ta không phải khai báo trước kích thước của danh sách. Chúng ta chỉ xin cấp phát bộ nhớ khi thêm phần tử mới vào danh sách liên kết, và sẽ xóa phần tử đó khỏi bộ nhớ khi nó bị xóa khỏi danh sách liên kết.