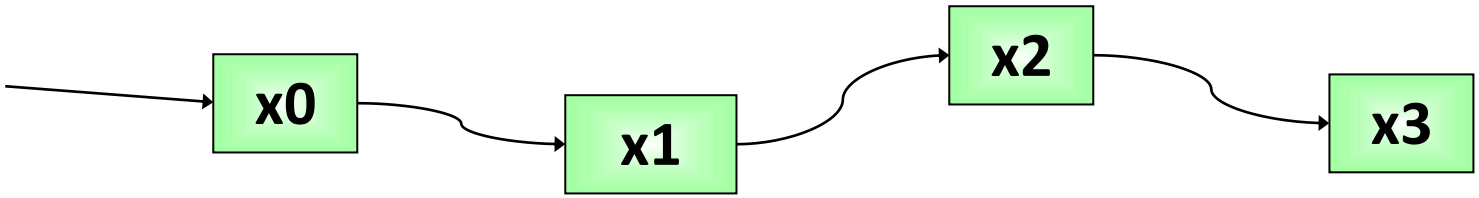
NỘI DUNG

DANH SÁCH LIÊN KẾT ĐƠN (LIST)

Tổ Chức Của DSLK Đơn



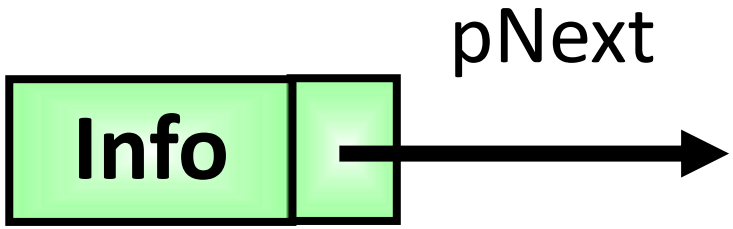
- Mỗi phần tử liên kết với phần tử đứng liền sau trong danh sách
- Mỗi phần tử trong danh sách liên kết đơn là một cấu trúc có hai thành phần
  - ***Thành phần dữ liệu***: Lưu trữ thông tin về bản thân phần tử
  - ***Thành phần liên kết***: Lưu địa chỉ phần tử đứng sau trong danh sách hoặc bằng NULL nếu là phần tử cuối danh sách.

CTDL của DSLK đơn

- Cấu trúc dữ liệu của 1 nút trong List đơn

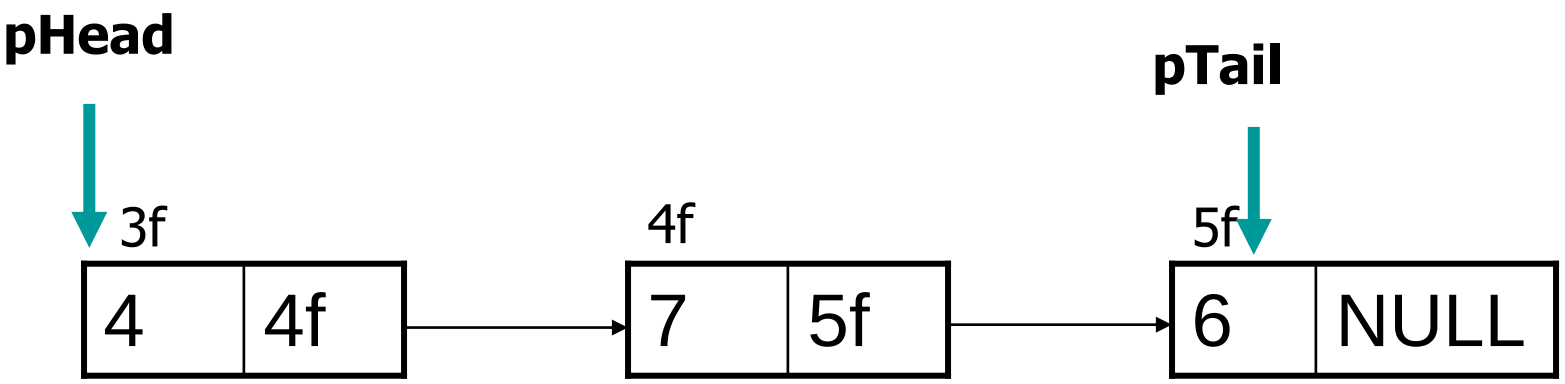
```
struct Node
{  Data   Info;  // Lưu thông tin bản thân
    Node  *pNext; //Lưu địa chỉ của Node đứng sau
};
```

- Cấu trúc dữ liệu của DSLK đơn




```
struct List
{  Node  *pHead;//Lưu địa chỉ Node đầu tiên trong List
   Node  *pTail; //Lưu địa chỉ của Node cuối cùng trong List
};  // kiểu danh sách liên kết đơn
```

Ví dụ tổ chức DSLK đơn trong bộ nhớ



Trong ví dụ trên thành phần dữ liệu là 1 số nguyên

Các thao tác cơ bản trên DSLK đơn

- Tạo 1 danh sách liên kết đơn rỗng
- Tạo 1 nút có trường Infor bằng x
- Tìm một phần tử có Info bằng x
- Thêm một phần tử có khóa x vào danh sách
- Hủy một phần tử trong danh sách
- Duyệt danh sách
- Sắp xếp danh sách liên kết đơn

Khởi tạo danh sách liên kết

- Địa chỉ của nút đầu tiên, địa chỉ của nút cuối cùng đều không có

```
void CreateList(List &l)
{
    l.pHead=l.pTail=NULL;
}
```


Tạo 1 phần tử mới

- Hàm trả về địa chỉ phần tử mới tạo

```
Node*    CreateNode(Data x)
{  Node *p = new Node;//Cấp phát vùng nhớ cho phần tử
   if ( p==NULL) return p;
   p ->Info = x; //gán dữa liệu cho nút
   p->pNext = NULL;
   return p;
}
```


Thêm 1 phần tử vào DSLK

- **Nguyên tắc thêm:** Khi thêm 1 phần tử vào List thì có làm cho pHead, pTail thay đổi?
- **Các vị trí cần thêm 1 phần tử vào List:**
  - Thêm vào đầu List đơn
  - Thêm vào cuối List
  - Thêm vào sau 1 phần tử q trong list

Thuật toán thêm 1 phần tử vào đầu DSLK

- Thêm nút p vào đầu danh sách liên kết đơn

Bắt đầu:

Nếu List rỗng thì

+ pHead = pTail = p;

Ngược lại

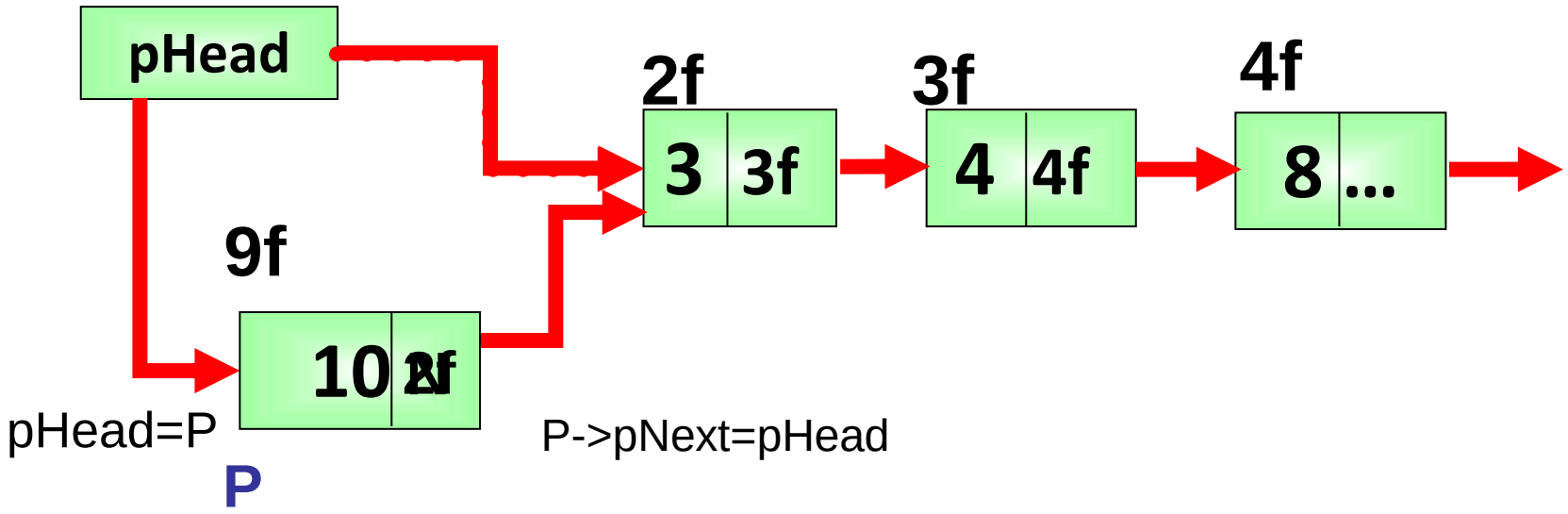
+ p->pNext = pHead;

+ pHead = p

Hàm thêm 1 phần tử vào đầu List

```
void AddHead(LIST &l, Node* p)
{
        if (l.pHead==NULL)
        {
                l.pHead = l.pTail = p;
        }
        else
        {
                p->pNext = l.pHead;
                l.pHead = p;
        }
}
```


Minh họa thuật toán thêm vào đầu



Thuật toán thêm vào cuối DSLK

- Ta cần thêm nút p vào cuối list đơn

Bắt đầu:

Nếu List rỗng thì

+ pHead = pTail = p;

Ngược lại

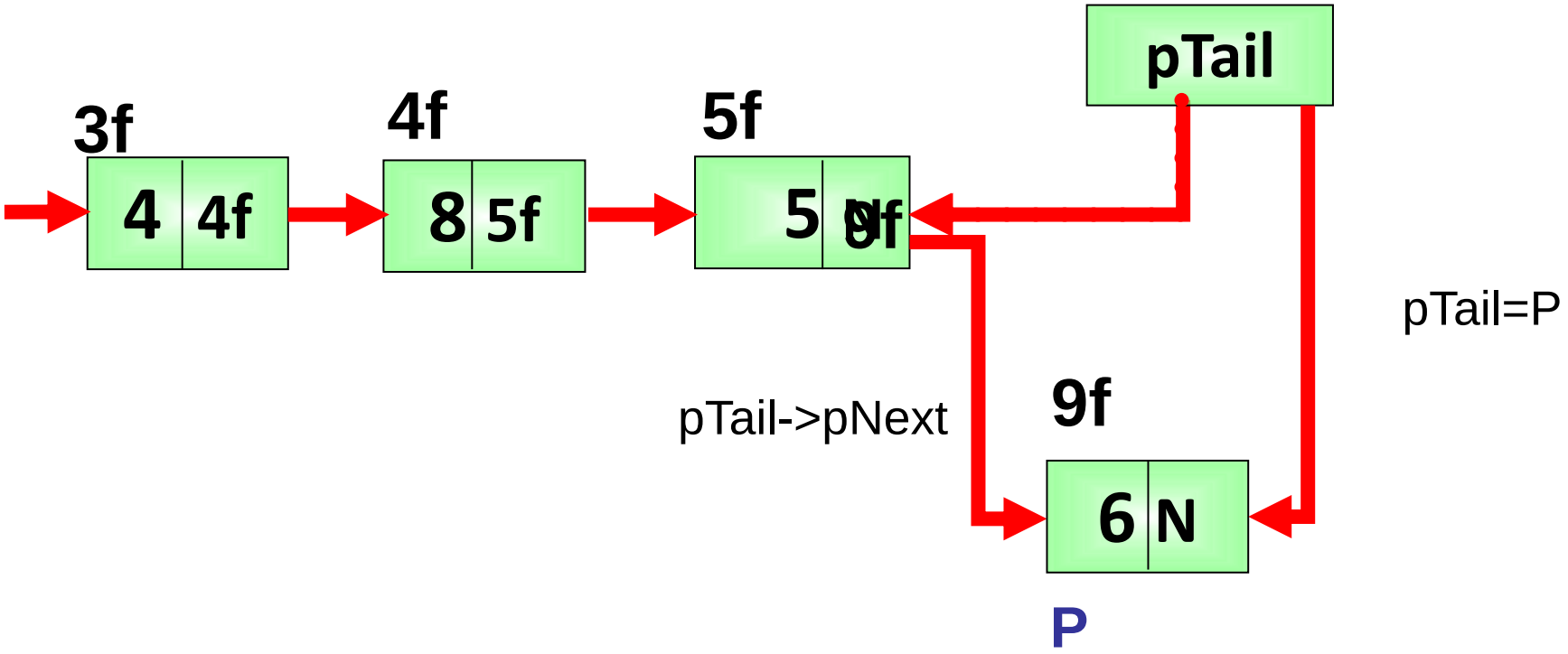
+ pTail->pNext=p;

+ pTail=p

Hàm thêm 1 phần tử vào cuối DSLKD

```
void AddTail(LIST &l, Node *p)
{
    if (l.pHead==NULL)
    {
        l.pHead = l.pTail = p;
    }
    else
    {
        l.pTail->Next = p;
        l.pTail = p;
    }
}
```


Minh họa thuật toán thêm vào cuối



Thuật toán phần tử q vào sau phần tử q

- Ta cần thêm nút p vào sau nút q trong list đơn

Bắt đầu:

Nếu (q!=NULL) thì

B1: p->pNext = q->pNext

B2:

+ q->pNext = p

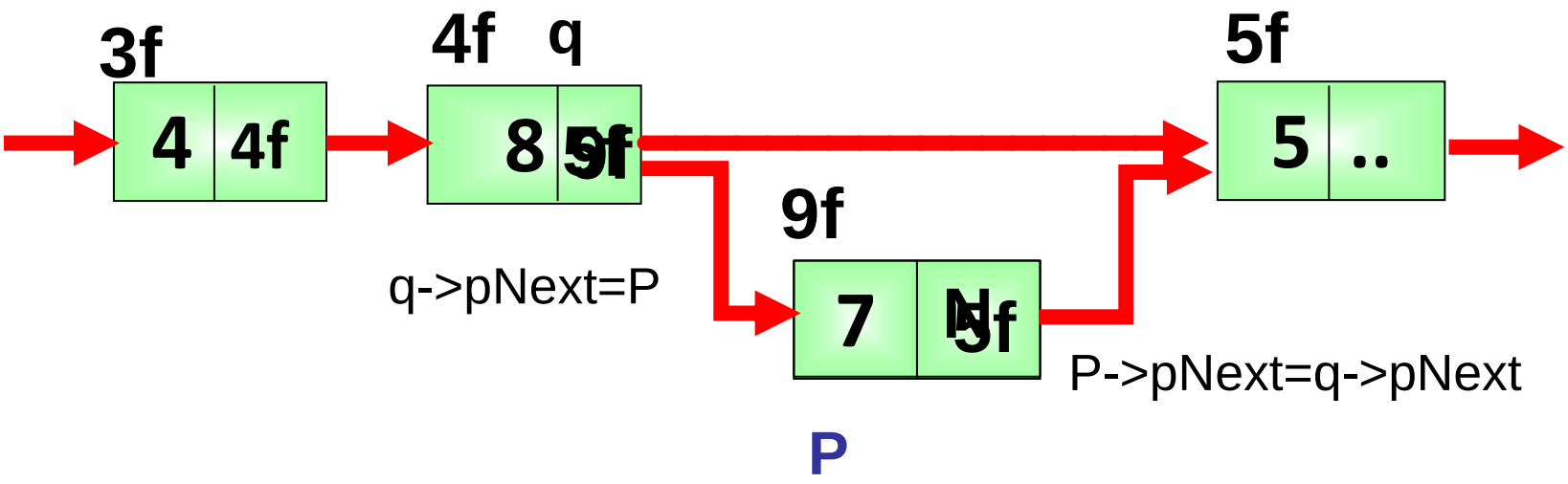
+ nếu q = pTail thì

pTail=p

Cài đặt thuật toán

```
void InsertAfterQ(List &l, Node *p, Node *q)
{
   if(q!=NULL)
   {
      p->pNext=q->Next;
      q->pNext=p;
      if(l.pTail==q)
            l.Tail=q;
   }
   else
      AddHead(l,q);// thêm q vào đầu list
}
```


Minh họa thuật toán



Tìm 1 phần tử trong DSLK đơn

- Tìm tuần tự (hàm trả về), các bước của thuật toán tìm nút có Info bằng x trong list đơn

Bước 1: p=pHead;// địa chỉ của phần tử đầu trong list đơn

Bước 2:

Trong khi p!=NULL và p->Info!=x

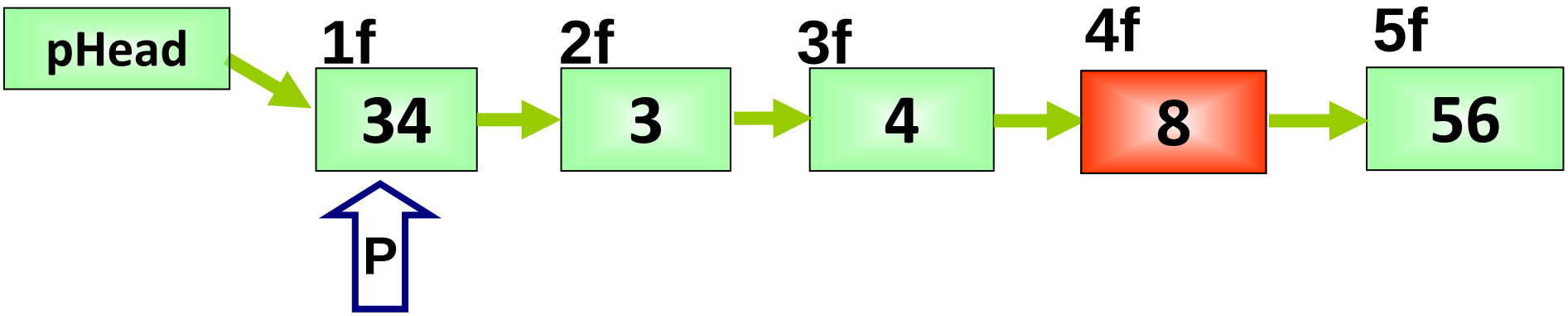
p=p->pNext;// xét phần tử kế

Bước 3:

+ Nếu p!=NULL thì p lưu địa chỉ của nút có Info = x

+ Ngược lại : Không có phần tử cần tìm

Minh họa thuật toán tìm phần tử trong DSLK



X = 8

Tìm thấy, hàm trả về địa chỉ của nút tìm thấy là 4f

Duyệt danh sách

- Duyệt danh sách là thao tác thường được thực hiện khi có nhu cầu cần xử lý các phần tử trong danh sách như:
  - Đếm các phần tử trong danh sách
  - Tìm tất cả các phần tử trong danh sách thoả điều kiện
  - Hủy toàn bộ danh sách

Thuật toán duyệt danh sách

- Bước 1:

  p = pHead;// p lưu địa chỉ của phần tử đầu trong List

- Bước 2:

  Trong khi (danh sách chưa hết) thực hiện

  + xử lý phần tử p

  + p=p->pNext;// qua phần tử kế

Cài đặt in các phần tử trong List

```
void PrintList(List l)
{
    Node *p=l.pHead;
    while(p!=NULL)
    {       cout<< p->Info;
            p=p->pNext;
    }
}
```