Cấu trúc dữ liệu

CÁC KIỂU DỮ LIỆU TRỪU TƯỢNG CƠ BẢN (BASIC ABSTRACT DATA TYPES)

Bộ môn Công Nghệ Phần Mềm

MỤC TIÊU

- Nắm vững các kiểu dữ liệu trừu tượng như: danh sách, ngăn xếp, hàng đợi.
- Cài đặt các kiểu dữ liệu trừu tượng bằng ngôn ngữ lập trình cụ thể.
- Ứng dụng được các kiểu dữ liệu trừu tượng trong bài toán thực tế.

NỘI DUNG

- Kiểu dữ liệu trừu tượng danh sách (LIST)
- Kiểu dữ liệu trừu tượng ngăn xếp (STACK)
- Kiểu dữ liệu trừu tượng hàng đợi (QUEUE)
- *Danh sách liên kết kép (Doubly-Linked Lists)*

NỘI DUNG

- Kiểu dữ liệu trừu tượng danh sách (LIST)
- Kiểu dữ liệu trừu tượng ngăn xếp (STACK)
- Kiểu dữ liệu trừu tượng hàng đợi (QUEUE)
- *Danh sách liên kết kép (Doubly-Linked Lists)*

DANH SÁCH

- Khái niệm danh sách
- Các phép toán trên danh sách
- Cài đặt danh sách
  - Dùng mảng (Danh sách ĐẶC)
  - Dùng con trỏ (Danh sách LIÊN KẾT)

KHÁI NIỆM VỀ DANH SÁCH

- Là tập hợp hữu hạn các phần tử có cùng kiểu.
- Kiểu chung được gọi là kiểu phần tử (element type).
- Ta thường biểu diễn dạng: $a_1, a_2, a_3, ..., a_n$.
- Nếu
  - n=0: danh sách rỗng.
  - n>0: phần tử đầu tiên là a_1, phần tử cuối cùng là a_n.
- Độ dài của danh sách: số phần tử của danh sách.
- Các phần tử trong danh sách có thứ tự tuyến tính theo vị trí xuất hiện. Ta nói a_i đứng trước a_{i+1} (i=1..n-1).

CÁC PHÉP TOÁN TRÊN DANH SÁCH

Tên phép toán	Công dụng
MakeNull_List(L)	Khởi tạo một danh sách L rỗng
Empty_List(L)	Kiểm tra xem danh sách L có rỗng hay không
Full_List(L)	Kiểm tra xem danh sách L có đầy hay không
Insert_List(X,P,L)	Xen phần tử có nội dung X vào danh sách L tại vị trí P
Delete_List(P,L)	Xóa phần tử tại vị trí P trong danh sách L
EndList(L)	Trả về vị trí sau phần tử cuối trong ds L
Locate_List(X,L)	Trả về kết quả là vị trí của phần tử có nội dung X đầu tiên trong danh sách L, EndList(L) nếu không tìm thấy

CANTHO UNIVERSITY

CÁC PHÉP TOÁN TRÊN DANH SÁCH

Tên phép toán	Công dụng
Retrieve(P,L)	Trả về nội dung phần tử tại vị trí P trong danh sách L, thông báo lỗi nếu vị trí P không có trong ds
Next(P,L)	Trả về kết quả là vị trí của phần tử đi sau phần tử tại vị trí P trong danh sách L, EndList(L) nếu phần tử tại vị trí P là phần tử cuối cùng, thông báo lỗi nếu vị trí P không có trong danh sách
Previous(P,L)	Trả về kết quả là vị trí của phần tử đứng trước phần tử tại vị trí P trong danh sách L, thông báo lỗi nếu ví trí P là vị trí đầu tiên hoặc không có trong danh sách L
First(L)	Trả về kết quả là vị trí của phần tử đầu danh sách, EndList(L) nếu danh sách rỗng
Print_List(L)	Hiển thị các phần tử trong danh sách L theo thứ tự xuất hiện

CÁC PHÉP TOÁN - BÀI TẬP

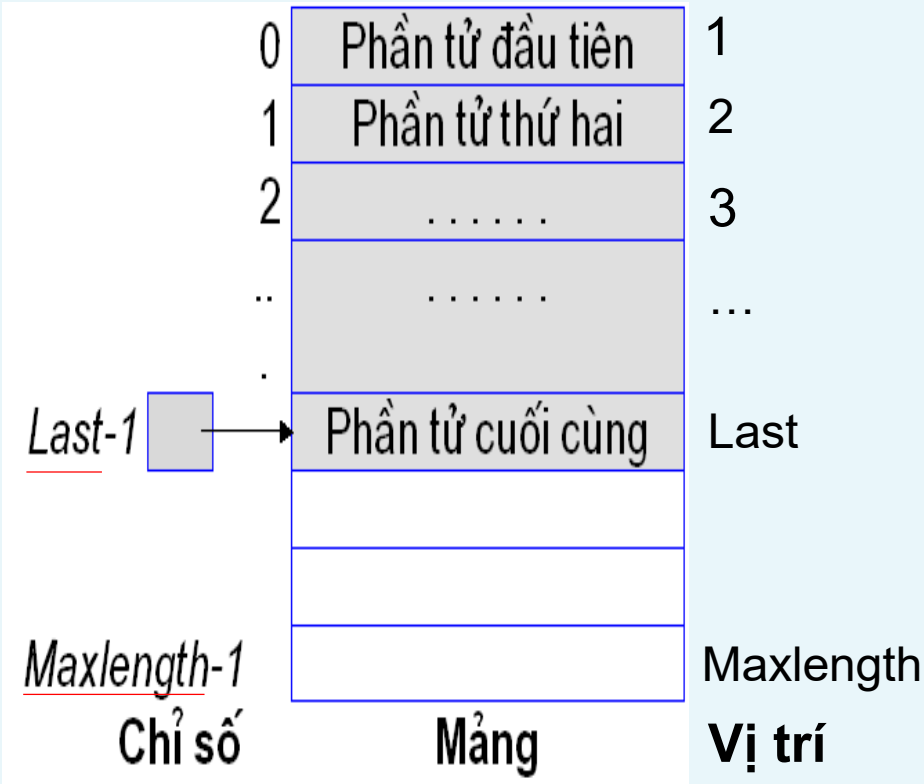
- Muốn thêm phần tử x vào đầu hay cuối danh sách ta gọi phép toán nào và gọi phép toán đó như thế nào?

CANTHO UNIVERSITY

CÀI ĐẶT DANH SÁCH

- Khái niệm danh sách
- Các phép toán trên danh sách
- Cài đặt danh sách
  - Dùng mảng (Danh sách ĐẶC)
  - Dùng con trỏ (DS LIÊN KẾT)

CÀI ĐẶT DANH SÁCH BẰNG MẢNG (DANH SÁCH ĐẶC)



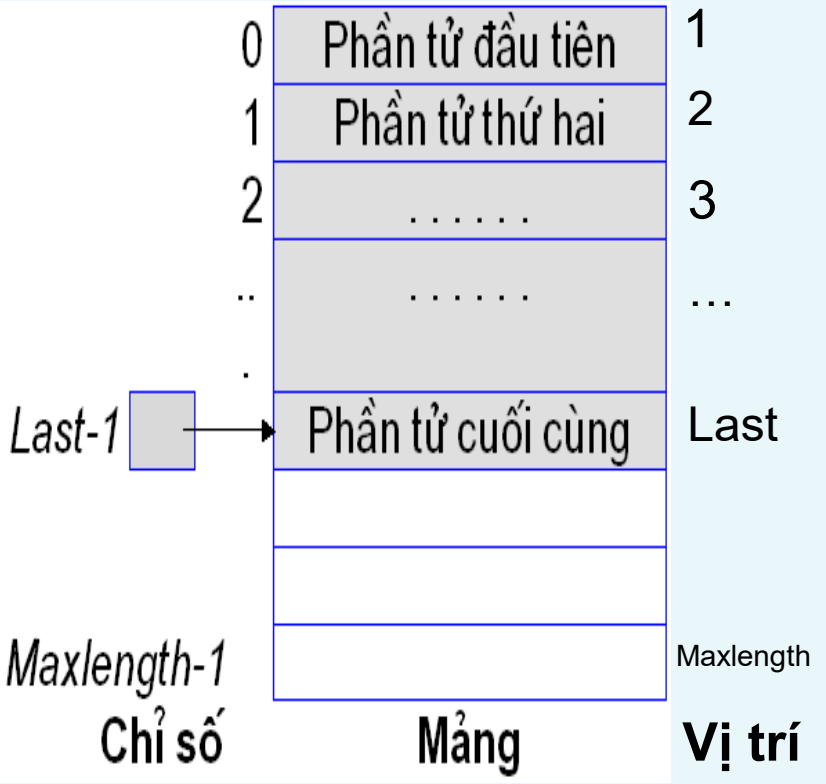
Elements

- Dùng 1 *mảng* (Elements) để lưu trữ liên tiếp các phần tử, bắt đầu từ vị trí đầu tiên.
- Ta phải ước lượng *số phần tử tối đa* của danh sách (Maxlength).
- Ta phải lưu trữ *độ dài hiện tại* của danh sách (Last).

Ta định nghĩa vị trí của một phần tử trong danh sách là "chỉ số của mảng tại vị trí lưu trữ phần tử đó + 1".

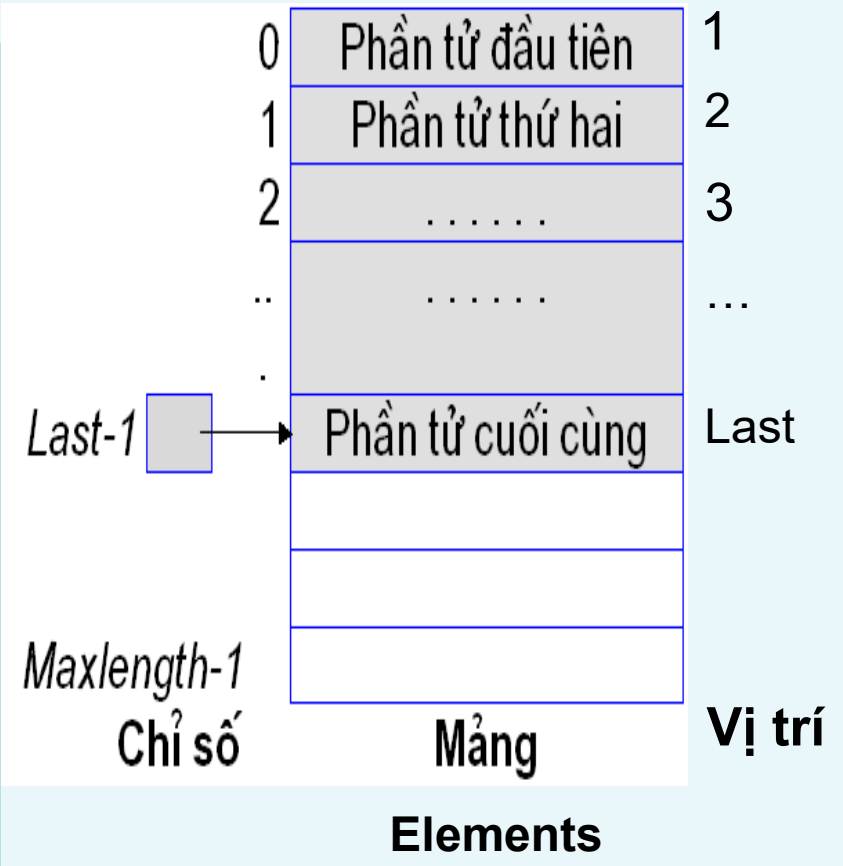
KHAI BÁO

```
//Độ dài tối đa của danh sách-ds
#define MaxLength <n>
//kiểu của phần tử trong ds
typedef <datatype> ElementType;
//kiểu vị trí cuả các phần tử
typedef int Position;
typedef struct {
   //mảng chứa các phần tử của ds
   ElementType Elements[MaxLength];
   //giữ độ dài danh sách
   Position Last;
 } List;
List L;
```

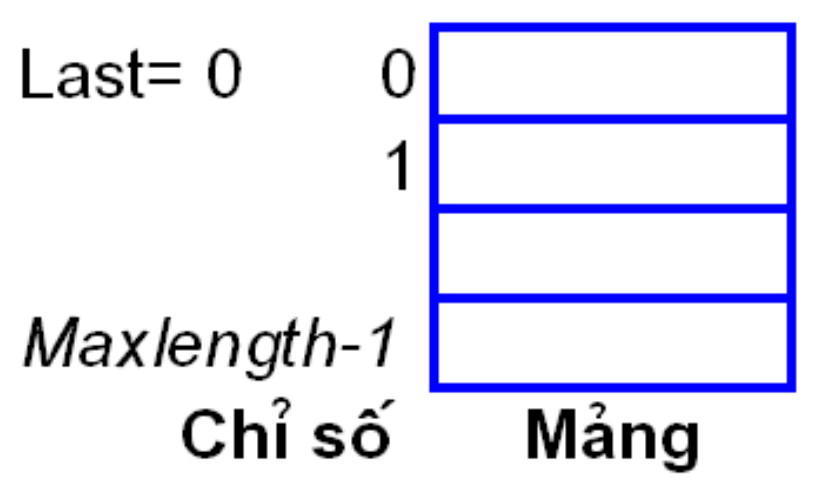


KHAI BÁO

```
//Độ dài tối đa của danh sách
#define MaxLength 300
//kiểu của phần tử trong danh sách
typedef int ElementType;
//kiểu vị trí cuả các phần tử
typedef int Position;
typedef struct {
   //mảng chứa các phần tử của ds
   ElementType Elements[MaxLength];
   //giữ độ dài danh sách
   Position Last;
 } List;
List L;
```



KHỞI TẠO DANH SÁCH RỖNG



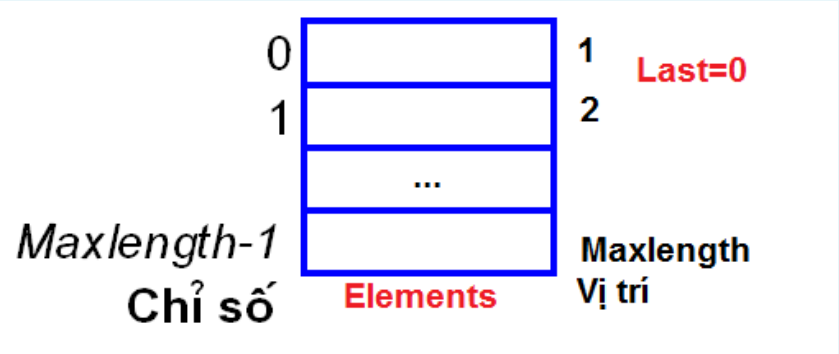
- Cho độ dài danh sách bằng 0.

```
void MakeNull_List(List *L)
{
        L->Last=0;
}
```

KIỂM TRA DANH SÁCH RỖNG

- Xem độ dài danh sách có bằng 0 không?

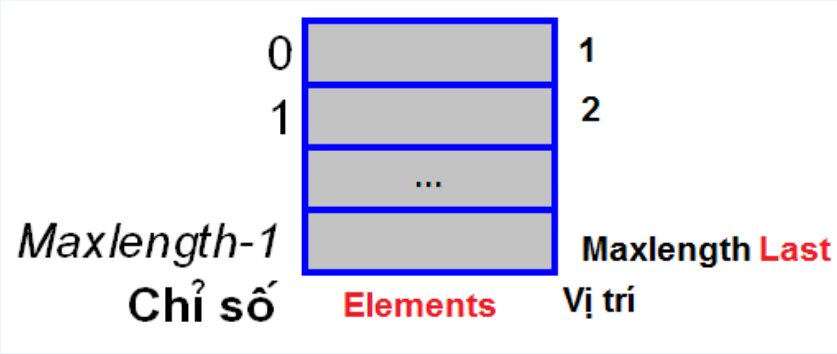
```
int Empty_List(List L)
{
        return L.Last==0;
}
```



KIỂM TRA DANH SÁCH ĐẦY

- Xem độ dài danh sách có bằng MaxLength không?

```
int Full_List(List L)
{
        return L.Last==MaxLength;
}
```

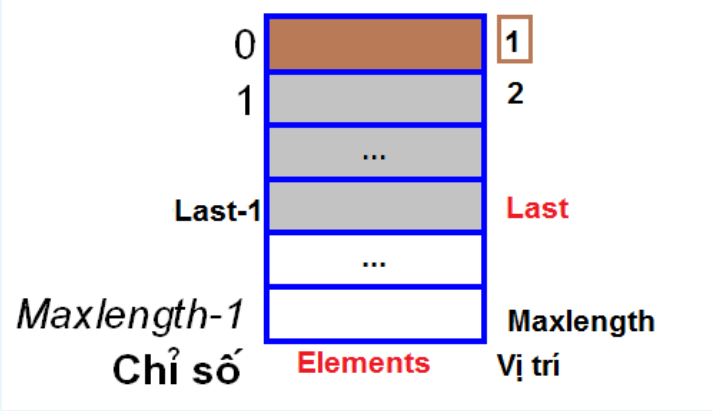


CANTHO UNIVERSITY

XÁC ĐỊNH VỊ TRÍ ĐẦU / SAU VỊ TRÍ CUỐI

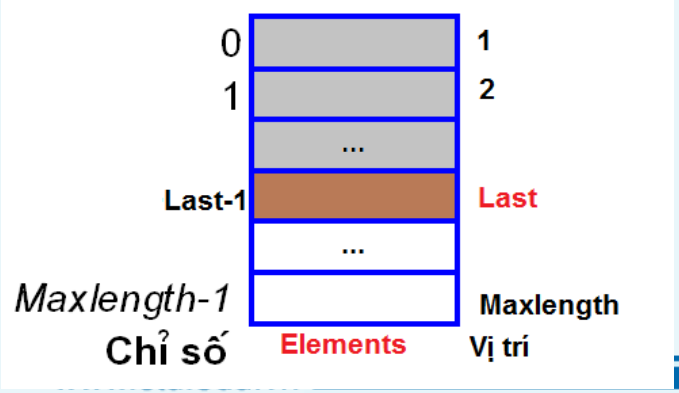
- Xác định vị trí đầu tiên trong danh sách.

```
Position First(List L)
{
   return 1;
}
```



- Xác định vị trí sau phần tử cuối trong danh sách.

```
Position EndList(List L)
{
  return L.Last+1;
}
```



XEN PHẦN TỬ X VÀO VỊ TRÍ P

- Xen phần tử x='k' vào vị trí p=3 trong danh sách L (chỉ số 2 trong mảng).

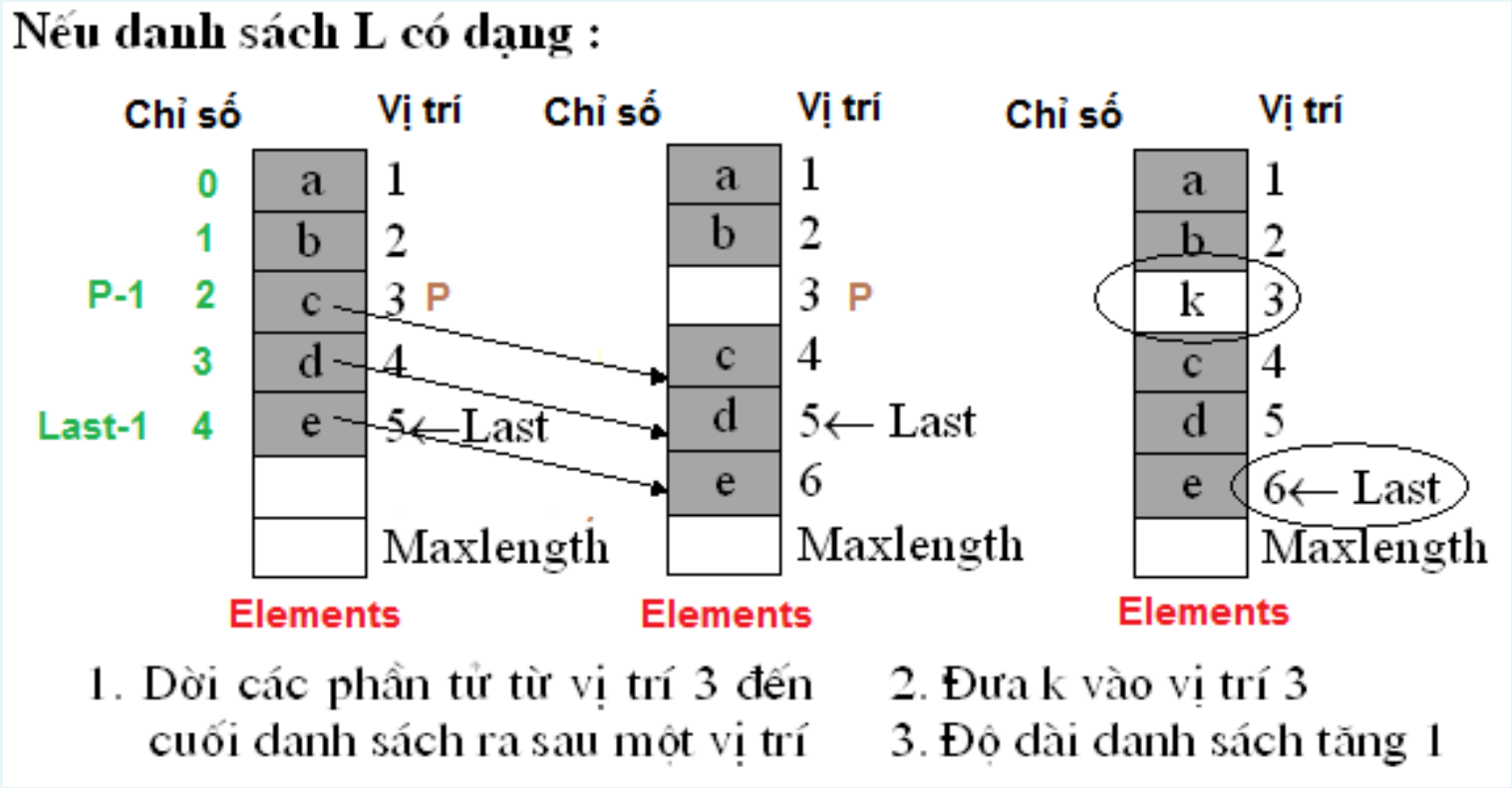
- **Nếu L có dạng :**

⇨ Trường hợp này mảng đầy => báo lỗi

Chỉ số mảng	Nội dung	Vị trí trong dsách
0	a	1
1	b	2
2	c	3
3	d	4
4	g	:
5	h	Maxlength ←Last

XEN PHẦN TỬ X VÀO VỊ TRÍ P

- Xen phần tử x=’k’ vào vị trí p=3 trong danh sách L (chỉ số 2 trong mảng).



XEN PHẦN TỬ X VÀO VỊ TRÍ P

- Thuật toán

  Để chèn x vào vị trí p của L, ta làm như sau:

  – Nếu mảng đầy thì thông báo lỗi.

  – Ngược lại, nếu vị trí p không hợp lệ thì báo lỗi.

  – Ngược lại:

    - Dời các phần tử từ vị trí p đến cuối danh sách ra sau một vị trí.
    - Đưa phần tử mới x vào tại vị trí p.
    - Độ dài danh sách tăng 1.

XEN PHẦN TỬ X VÀO VỊ TRÍ P

CANTHO UNIVERSITY

Cài đặt

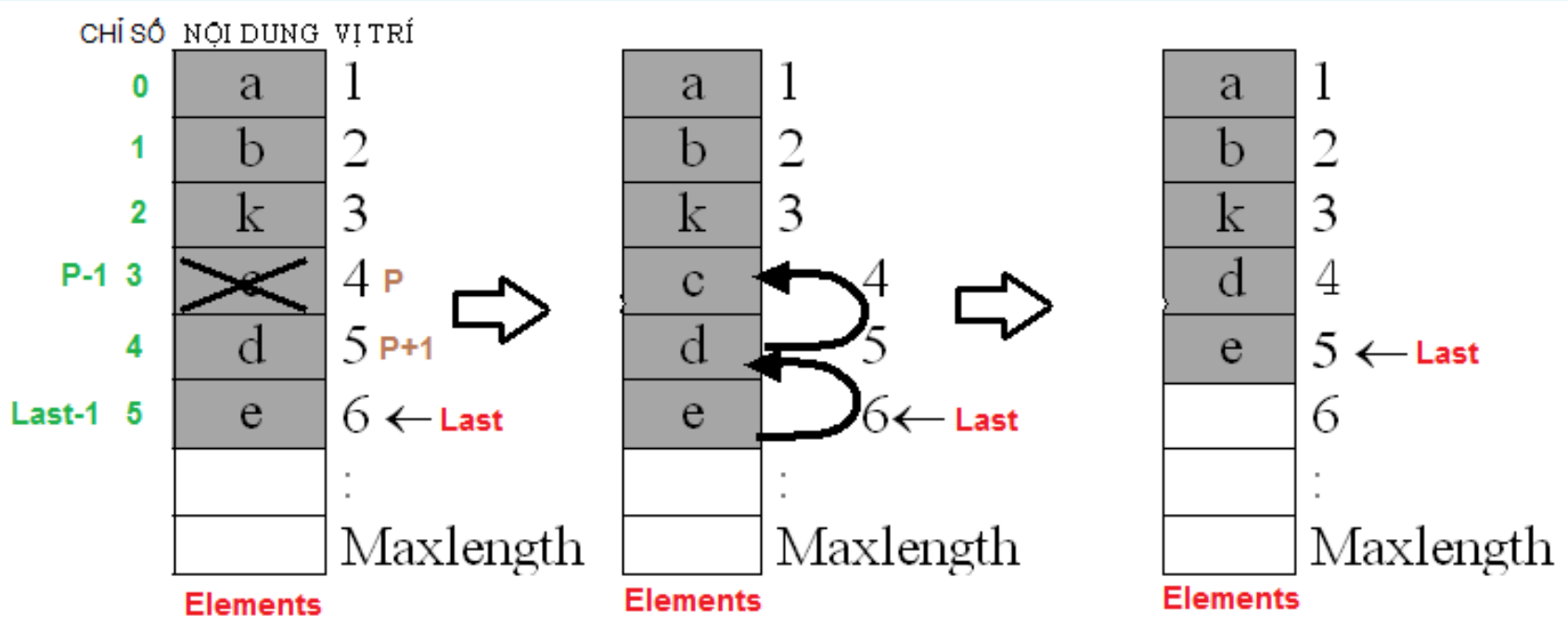
Full_List(*L)

P<First(*L)) || (P>EndList(*L))

```
void Insert_List(ElementType X,Position P, List *L){
    if (L->Last==MaxLength)
       printf("Danh sach day");
    else if ((P<1 || (P>(L->Last+1)))
       printf("Vi tri khong hop le");
    else {
       Position Q;
       /*Dời các phần tử từ vị trí p
      đến cuối dsách ra sau 1 vị trí*/
       for(Q=L->Last;Q>P-1;Q--)
          L->Elements[Q]=L->Elements[Q-1];
          //Đưa x vào vị trí p
          L->Elements[P-1]=X;
          //Tăng độ dài danh sách lên 1
          L->Last++;
    }  }
```

XÓA MỘT PHẦN TỬ TẠI VỊ TRÍ P TRONG DS

- Ví dụ: Xóa phần tử vị trí p=4 của L.



XÓA MỘT PHẦN TỬ TẠI VỊ TRÍ P TRONG DS

Thuật toán

- Nếu p là một vị trí không hợp lệ thì thông báo lỗi.
- Ngược lại:
  - Di dời các phần tử từ vị trí p+1 đến cuối danh sách ra trước một vị trí.
  - Độ dài của danh sách giảm 1.

XÓA MỘT PHẦN TỬ TẠI VỊ TRÍ P TRONG DS

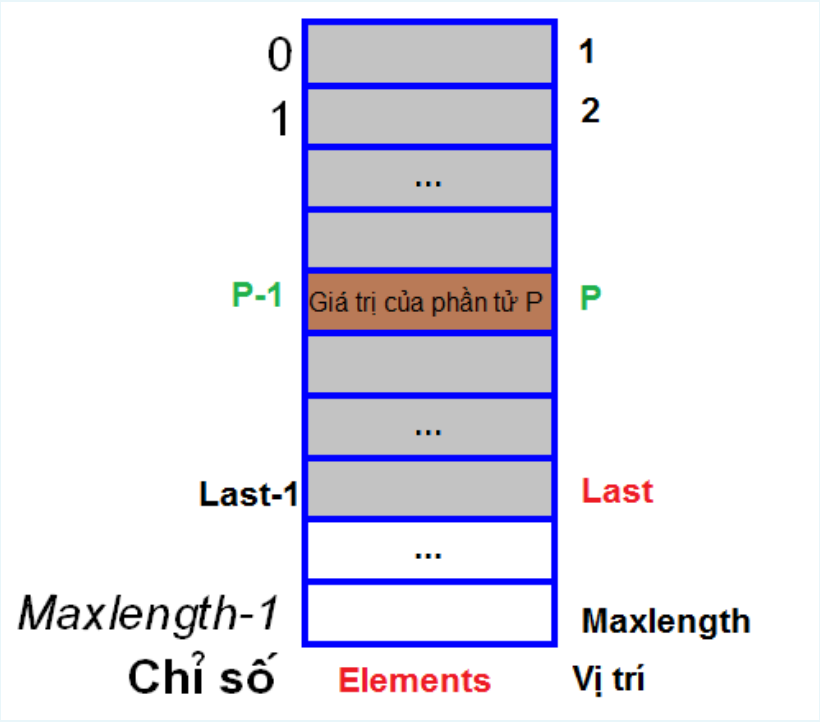
Cài đặt

```
void Delete_List(Position P,List *L){
  if ((P<1) || (P>L->Last))
      printf("Vi tri khong hop le");
  else if (Empty_List(*L))
      printf("Danh sach rong!");
  else{
      Position Q;
      /*Dời các phần tử từ vị trí p+1 đến cuối
        danh sách ra trước 1 vị trí*/
      for(Q=P-1;Q<L->Last-1;Q++)
         L->Elements[Q]=L->Elements[Q+1];
      L->Last--;
  }
}
```

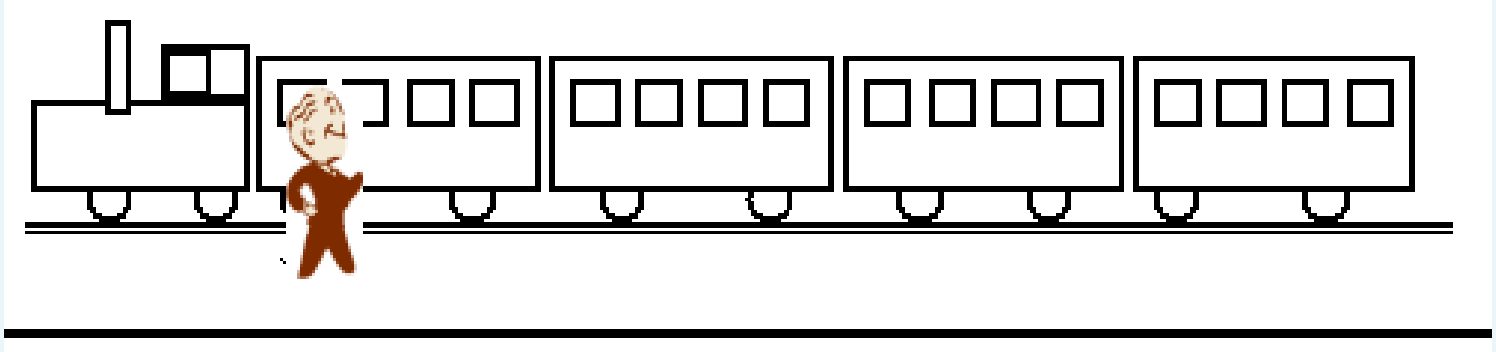
XÁC ĐỊNH NỘI DUNG PHẦN TỬ TẠI VỊ TRÍ P

- Xác định nội dung phần tử tại vị trí P trong DS.

```
ElementType Retrieve(Position P, List L)
{
   return L.Elements[P-1];
}
```



TÌM KIẾM PHẦN TỬ X TRONG DS



Thuật toán

- Bắt đầu từ phần tử đầu tiền trong danh sách, ta tiến hành tìm từ đầu danh sách cho đến khi tìm thấy hoặc cuối danh sách.
  - Nếu giá trị tại vị trí P bằng X

    ```
    Retrieve(P,L) == X
    ```

    thì dừng tìm kiếm.
  - Ngược lại (giá trị tại vị trí P khác X) thì đến vị trí kế tiếp

    ```
    P = Next(P,L)
    ```
- Trả về vị trí phần tử được tìm thấy hoặc vị trí Last+1(EndList) nếu không tìm thấy.

TÌM KIẾM PHẦN TỬ X TRONG DS

Cài đặt

```
Position Locate(ElementType X, List L){
   Position P;
   int Found = 0;
   P = First(L); //vị trí phần tử đầu tiên
   /*trong khi chưa tìm thấy và chưa kết
   thúc danh sách thì xét phần tử kế tiếp*/
   while ((P != EndList(L)) && (Found == 0))
      if (Retrieve(P,L) == X) Found = 1;
      else P = Next(P, L);
   return P;
}
```

ĐÁNH GIÁ GIẢI THUẬT TÌM KIẾM

- Thời gian tìm kiếm
  - Nhanh nhất (tốt nhất) là khi nào, x ở đâu?
  - Xấu nhất khi nào?
- Độ phức tạp của giải thuật thường được xác định là trong trường hợp xấu nhất O(n).

CÁC PHÉP TOÁN KHÁC

- Xác định vị trí kế tiếp trong danh sách.

```
Position Next(Position P, List L)
{
   return P+1;
}
```

- Xác định vị trí trước trong danh sách.

```
Position Previous(Position P, List L)
{
   return P-1;
}
```

CÁC PHÉP TOÁN KHÁC

- In danh sách.

```
void Print_List(List L){
   Position P= First(L);
   while (P != EndList(L)){
         printf("%d ", L.Elements[P-1]);
         P=Next(P,L);
   }
   printf("\n");
}
```

Retrieve(P,L)

BÀI TẬP

- Vận dụng các phép toán trên danh sách đặc để viết chương trình nhập vào một danh sách các số nguyên và hiển thị danh sách vừa nhập ra màn hình.
- Thêm phần tử có nội dung x vào danh sách tại ví trí p (trong đó x và p được nhập từ bàn phím).
- Xóa phần tử đầu tiên có nội dung x (nhập từ bàn phím) ra khỏi danh sách.
- Sử dụng các phép toán trừu tượng trên danh sách, hãy viết hàm delete_duplicate(LIST L) loại bỏ những gia trị trùng lặp trong danh sách.

Q&A?