



CANTHO UNIVERSITY

Cấu trúc dữ liệu

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

Bộ môn Công Nghệ Phần Mềm



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 (Double – Lists)



NGĂN XẾP (STACK)

- ĐỊNH NGHĨA
- CÁC PHÉP TOÁN
- CÀI ĐẶT
 - CÀI ĐẶT BẰNG DANH SÁCH LIÊN KẾT
 - CÀI ĐẶT BẰNG MẢNG



NGĂN XẾP (STACK)

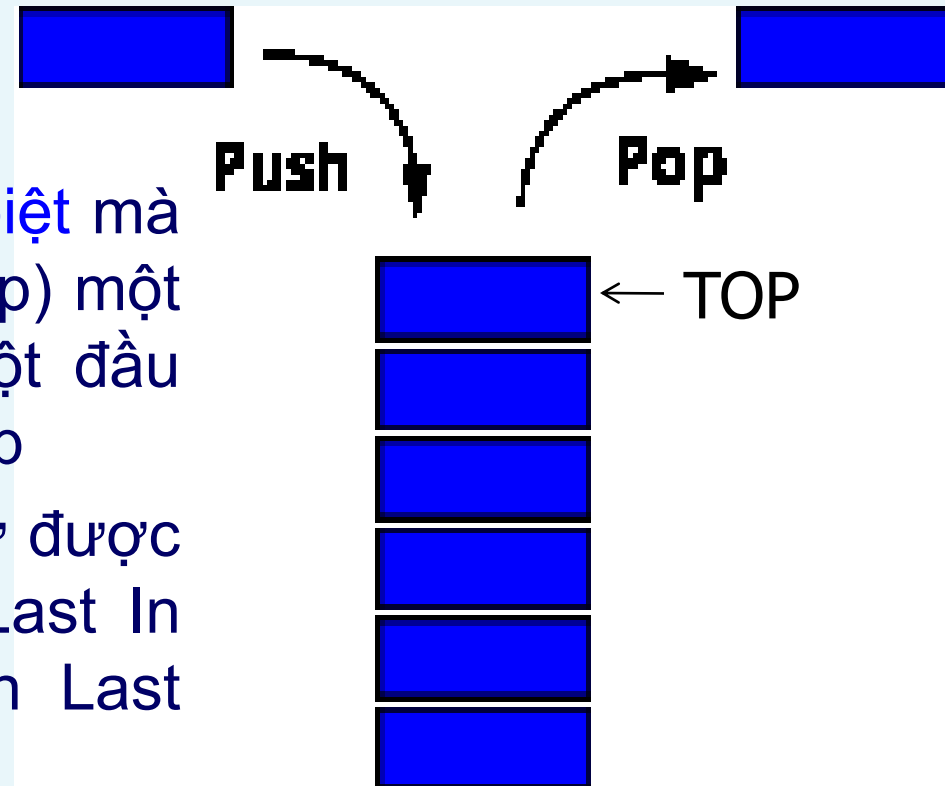
- ĐỊNH NGHĨA
- CÁC PHÉP TOÁN
- CÀI ĐẶT
 - CÀI ĐẶT BẰNG DANH SÁCH LIÊN KẾT
 - CÀI ĐẶT BẰNG MẢNG



ĐỊNH NGHĨA

Ngăn xếp:

- Là một **dạng danh sách đặc biệt** mà việc thêm (Push) hay xóa (Pop) một phần tử chỉ thực hiện tại một đầu gọi là **đỉnh (TOP)** của ngăn xếp
- Việc thêm và xóa một phần tử được thực hiện theo dạng **LIFO** (Last In First Out) hay **FILO** (First In Last Out)





NGĂN XẾP (STACK)

- ĐỊNH NGHĨA
- CÁC PHÉP TOÁN
- CÀI ĐẶT
 - CÀI ĐẶT BẰNG DANH SÁCH LIÊN KẾT
 - CÀI ĐẶT BẰNG MẢNG



CÁC PHÉP TOÁN

Phép toán	Diễn giải
<code>makenullStack(S)</code>	Tạo một ngăn xếp rỗng (S)
<code>emptyStack(S)</code>	Kiểm tra xem ngăn xếp S có rỗng hay không. Hàm cho kết quả 1 (true) nếu ngăn xếp rỗng và 0 (false) trong trường hợp ngược lại.
<code>full(S)</code>	Kiểm tra xem ngăn xếp S có đầy hay không
<code>push(x,S)</code>	Thêm phần tử x vào đỉnh ngăn xếp S. Tương đương: <code>insertList(x,first(S),S)</code>
<code>pop(S)</code>	Xóa phần tử tại đỉnh ngăn xếp S. Tương đương: <code>deleteList(first(S),S)</code>
<code>top(S)</code>	Trả về phần tử đầu tiên trên đỉnh ngăn xếp S, tương đương: <code>retrieve(first(S),S)</code>



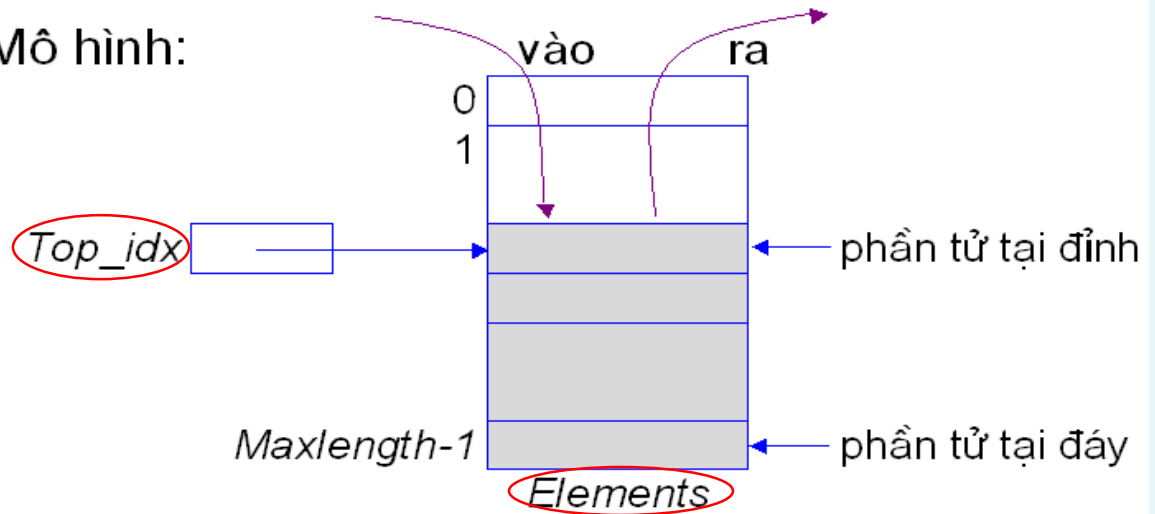
NGĂN XẾP (STACK)

- ĐỊNH NGHĨA
- CÁC PHÉP TOÁN
- CÀI ĐẶT
 - CÀI ĐẶT BẰNG MẢNG
 - CÀI ĐẶT BẰNG DANH SÁCH LIÊN KẾT



CÀI ĐẶT NGĂN XẾP BẰNG MẢNG

Mô hình:



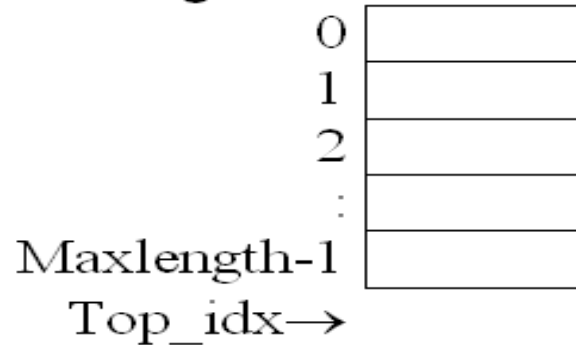
- Khai báo

```
#define MaxLength ...//độ dài của mảng
typedef ... ElementType;//kiểu phần tử của ngăn xếp
typedef struct {
    //Lưu nội dung của các phần tử
    ElementType Elements[MaxLength];
    int Top_idx; //giữ vị trí đỉnh ngăn xếp
}Stack;
```



KHỞI TẠO NGĂN XẾP RỖNG

S rỗng :



- Khi ngăn xếp S rỗng ta cho đỉnh ngăn xếp được khởi tạo bằng Maxlength

```
void makenullStack(Stack *pS) {  
    pS->Top_idx=MaxLength;  
}
```



KIỂM TRA NGĂN XẾP RỒI?

- Ta kiểm tra xem đỉnh ngăn xếp có bằng MaxLength không?

```
int emptyStack(Stack S) {  
    return S.Top_idx==MaxLength;  
}
```



KIỂM TRA NGẪN XẾP ĐẦY?

S đây :

Top_idx → 0

1

2

:

Maxlength-1



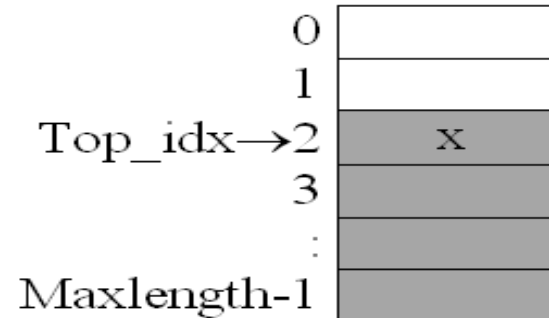
- Ta kiểm tra xem Top_idx có chỉ vào 0 hay không?

```
int full(Stack S) {  
    return S.Top_idx==0;  
}
```



TRẢ VỀ PHẦN TỬ ĐẦU NGĂN XẾP

Ví dụ :



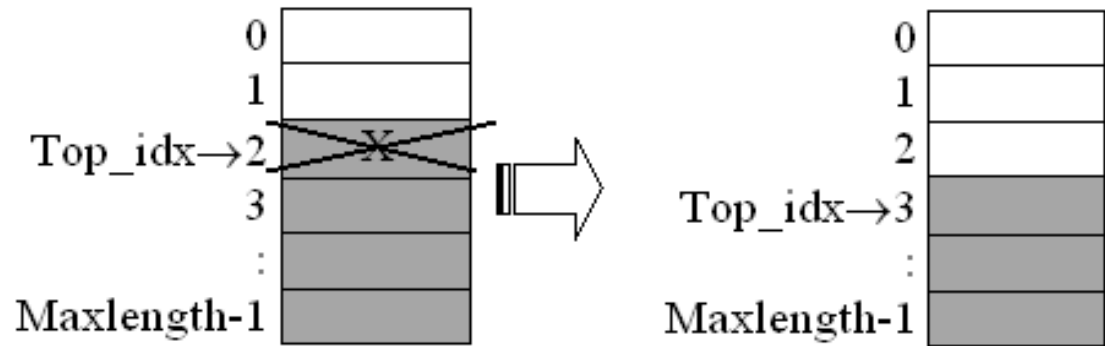
- Giải thuật: Kết quả của phép toán trên ngăn xếp là x
 - Nếu ngăn xếp rỗng thì thông báo lỗi
 - Ngược lại, trả về giá trị được lưu trữ tại ô có chỉ số là Top_idx

```
ElementType top(Stack S) {  
    if (!emptyStack(S))  
        return S.Elements[S.Top_idx];  
    else printf("Loi! Ngan xep rong");  
}
```



XÓA PHẦN TỬ TẠI ĐỈNH NGĂN XẾP

Ví dụ :



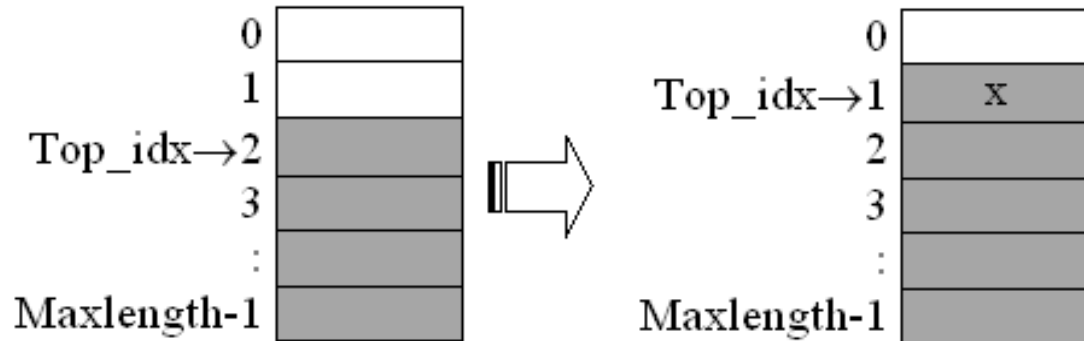
- Giải thuật:
 - Nếu ngăn xếp rỗng thì thông báo lỗi
 - Ngược lại, tăng Top_idx lên 1 đơn vị

```
void pop(Stack *pS) {  
    if (!emptyStack(*pS))  
        pS->Top_idx=pS->Top_idx+1;  
    else printf("Loi! Ngan xep rong!");  
}
```



THÊM PHẦN TỬ x VÀO NGĂN XẾP

Ví dụ :



Giải thuật :

- Nếu ngăn xếp đầy thì thông báo lỗi
- Ngược lại, giảm Top_idx xuống 1 đơn vị rồi đưa giá trị x vào ô có chỉ số Top_idx

```
void push(ElementType x, Stack *pS) {  
    if (fullStack(*pS))  
        printf("Loi! Ngan xep day!");  
    else{  
        pS->Top_idx=pS->Top_idx-1;  
        pS->Elements[pS->Top_idx]=x; }  
}
```



BÀI TẬP

- Viết hàm đổi số nguyên n sang số nhị phân (có sử dụng các phép toán trên ngăn xếp).
- Vận dụng các phép toán trên ngăn xếp để viết chương trình nhập vào n số nguyên và hiển thị kết quả vừa nhập ra màn hình.



NGĂN XẾP (STACK)

- ĐỊNH NGHĨA
- CÁC PHÉP TOÁN
- CÀI ĐẶT
 - CÀI ĐẶT BẰNG MẢNG
 - CÀI ĐẶT BẰNG DANH SÁCH LIÊN KẾT



CÀI ĐẶT NGĂN XẾP BẰNG DSLK

- Khai báo

```
typedef List Stack;
```

- Tạo ngăn xếp rỗng

```
void makenullStack(Stack *pS) {  
    makenullList(pS);  
}
```

- Kiểm tra ngăn xếp rỗng

```
int emptyStack(Stack S) {  
    return emptyList(S);  
}
```



CÀI ĐẶT NGĂN XẾP BẰNG DSLK

- Thêm phần tử vào ngăn xếp

```
void push(Elementtype x, Stack *pS) {  
    insertList(x, first(*pS), pS);  
}
```

- Xóa phần tử ra khỏi ngăn xếp

```
void pop(Stack *pS) {  
    deleteList(first(*pS), pS);  
}
```



CÀI ĐẶT NGĂN XẾP BẰNG DSLK

- Trả về phần tử ở đỉnh ngăn xếp

```
void pop(Stack S) {  
    retrieve(first(S), S);  
}
```



CANTHO UNIVERSITY

Q&A?