BÀI 3

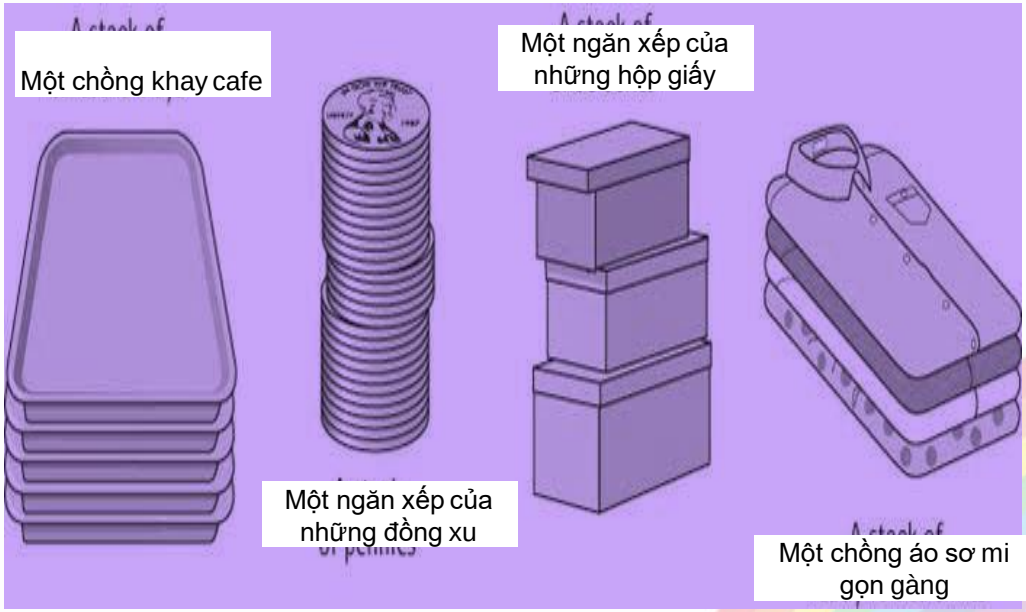
NGĂN XẾP

MỤC TIÊU

- Mô tả đúng về khái niệm của ngăn xếp và phân biệt ngăn xếp với danh sách;
- Trình bày các đặc tả ngăn xếp một cách chính xác;
- Mô tả các phương án cài đặt ngăn xếp bằng mảng và bằng danh sách liên kết;
- Sử dụng cấu trúc dữ liệu ngăn xếp để giải quyết các bài toán trong thực tế.

NỘI DUNG

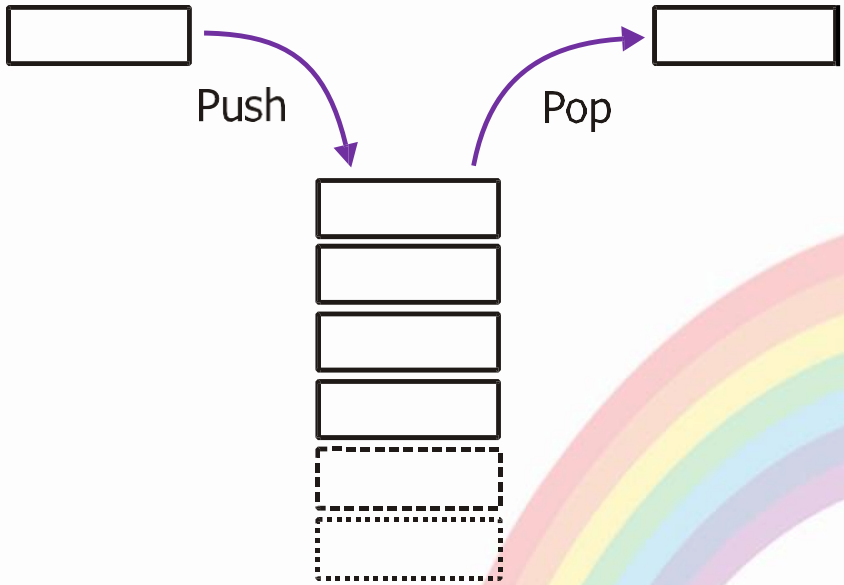
- Khái niệm ngăn xếp;
- Đặc tả ngăn xếp;
- Các phương án cài đặt ngăn xếp;
- Ứng dụng của ngăn xếp.



1. KHÁI NIỆM NGĂN XẾP

1. KHÁI NIỆM NGĂN XẾP

Ngăn xếp là một dạng đặc biệt của danh sách mà việc bổ sung (push) hay loại bỏ (pop) một phần tử đều được thực hiện ở một đầu của danh sách gọi là đỉnh (Nguyên tắc LIFO - Last In First Out).



Hình 3.1. Lược đồ ngăn xếp

2. ĐẶC TẢ NGĂN XẾP

2. ĐẶC TẢ NGĂN XẾP

- Đặc tả dữ liệu:
  - Có nhiều nút cùng một kiểu;
  - Có đỉnh stack (top);
- Đặc tả các tác vụ trên stack:
  - Initialize:
    - Chức năng: Khởi động stack;
    - Dữ liệu nhập: Không;
    - Dữ liệu xuất: stack top về vị trí khởi đầu.
  - Empty:
    - Chức năng kiểm tra stack có bị rỗng không;
    - Dữ liệu nhập: Không;
    - Dữ liệu xuất: True or False (True: khi stack rỗng, False: stack không bị rỗng).

2. ĐẶC TẢ NGĂN XẾP

- Push:
  - Chức năng: Thêm nút mới tại đỉnh stack;
  - Dữ liệu nhập: Nút mới;
  - Dữ liệu xuất: Không.
- Pop:
  - Chức năng: Xóa nút tại đỉnh stack;
  - Dữ liệu nhập: Không;
  - Điều kiện: Stack không bị rỗng;
  - Dữ liệu xuất: Nút bị xóa.
- Stacktop:
    - Chức năng: Truy xuất nút tại đỉnh stack;
    - Dữ liệu nhập: Không;
    - Điều kiện: Stack không bị rỗng;
    - Dữ liệu xuất: Nút tại đỉnh stack.

2. ĐẶC TẢ NGĂN XẾP

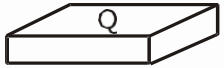
- Stacksize:
  - Chức năng: Xác định số nút hiện có trong stack;
  - Dữ liệu: Không;
  - Dữ liệu xuất: Số nút hiện có trong stack.
- Clearstack:
  - Chức năng: Xóa tất cả các nút ở trong stack;
  - Dữ liệu nhập: Không;
  - Dữ liệu xuất: Stack top về vị trí khởi đầu.
- Copystack:
  - Chức năng: Copy stack thành stack mới;
  - Dữ liệu nhập: Stack nguồn;
  - Dữ liệu xuất: Stack đích giống stack nguồn.
- Che dấu thông tin;
- Tính khả thi và hiệu quả của ứng dụng;
- Tính trong sáng của chương trình;
- Thiết kế từ trên xuống.

3. CÀI ĐẶT NGĂN XẾP

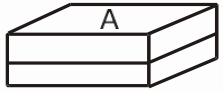
3. CÀI ĐẶT NGĂN XẾP

- Cài đặt ngăn xếp bằng mảng;
- Cài đặt ngăn xếp bằng danh sách liên kết.

Bổ sung hộp Q vào ngăn xếp trống:



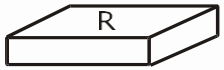
Bổ sung hộp A vào ngăn xếp:



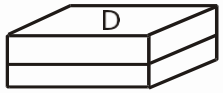
Lấy 1 hộp ra khỏi ngăn xếp:

Lấy 1 hộp ra khỏi ngăn xếp: (Trống)

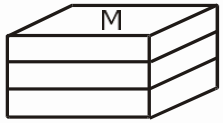
Bổ sung hộp R vào ngăn xếp:



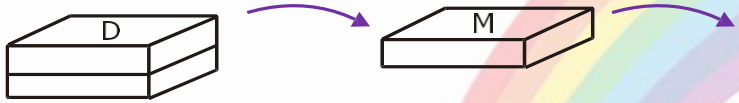
Bổ sung hộp D vào ngăn xếp:



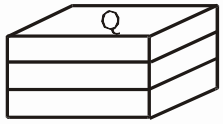
Bổ sung hộp M vào ngăn xếp:



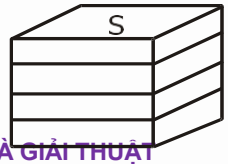
Lấy 1 hộp ra khỏi ngăn xếp:



Bổ sung hộp Q vào ngăn xếp:

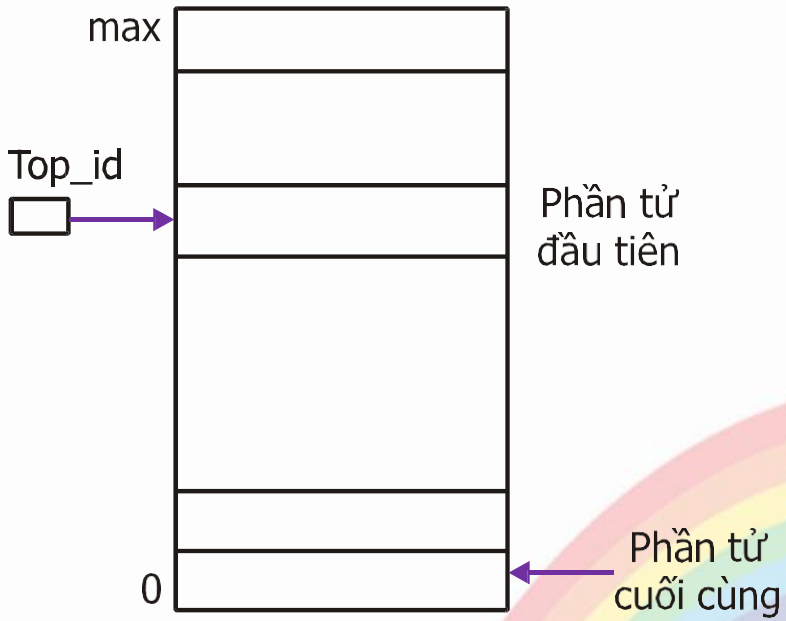


Bổ sung hộp S vào ngăn xếp:



3.1. CÀI ĐẶT NGĂN XẾP BẰNG MẢNG

- Ta sử dụng một mảng 1 chiều s để biểu diễn ngăn xếp;
- Thiết lập phần tử đầu tiên của mảng, s[0], làm đáy ngăn xếp;
- Đỉnh hiện tại của ngăn xếp: Biến số nguyên top_id;
  - Nếu ngăn xếp có n phần tử thì top sẽ có giá trị bằng n – 1;
  - Còn khi ngăn xếp chưa có phần tử nào (ngăn xếp rỗng) thì ta quy ước top_id sẽ có giá trị – 1.



Hình 3.3. Cài đặt ngăn xếp bằng mảng

3.1. CÀI ĐẶT NGĂN XẾP BẰNG MẢNG

Cài đặt ngăn xếp bằng mảng qua khai báo dưới đây:

```
#define     max …//khái báo độ lớn cực đại trong ngăn xếp
typedef     <kiểu dữ liệu> ElementType          struct Stack
   {
              int Top_id;
              ElementType Element[max];
    };
   Stack S;
```

3.1. CÀI ĐẶT NGĂN XẾP BẰNG MẢNG

Cài đặt các thao tác trên stack:

- Thao tác 1: Khởi tạo Stack

```
void StackInit(Stack *S);//khoi tao stack
    {
        S->Top_id=-1;
    }
```

- Thao tác 2: Kiểm tra stack có rỗng không:

```
int StackEmpty (Stack S);//kiem tra stack
    {
          return(S.Top_id==-1);
    }
```

3.1. CÀI ĐẶT NGĂN XẾP BẰNG MẢNG

Thao tác 3: Bổ sung thêm phần tử vào stack (bổ sung phần tử X vào stack, cài đặt bởi stack S mà Top_id đang trỏ tới đỉnh):

```
void PUSH(Stack *S,ElementType x)
//Kiểm tra xem stack đã đầy chưa
if (S->Top_id==max-1)
{    printf("Strack tran");
     return;          }
            //Nếu stack chưa đầy, thì di chuyển top_ip lên 1
S->Top_id ++;                                        [S->Top_id--];
//Nạp phần tử mới vào stack  S-
>Element[S->Top_id]=x;
//Thoát khỏi chương trình
return;
```

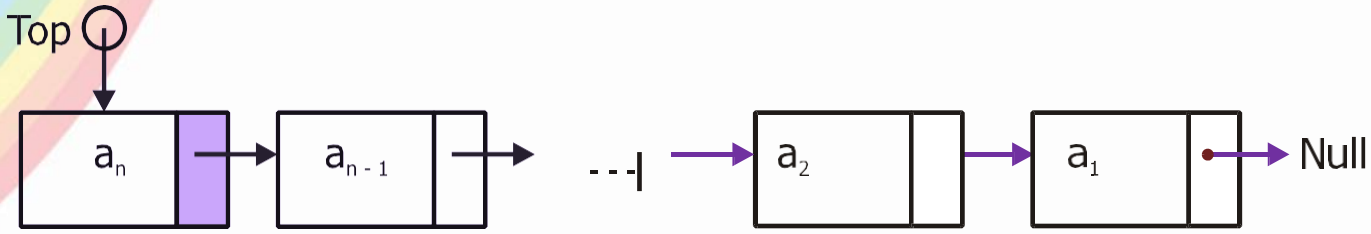
3.1. CÀI ĐẶT NGĂN XẾP BẰNG MẢNG

Thao tác 4: Lấy một phần tử khỏi đỉnh Stack:

```
ElementType POP(Stack *S)
{       if (StackEmpty(*S))
            {           printf("ngan xep rong");                    }

     else
            {
                    Return * S.Element[S->Top_id--]; }
}
```

3.2. CÀI ĐẶT NGĂN XẾP BẰNG DANH SÁCH LIÊN KẾT



- Đỉnh của stack là đầu của danh sách liên kết;
- Ta sử dụng con trỏ Top trỏ đến đỉnh stack;
- Hình dưới đây minh hoạ danh sách liên kết biểu diễn stack (a_1, a_2,..., a_n) với đỉnh là a_n.

3.2. CÀI ĐẶT NGĂN XẾP BẰNG DANH SÁCH LIÊN KẾT

Khai báo một ngăn xếp bằng danh sách liên kết:

```
typedef <kieu du lieu> ElementType
struct StackNode
{                    ElementType Data;
                struct StackNode *Next;
};
typedef struct
{
                              StackNode *Top;
}Stack;
```

3.2. CÀI ĐẶT NGĂN XẾP BẰNG DANH SÁCH LIÊN KẾT

Các thao tác trên ngăn xếp được cài đặt bằng danh sách liên kết:

- Thao tác khởi tạo ngăn xếp:

```
void StackInit(Stack *S)
    {
            S->Top=NULL;
            return;
     }
```

- Thao tác xác định điều kiện rỗng của ngăn xếp:

```
int StackEmpty(Stack *S)
{
        return (S->Top==NULL);
}
```

```
p=S->Top;
S->Top=S->Top->Next;
```

3.2. CÀI ĐẶT NGĂN XẾP BẰNG DANH SÁCH LIÊN KẾT

Thao tác thêm một phần tử vào danh sách:

```
void PUSH(Stack *S,ElementType x)
{    StackNode *p
     p=(StackNode*) malloc (sizeof(struct StackNode));
     (*p).Data =x;
          p->Next=S->Top ;
          S->Top=p;
          return;
}
```

3.2. CÀI ĐẶT NGĂN XẾP BẰNG DANH SÁCH LIÊN KẾT

- Thao tác lấy một phần tử khỏi đỉnh của ngăn xếp S:

```
ElementType POP(Stack *S)
    { StackNode*p
            if(StackEmpty(S))
              {     printf("ngan xep rong");  return
              NULL;                          }
            else
            {           p=S->Top;
                        S->Top=S->Top->Next;
                  return p->Data;                }
     }
```

4. ỨNG DỤNG CỦA NGĂN XẾP

4. ỨNG DỤNG CỦA NGĂN XẾP

Ứng dụng ngăn xếp trong tính toán giá trị của biểu thức (Ký pháp nghịch đảo Balan).

- Trong đa số các ngôn ngữ lập trình, các biểu thức được biểu diễn như trên gọi là kí pháp trung tố (infix);
- Nên khi xác định giá trị của một biểu thức số học ta đưa ra thuật toán sau. Thuật toán này gồm hai giai đoạn:
  - Chuyển biểu thức số học thông thường (dạng trung tố – infix) sang biểu thức số học dạng hậu tố (postfix – dạng ký pháp nghịch đảo Balan gọi tắt là biểu thức Balan (phép toán được đặt sau các toán hạng));
  - Tính giá trị của biểu thức số học Balan postfix.

4. ỨNG DỤNG CỦA NGĂN XẾP

Ví dụ 3.2:

Biểu thức thông thường (trung tố)	Biểu thức Balan
a * b/ c	ab * c /
a * (b + c) - d/e	abc + * de / -

4. ỨNG DỤNG CỦA NGĂN XẾP

Ví dụ 3.3: Xét biểu thức sau ở dạng trung tố: (1 + 3) * (5 – (6 – 4)). Biểu thức được biểu diễn ở dạng biểu thức balan là: **1 3 + 5 6 4 - - *** Ta có bảng kết quả với thuật toán tính giá trị của biểu thức số học Balan:

Đọc	Xử lý	Stack
1	Đẩy vào Stack	1
3	Đẩy vào Stack	3
+	Lấy 3 và 1 ra khỏi Stack và tính được 3 + 1 = 4, đẩy 4 vào Stack	4
5	Đẩy vào Stack	4, 5
6	Đẩy vào Stack	4, 5, 6
4	Đẩy vào Stack	4, 5, 6, 4
–	Lấy 4 và 6 ra khỏi Stack và tính được 6 – 4 = 2, đẩy 2 vào Stack	4, 5, 2
–	Lấy 2 và 5 ra khỏi Stack và tính được 5 – 2 = 3, đẩy 3 vào Stack	4, 3
*	Lấy 3 và 4 ra khỏi Stack và tính được 4 * 3 = 12, đẩy 12 vào Stack	12

4. ỨNG DỤNG CỦA NGĂN XẾP

Ví dụ: Xét biểu thức: E = a * (b + c) – d # (Dấu # báo kết thúc biểu thức trung tố)

Đọc biểu thức trung tố	Xử lý	Stack	Biểu thức Balan
	Khởi động ngăn xếp rỗng	$	
a	Hiển thị a	$	a
*	Đẩy toán tử * vào ngăn xếp	$ *	a
(	Đẩy vào ngăn xếp	$, *, (	a
b	Hiển thị b	$, *, (	ab
+	Xét thấy Fri('+') > Fri('('), đẩy vào ngăn xếp	$, *, (, +	ab
c	Hiển thị c	$, *, (, +	abc
)	Lấy toán tử + ra khỏi ngăn xếp và hiển thị	$, *, (	abc+
	Lấy (ra khỏi ngăn xếp	$, *	abc+
–	Xét thấy Fri('–') > Fri('*'), lấy toán tử * ra khỏi ngăn xếp và hiển thị	$	abc+*
	Đẩy toán tử – ngăn xếp	$, –	abc+*
d	Hiển thị d	$, –	abc+*d
#	Lấy toán tử – ra khỏi ngăn xếp và hiển thị	$	abc+*d–

4. ỨNG DỤNG CỦA NGĂN XẾP

Ứng dụng ngăn xếp để loại bỏ đệ quy của chương trình:

- Bước 1: Lưu các biến cục bộ và địa chỉ trở về;
- Bước 2: Nếu thoả điều kiện ngừng đệ quy thì chuyển sang bước 3; Nếu không thì tính toán từng phần và quay lại bước 1 (đệ quy tiếp);
- Bước 3: Khôi phục lại các biến cục bộ và địa chỉ trở về.

4. ỨNG DỤNG CỦA NGĂN XẾP

Thiết kế thuật toán chuyển biểu thức số học thông thường sang biểu thức số học Balan.

- Sử dụng stack S để lưu các dấu mở ngoặc trái và các dấu phép toán + , -, * và /.
- Đưa vào ký hiệu $ để đánh dấu đáy của stack. Khi đỉnh stack chứa $, có nghĩa là stack rỗng.
- Xây dựng một hàm Pri để xác định độ ưu tiên của các phép toán và các kí hiệu $, hàm Pri xác định độ ưu tiên như sau : Pri (‘$’) < Pri (‘(‘) < Pri (‘+’) = Pri (‘-‘) < Pri (‘*’) = Pri(‘/’).

TÓM LƯỢC CUỐI BÀI

- Trình bày đúng khái niệm về ngăn xếp. Phân biệt điểm giống và khác giữa ngăn xếp và danh sách;
- Mô tả đúng các đặc tả về ngăn xếp và các phương án cài đặt cho ngăn xếp;
- Trình bày được các ứng dụng của ngăn xếp và vận dụng ngăn xếp vào giải quyết các bài toán.