



Đệ Quy Trong C++

(RECURSION)

Cấu trúc dữ liệu ngăn xếp.

1. Ngăn xếp(stack) là một cấu trúc dữ liệu quan hệ mật thiết với cơ chế hoạt động của đệ quy.

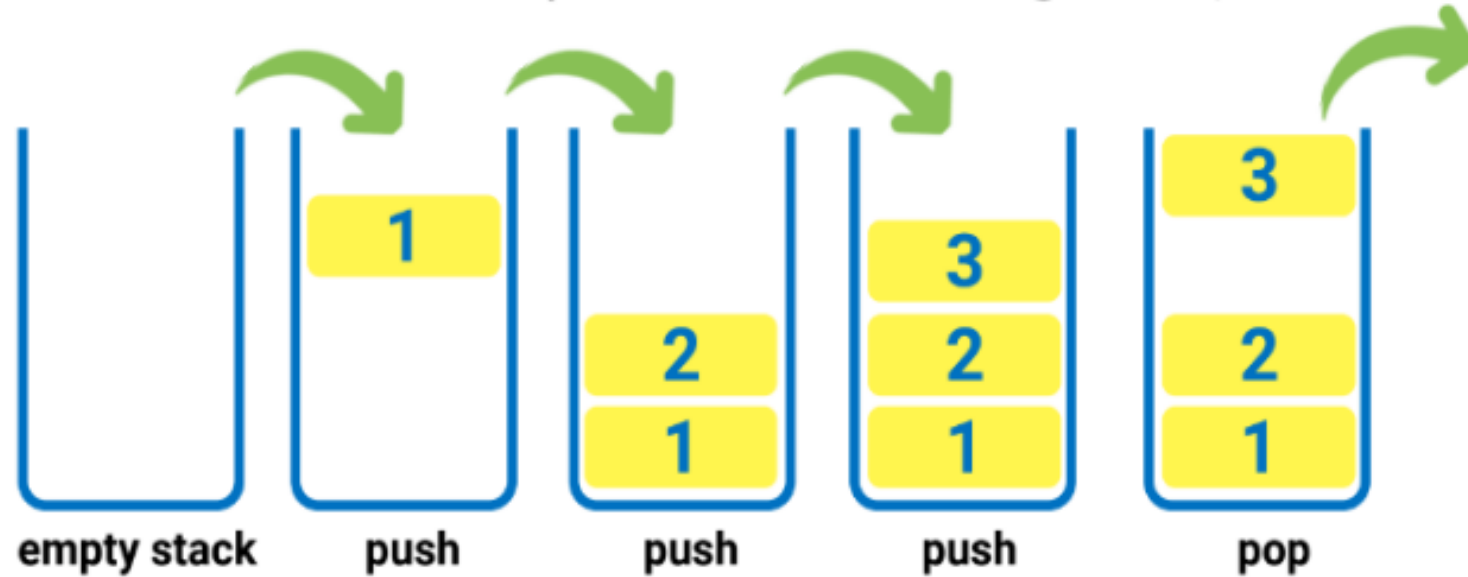


Ngăn xếp là một cấu trúc dữ liệu hỗ trợ **2 thao tác push và pop**. Trong đó push giúp thêm 1 phần tử vào đỉnh ngăn xếp, pop giúp xóa 1 phần tử khỏi đỉnh ngăn xếp. Cả 2 thao tác này đều **được thực hiện ở đỉnh ngăn xếp**.

1. Cấu trúc dữ liệu ngăn xếp



Ngăn xếp hoạt động theo nguyên tắc viết tắt là **LIFO (Last In First Out)** nghĩa là vào cuối thì ra đầu. Các phần tử vào cuối cùng sẽ được ra đầu tiên



Trong chương trình tồn tại một bộ nhớ là **bộ nhớ ngăn xếp**, cách hoạt động của bộ nhớ này **tương tự** như cách hoạt động của cấu trúc dữ liệu ngăn xếp.



1

Hàm đệ quy.

Đệ Quy Là Gì?

Đệ quy có nghĩa là một hàm tự gọi lại chính nó.

```
void recurse(){
```

```
    .. .. .  
    recurse();
```

```
    .. .. .  
}
```

```
int main(){
```

```
    .. .. .  
    recurse();
```

```
    .. .. .  
}
```

recursive call





2

Thành phần của một hàm đệ quy.

Thành Phần Của Đệ Quy

Phần cơ sở: Điều kiện để thoát khỏi đệ quy. Nếu như không có phần này, hàm đệ quy sẽ thực hiện mãi mãi gây ra tràn bộ nhớ Stack.

Phần đệ quy: Thân hàm có chứa phần gọi đệ quy, thực hiện cho đến khi thỏa mãn điều kiện ở phần cơ sở.



Đệ quy thường dựa trên công thức toán học gọi là **công thức truy hồi** và **một bài toán con nhỏ nhất**. Khi viết hàm đệ quy ta **cần xác định** được bài toán con nhỏ nhất để **làm điểm dừng** cho hàm đệ quy và công thức truy hồi để **tìm ra lời giải** của bài toán lớn hơn thông qua đáp án của bài toán nhỏ hơn.

Một Số Ví Dụ Minh Họa

1. Tính tổng $S(n) = 1 + 2 + 3 + 4 + 5 + \dots + n$

```
int sum(int n){  
    if (n == 0)  
        return n;  
    else  
        return n + sum(n - 1);  
}
```

2. Tính giai thừa $F(n) = 1.2.3.4.5....n$

```
int F(int n){  
    if (n == 0)  
        return 1;  
    else  
        return n * F(n - 1);  
}
```




THANKS

Full House Tất Cả Là Một Nhà!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.