

# Cấu trúc dữ liệu - Stack (Ngăn xếp)

uit.edu.vn

Ngày 27 tháng 3 năm 2023

## Minh họa Stack

LIFO: Last In First Out: Vào sau ra trước

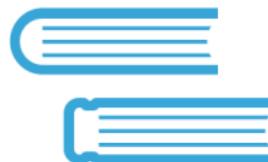
- **push()**



## Minh họa Stack

LIFO: Last In First Out: Vào sau ra trước

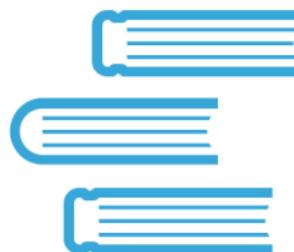
- **push()**



## Minh họa Stack

LIFO: Last In First Out: Vào sau ra trước

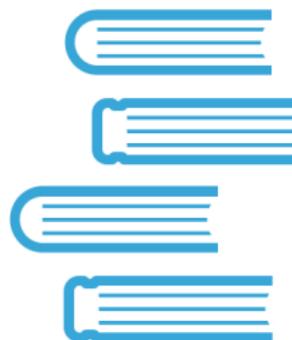
- **push()**



## Minh họa Stack

LIFO: Last In First Out: Vào sau ra trước

- **push()**



## Minh họa Stack

LIFO: Last In First Out: Vào sau ra trước

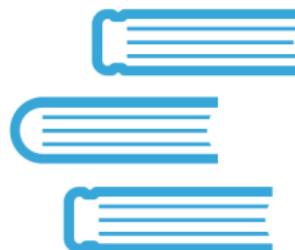
- **pop()**



## Minh họa Stack

LIFO: Last In First Out: Vào sau ra trước

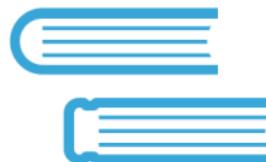
- **pop()**



# Minh họa Stack

LIFO: Last In First Out: Vào sau ra trước

- **pop()**



# Minh họa Stack

LIFO: Last In First Out: Vào sau ra trước

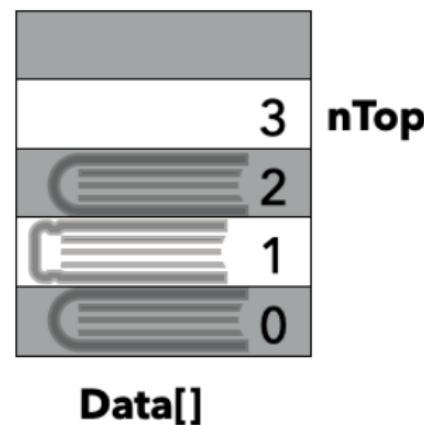
- **pop()**



# Cài đặt lớp Stack

Khai báo 1 lớp Stack có các hàm sau:

- `empty()`: kiểm tra ngăn xếp có rỗng không?
- `size()`: lấy kích thước của ngăn xếp
- `push()`: thêm 1 phần tử vào ngăn xếp
- `pop()`: Xoá 1 phần tử ra khỏi ngăn xếp
- `top()`: Chỉ đến phần tử đầu của ngăn xếp

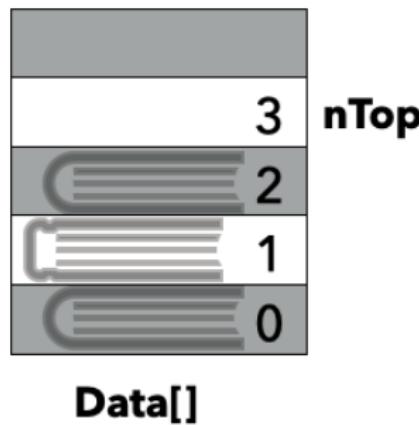


## Cài đặt lớp Stack

Khai báo 1 lớp Stack có các hàm sau:

- empty(): kiểm tra ngăn xếp có rỗng không?
- size(): lấy kích thước của ngăn xếp
- push(): thêm 1 phần tử vào ngăn xếp
- pop(): Xoá 1 phần tử ra khỏi ngăn xếp
- top(): Chỉ đến phần tử đầu của ngăn xếp

```
class Stack
{
    int Data[MAX]
    int nTop
}
```



## Bài tập: Cài đặt lớp stack

```
1 template <typename _DataType>
2 class stack{
3 private:
4     _DataType Data[MAX];
5     int nTop;
6 private: // cài đặt các hàm bên dưới, hàm khởi tạo .
7     ? empty()
8     ? pop()
9     ? push
10    ? top()
11 };
12 int main() {
13     stack<int> s;
14     s.push(1);
15     s.push(2);
16     s.push(3);
17     cout<<s.top();s.pop();cout<<" "<<s.empty()<<endl; //3 false
18     cout<<s.top();s.pop();cout<<" "<<s.empty()<<endl; //2 false
19     s.push(4);
20     s.push(5);
21     cout<<s.top();s.pop();cout<<" "<<s.empty()<<endl; //5 false
22     cout<<s.top();s.pop();cout<<" "<<s.empty()<<endl; //4 false
23     cout<<s.top();s.pop();cout<<" "<<s.empty()<<endl; //1 true
24 }
```

## Đáp án - Cài đặt lớp stack

```
1 template <typename _DataType>
2 class stack{
3 private:
4     _DataType Data[MAX];
5     int nTop;
6 public:
7     stack(){
8         nTop = 0;}
9     bool empty(){
10        return (nTop ==0);}
11     void pop(){
12         nTop--;}
13     void push(_DataType _data){
14         Data[nTop] = _data;
15         nTop = nTop+1; }
16     _DataType top(){
17         return Data[nTop-1]; }
18 };
19 int main() {
20     stack<int> s;
21     s.push(1);
22     s.push(2);
```

## Thay lớp stack tự viết bằng lớp stack của STL

```
1  /*template <typename _DataType>
2   class stack{
3     private:
4       _DataType Data[MAX];
5       int nTop;
6     public:
7       stack() { nTop = 0; }
8       bool empty() { return (nTop == 0); }
9       void pop() { nTop--; }
10      void push(_DataType _data) {
11          Data[nTop] = _data;
12          nTop = nTop + 1; }
13      _DataType top() { return Data[nTop - 1]; }
14  } */
15 #include <stack> // Dùng thư viện stack có sẵn
16 int main() {
17     stack<int> s;
18     s.push(1);
19     s.push(2);
20     s.push(3);
21     cout << s.top(); s.pop(); cout << " " << s.empty() << endl; // 3 false
22     cout << s.top(); s.pop(); cout << " " << s.empty() << endl; // 2 false
```

# Mở rộng

- Khử đệ qui. Ví dụ bài toán duyệt cây dùng đệ qui và stack/queue
- Cấu trúc dữ liệu head, stack
- Trang web stackoverflow :D

# Bài tập

- ① Kiểm tra tính hợp lệ của chuỗi có dấu ngoặc. Link:  
[HackerRank](#) [LeetCode](#)
- ② Hình chữ nhật lớn nhất trong biểu đồ histogram. Link:  
[HackerRank](#) [LeetCode](#)
- ③ Giải nén chuỗi. Link [LeetCode](#)