

Stack – Ngăn xếp

Tín Nguyễn - <https://github.com/tinict>

1 Lý thuyết

Trong khoa học máy tính, một ngăn xếp (còn gọi là bộ xếp chồng, tiếng anh: Stack) là một cấu trúc dữ liệu trừu tượng hoạt động theo nguyên lý “vào sau ra trước” (Last In First Out) LIFO. Tức là phần tử cuối cùng được chèn vào ngăn xếp sẽ là phần tử đầu tiên được lấy ra khỏi ngăn xếp.

Một ví dụ trực quan: Bạn có một chồng sách và bạn để nó trong một cái hộp như hình phía dưới. Giả sử hộp này vừa khít các cuốn sách. Khi đó, bạn có các thao tác:

- Thêm một cuốn sách vào hộp(push của stack)
- Lấy một cuốn sách khỏi hộp, bạn chỉ lấy được thằng trên cùng(pop của stack)

Cấu trúc dữ liệu ngăn xếp bị giới hạn theo cách như trên. Như vậy, việc thao tác với ngăn xếp của chúng ta chỉ bao gồm các hành động sau:

- **Push:** Thêm một phần tử vào đỉnh của ngăn xếp, số phần tử của ngăn xếp tăng lên 1.
- **Pop:** Xóa bỏ phần tử đầu tiên ở đỉnh của ngăn xếp, số phần tử của ngăn xếp giảm đi 1.
- **Top:** Lấy giá trị của phần tử đầu tiên ở đỉnh của ngăn xếp, số phần tử của ngăn xếp không thay đổi.
- **IsEmpty:** Kiểm tra ngăn xếp trống hay không. Ngăn xếp trống là ngăn xếp không có phần tử nào.
- **IsFull:** Kiểm tra ngăn xếp đã đầy hay chưa. Thao tác này không phải lúc nào cũng có.
- **Size:** Lấy số lượng phần tử stack đang có.

2 Cài đặt ngăn xếp

a. Kiểm tra stack đầy (IsFull)

```
bool Stack::IsFull(int capacity)
{
    if(Size() == capacity)
        return true;
    return false;
}
```

b. Kiểm tra stack rỗng (IsEmpty)

```

bool Stack::IsEmpty()
{
    if(f == NULL)
        return false;
    return true;
}

```

c. Thêm 1 phần tử vào đỉnh của ngăn xếp (Push)

```

void Stack::Push(int x)
{
    if(f == NULL)
        f = createNode(x);
    else if(f != NULL)
    {
        node *temp = createNode(x);
        temp->next = f;
        f = temp;
    }
}

```

d. Lấy giá trị đầu tiên của đỉnh ngăn xếp (Top)

```

int Stack::Top()
{
    int t = f->data;
    f = f->next;
    return t;
}

```

e. Xóa bỏ phần tử đầu tiên của đỉnh ngăn xếp (Pop)

```

void Stack::Pop()
{
    node *p = f;
    node *temp = p->next;
    delete f;
    f = temp;
}

```

f. Đếm số lượng phần tử stack đang có (Size)

```

int Stack::Size()
{
    node *p = f;
    int count = 0;
}

```

```
while(p != NULL)
{
    count++;
    p = p->next;
}
return count;
}
```