Ngăn xếp (Stack)

Hàng đợi (Queue)

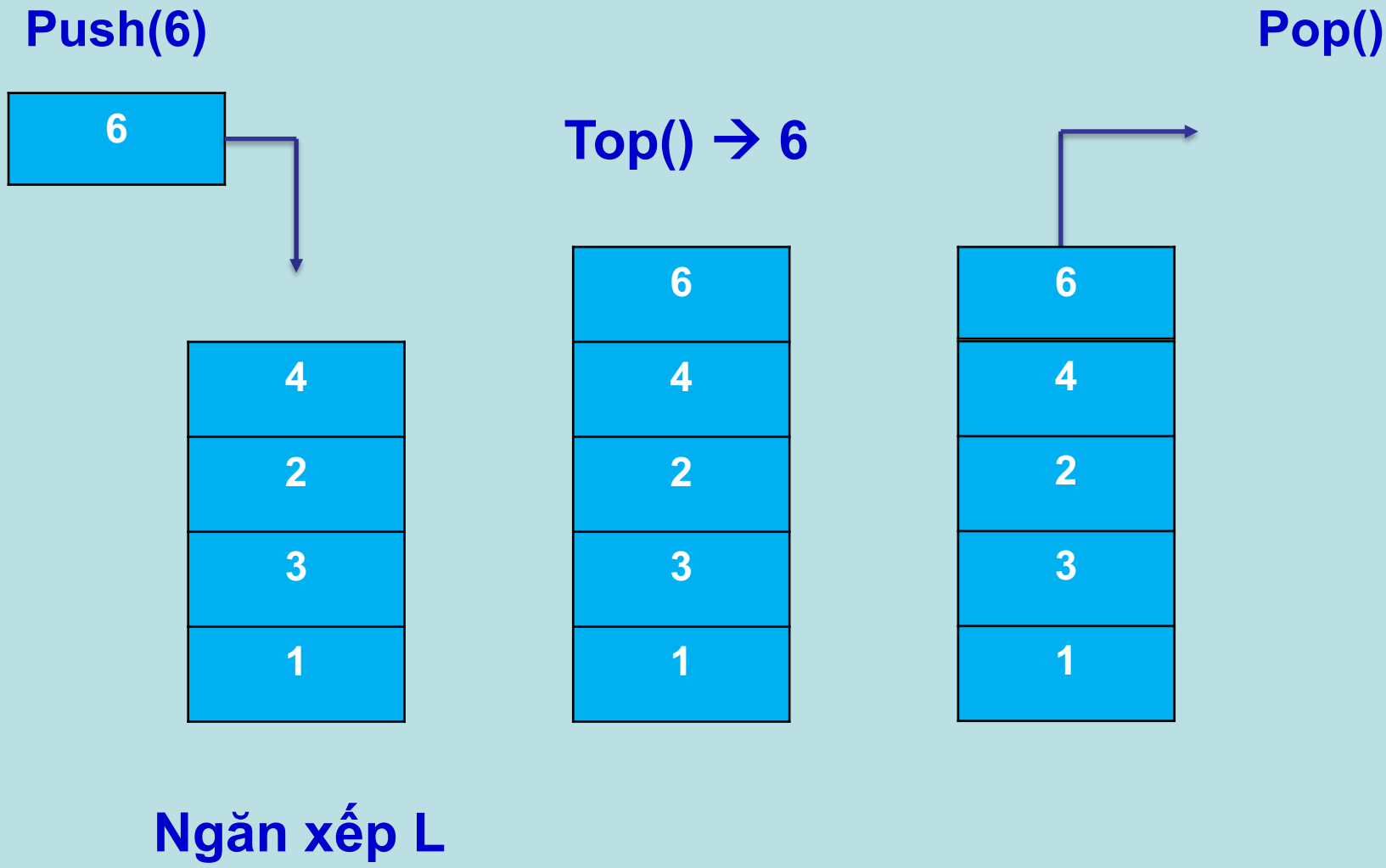
Ngăn xếp (Stack)

- **Stack (ngăn xếp):** là cấu trúc dữ liệu trừu tượng chứa các đối tượng làm việc theo cơ chế **LIFO** *(Last In First Out),* tức việc thêm 1 đối tượng vào Stack hoặc lấy 1 đối tượng ra khỏi Stack được thực hiện theo cơ chế "***vào sau ra trước".***
- **Stack** là một cấu trúc dạng thùng chứa các đối tượng, trong đó có 2 thao tác chính là : Push và Pop.

Các thao tác trên Stack

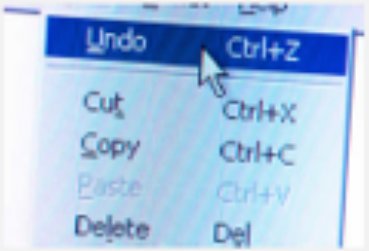
- **Push(p)**: Thêm đối tượng p vào Stack.
- **Pop()**: Lấy đối tượng từ Stack.
- **isEmpty()**: Kiểm tra Stack có rỗng hay không?
- **Top()**: Trả về giá trị của phần tử nằm đầu Stack mà không hủy nó khỏi Stack.

Ngăn xếp (Stack)



Ứng dụng của stack

- Máy ảo Java
- Parsing trong trình biên dịch
- Undo trong Word
- Nút Back trpng trình duyệt Web
- Cài đặt / thực hiện lời gọi hàm trong trình biên dịch
- …

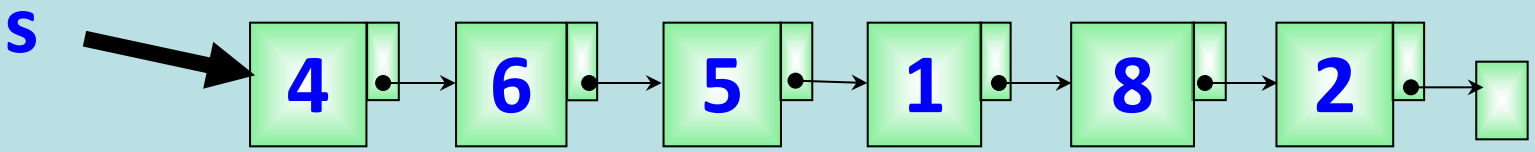


Cài đặt Stack

- Dùng mảng 1 chiều


```
Data S [N];
int  t;
```

- Dùng danh sách liên kết đơn



List s

- **Thêm và hủy cùng phía**

Cài Stack bằng DSLK đơn

- Kiểm tra tính rỗng của Stack

```
int IsEmpty(LIST s)
{
    if (s.pHead == NULL) //Stack rỗng
        return 1;
    else
        return 0;
}
```

Thêm 1 phần tử vào Stack

```
void Push(LIST &s, Node *Tam)
{  if (s.pHead == NULL)
   {    s.pHead = Tam;
        s.pTail = s.pHead;
   }
   else
   {    Tam->pNext = s.pHead;
        s.pHead = Tam;
   }
}
```

Lấy 1 phần tử từ Stack

```
int Pop(LIST &s, int &trave)
{   Node *p;
    if(IsEmpty(s) != 1)
    {
        if (s.pHead != NULL)
        {       p = s.pHead;
                trave = p->Info;
                s.pHead = s.pHead->Next;
                if (s.pHead == NULL)     s.Tail = NULL;
                return 1;
                delete p;
        }
    }
    return 0;
}
```

Hàng đợi (Queue)

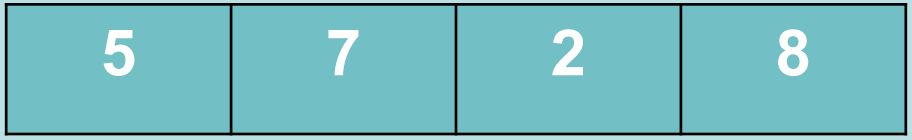
- **Queue (hàng đợi):** là 1 vật chứa các đối tượng làm việc theo cơ chế **FIFO** *(First In First Out)*, tức việc thêm 1 đối tượng vào Queue hay lấy 1 đối tượng ra khỏi Queue thực hiện theo cơ chế ***"vào trước ra trước".***
- **Queue** là một cấu trúc dạng ống có 2 đầu, dữ liệu vào ở một đầu và được lấy ra ở đầu kia. Thao tác chính trên queue là EnQueue và DeQueue: thêm vào và lấy đối tượng ra khỏi hàng đợi.

Các thao tác trên Queue

- **EnQueue(p)**: Thêm đối tượng p vào cuối hàng đợi.
- **DeQueue()**: Lấy đối tượng ở đầu hàng đợi
- **isEmpty()**: Kiểm tra hàng đợi có rỗng hay không?
- **Front()**: Trả về giá trị của phần tử nằm đầu hàng đợi mà không hủy nó.

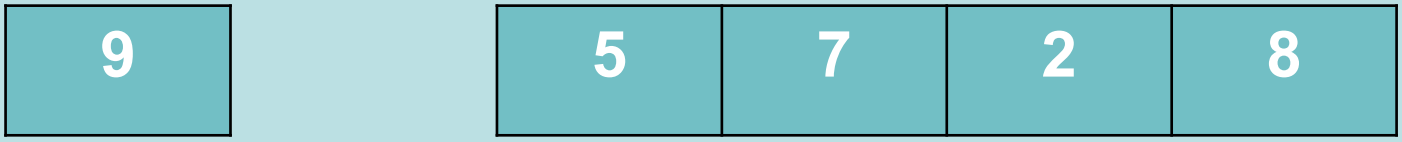
Các thao tác trên Queue

Hàng đợi Q



EnQueue(9)

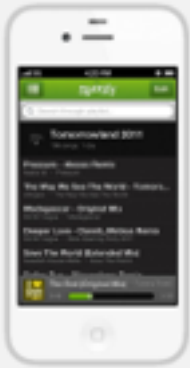
Front() → 8

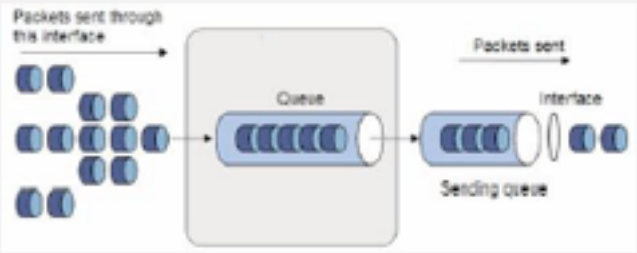


DeQueue()

Ứng dụng của queue

- Data buffer
- Hàng đợi xử lý trong các trung tâm chăm sóc khách hàng
- Xếp hàng thanh toán trong siêu thị
- Xếp hàng chơi trò chơi trong các khu vui chơi giải trí
- …



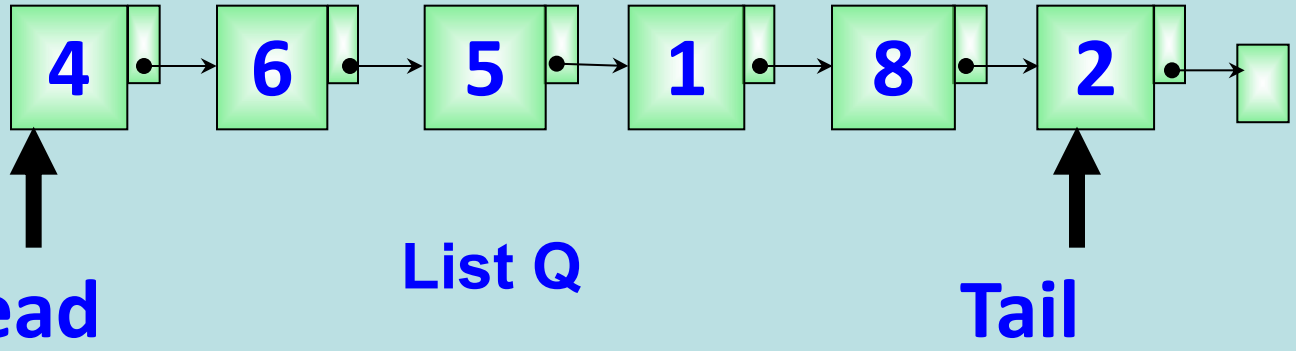


Cài đặt Queue

- Dùng mảng 1 chiều


```
Data   S [N];
int    f,r;
```

- Dùng danh sách liên kết đơn



*** Thêm và hủy khác phía**

Cài đặt Queue bằng List

- Kiểm tra Queue có rỗng?

```
int IsEmpty(LIST &Q)
{
        if (Q.pHead == NULL) //Queue rỗng
                return 1;
        else
                return 0;
}
```

Thêm 1 phần tử vào Queue

```
void EnQueue(LIST &Q, Node *Tam)
{  if (Q.pHead == NULL)
   {    Q.pHead = Tam;
        Q.pTail = Tam;
   }
   else
   {    Q.pTail->Next = tam;
        Q.pTail = tam;
   }
}
```

Lấy 1 phần tử từ Queue

```
int DeQueue(LIST &Q, int &trave)
{   Node *p;
    if (IsEmpty(Q) != 1)
        if (Q.pHead != NULL)
          {         p = Q.pHead;
                    trave = p->Info;
                    Q.pHead = Q.pHead->Next;
                    if(Q.pHead == NULL)    Q.pTail = NULL;
                    return 1;
                    delete p;
          }
     return 0;
}
```

Bài tập 1

1. Hãy cho biết nội dung của stack sau mỗi thao tác trong dãy:
EAS*Y**QUE***ST***I*ON

Với một kí tự tượng trưng cho thao tác thêm chữ cái tương ứng vào stack, dấu * tượng trưng cho thao tác lấy nội dung một phần tử trong stack in lên màn hình. Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình?

2. Cài đặt chương trình cho phép thực hiện các phép tính +,-,*,/ trên các số có tối đa 30 chữ số,

có chức năng nhớ (M+, M-, MC, MR).

3. Viết chương trình thực hiện các thao tác trên đa thức.

4.Hãy viết chương trình mô phỏng cho bài toán “Tháp Hà Nội” bằng cách sử dụng ngăn xếp.

5.Viết chương trình tìm tất cả các cặp dấu ngoặc tương ứng trong một chương trình viết bằng ngôn

ngữ C/C++.

Bài tập 2

Vấn đề : Cài đặt một queue sử dụng 2 stack ?

Ứng dụng :

- Job interview
- Cài đặt queue ưu tiên
- …