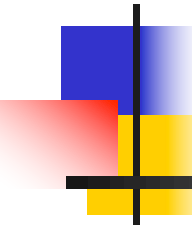
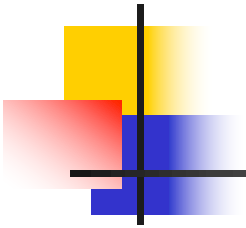


NGĂN XẾP VÀ HÀNG ĐỢI



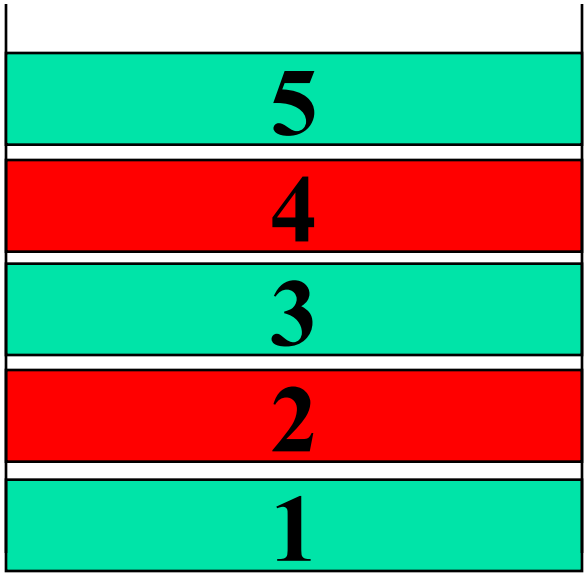
Nội dung bài học

- Một số danh sách tuyến tính đặc biệt:
  - Stack (ngăn xếp)
  - Queue (Hàng đợi)

1. Stack (ngăn xếp)

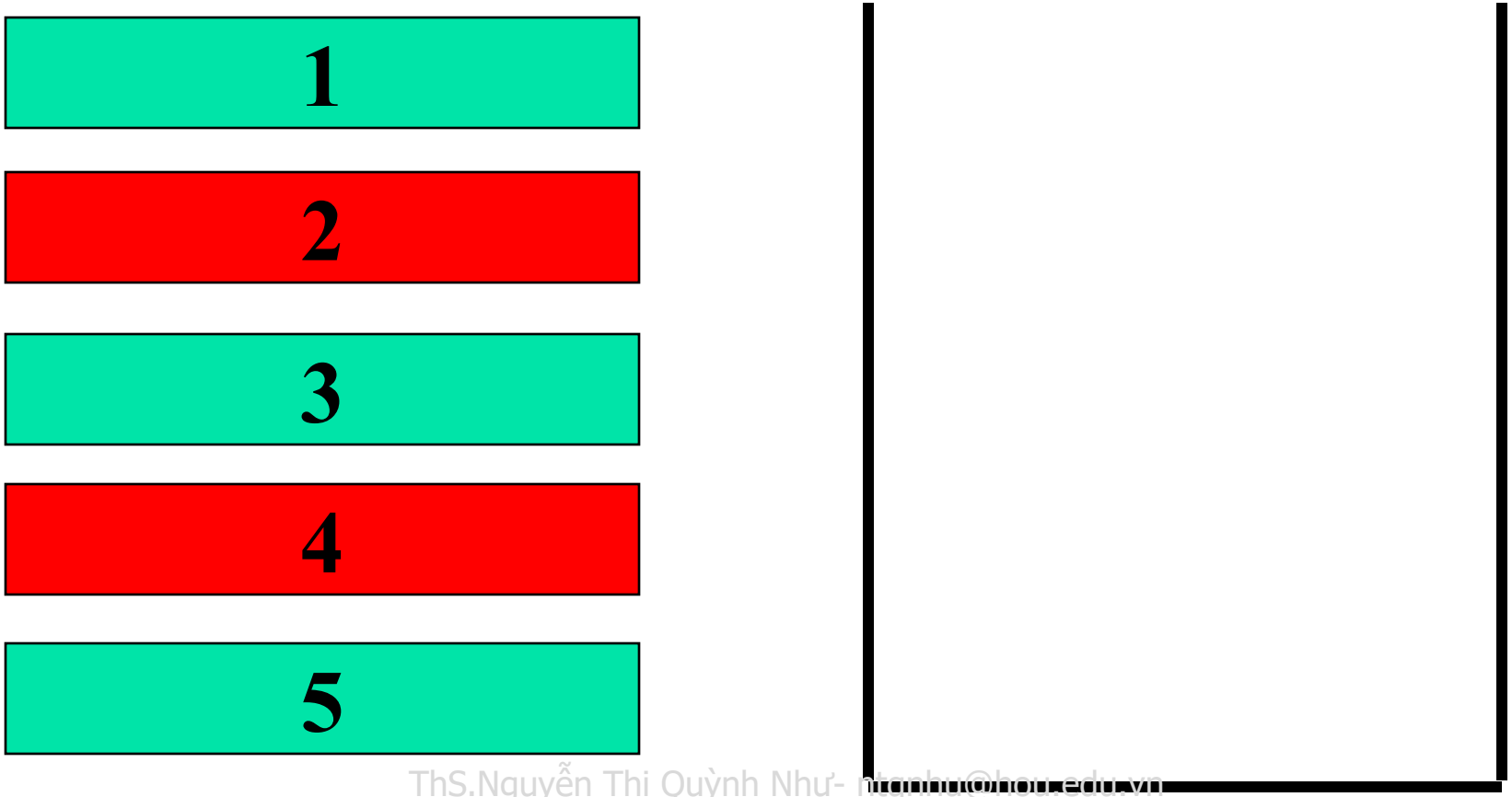
- ✓Ngăn xếp là một danh sách liên kết trong đó được trang bị hai phép toán bổ sung một phần tử vào cuối danh sách và loại bỏ một phần tử cũng ở cuối danh sách
- ✓Trong ngăn xếp một phần tử vào sau sẽ bị đẩy ra trước và phần tử vào trước sẽ bị đẩy ra sau ⇔ gọi là danh sách LIFO(*Last In First Out)*

Ví dụ về ngăn xếp:



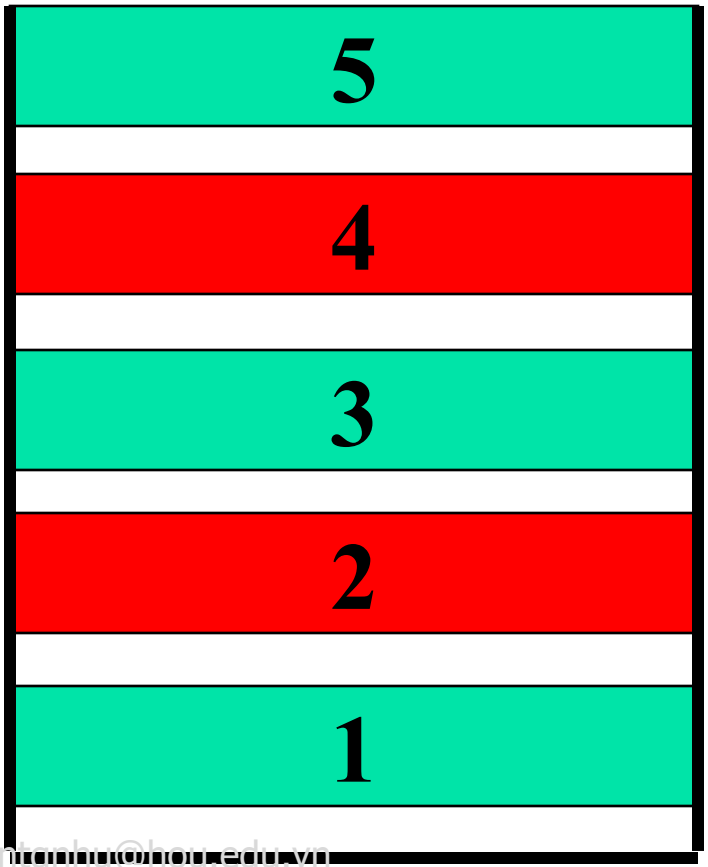
*Cơ chế làm việc của Stack

Bổ sung (nạp) phần tử vào Stack

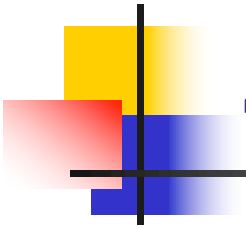


*Cơ chế làm việc của Stack

Hủy bỏ (lấy ra) phần tử từ Stack



Stack



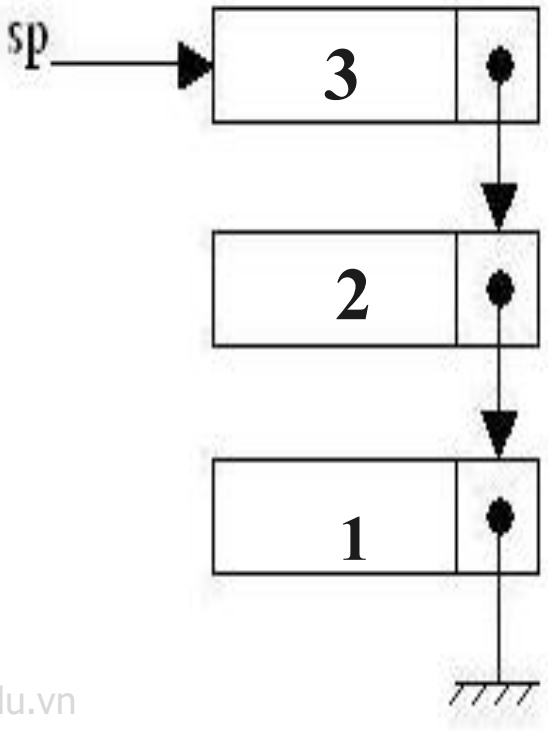
2. Biểu diễn Stack:

- Stack có thể biểu diễn bằng mảng hoặc danh sách liên kết.

Biểu diễn Stack bằng danh sách liên kết

- Stack là một danh sách liên kết đơn, tại mỗi thời điểm ta chỉ quan tâm nút đầu trong danh sách.
- Stack được định nghĩa như sau:

```
struct Node
{
    Data info;
    struct Node *next;
};
Node *sp;
```



Các thao tác trên Stack

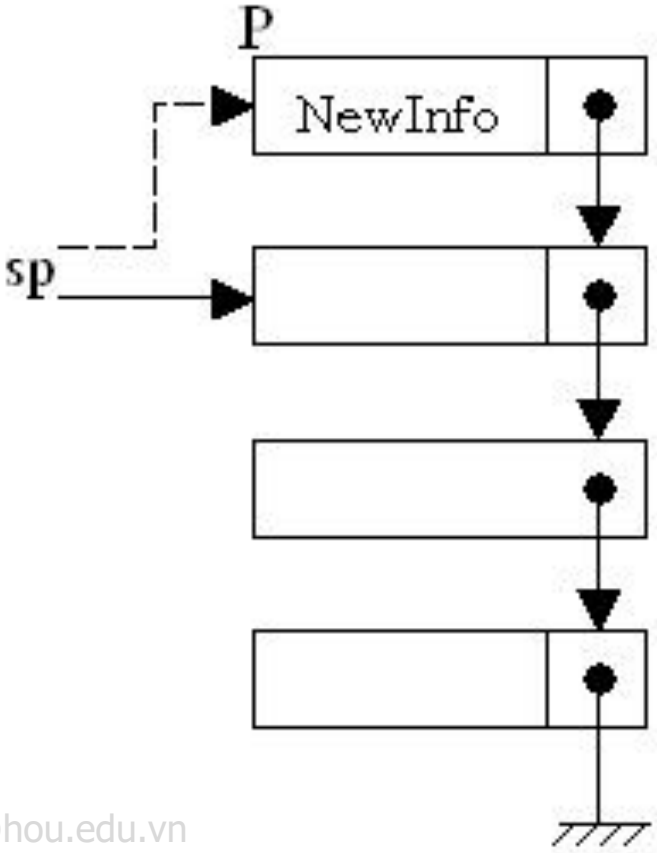
- Khởi tạo một stack rỗng:

```
void Initialize()
{
        sp = NULL;
}
```

Các thao tác trên Stack

- Thêm phần tử vào stack: bổ sung phần tử có trường dữ liệu là info vào stack

```
int Push(Node *&sp, Data info)
{
        Node *p;
        p = (Node*)calloc(1,sizeof(Node));
        if (p==NULL) return 0;
        p -> info = info;
        p -> next = sp;
        sp = p;
        return 1;
}
```

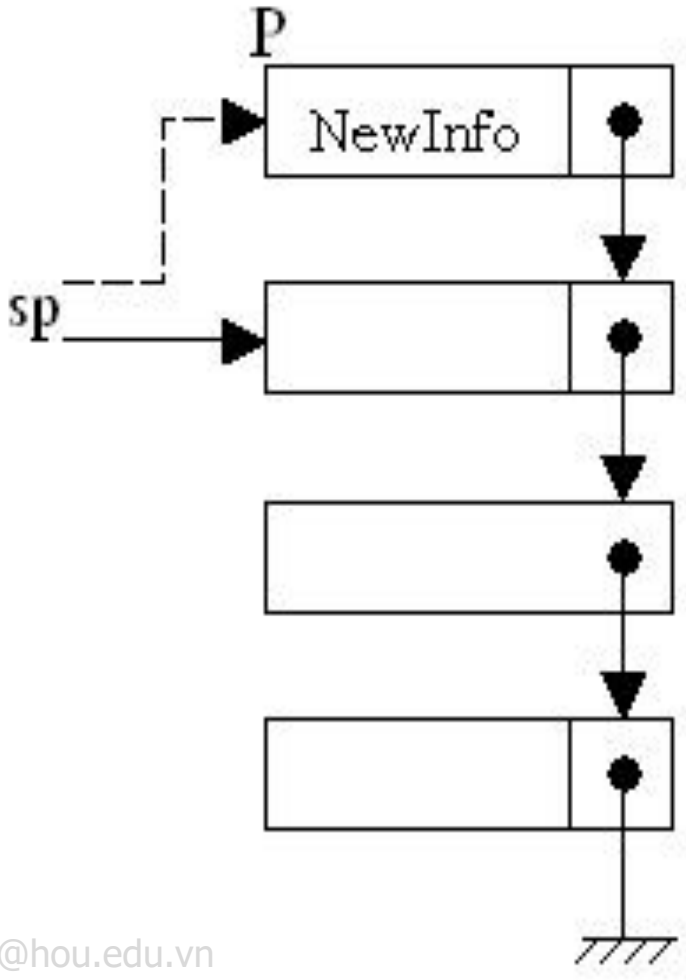


Các thao tác trên Stack

- Lấy một phần tử ra khỏi stack:

Lấy phần tử đầu tiên của stack ra khỏi Stack lưu trường info của phần tử đó.

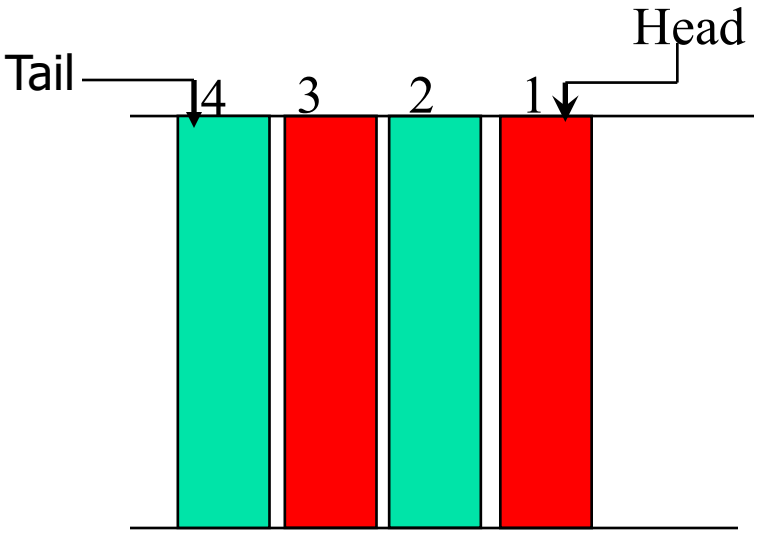
```
int Pop(Node *&sp, Data &x)
{
        Node *p;
        if (sp==NULL) return 0;
        else
        {x = sp -> info;
         p = sp;
         sp = sp -> next;
         free(p);
         return 1;
        }
}
```



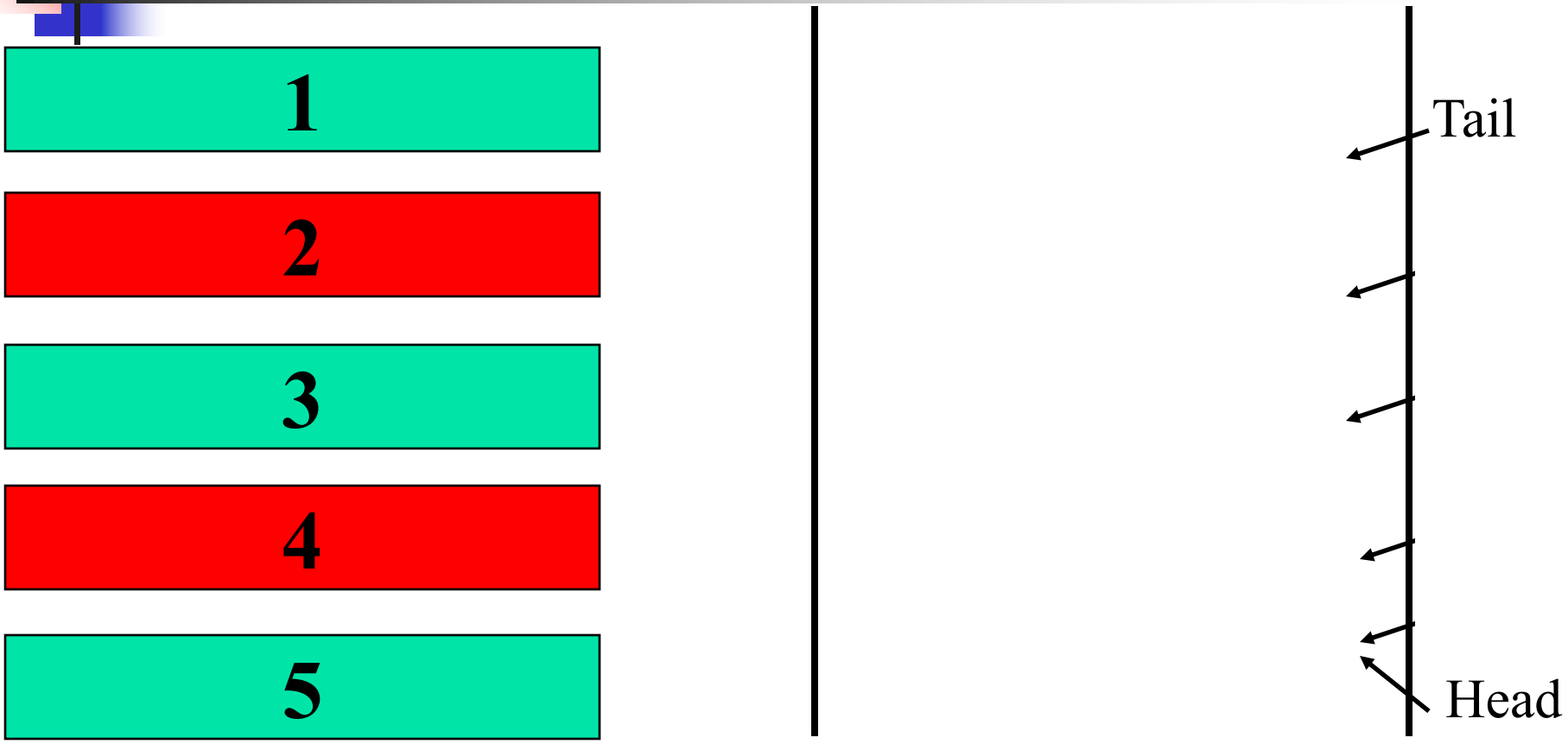
Hàng đợi (Queue)

- Hàng đợi là một kiểu danh sách trong đó được trang bị hai phép toán bổ sung một phần tử vào cuối danh sách và loại bỏ một phần tử ở đầu danh sách.
- Trong hàng đợi một phần tử vào trước sẽ bị đẩy ra trước và phần tử vào sau sẽ bị đẩy ra sau ⇔ gọi là danh sách FIFO(*First In First Out)*

Ví dụ về hàng đợi

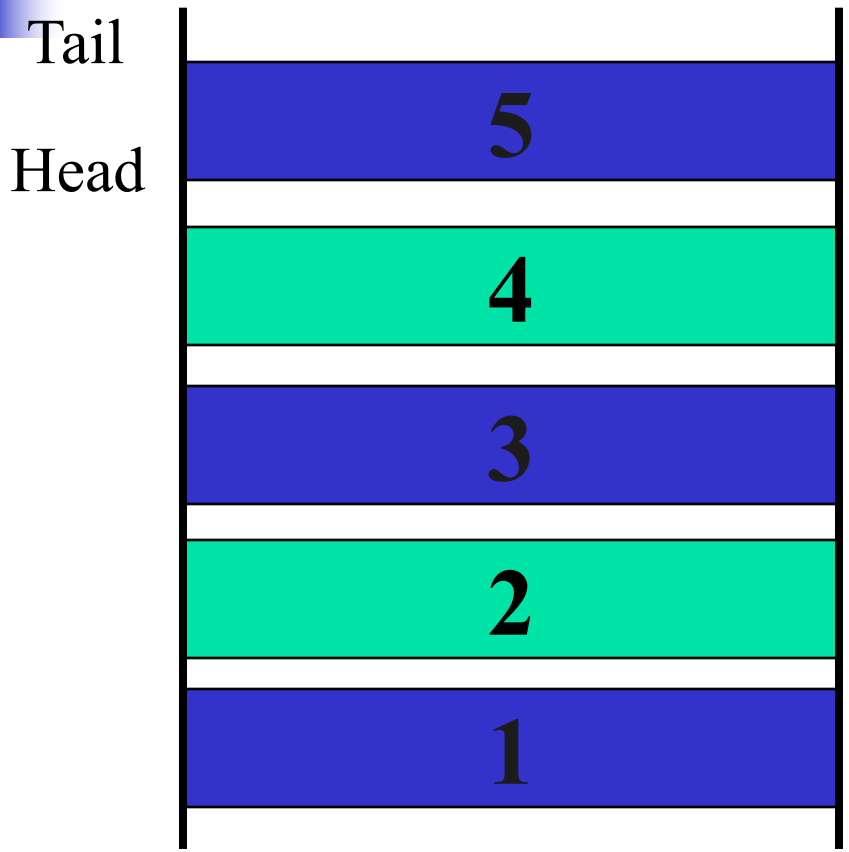


Cơ chế làm việc của Queue

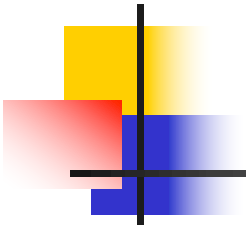


Bổ sung phần tử vào Queue

Cơ chế làm việc của Queue



Lấy phần tử ra khỏi Queue



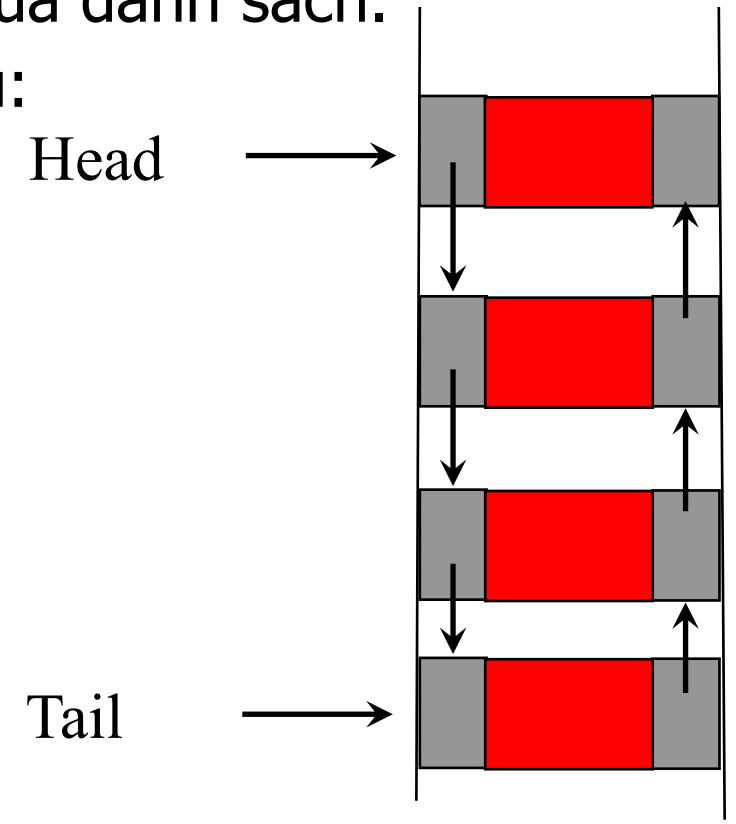
Biểu diễn Queue

- Hàng đợi Queue có thể được biểu diễn bằng mảng hoặc danh sách liên kết.
- Tại mỗi thời điểm ta cần biết vị trí đầu và cuối của hàng đợi.

Biểu diễn Queue bằng danh sách liên kết

- Hàng đợi Queue là một danh sách liên kết đôi. Tại mỗi thời điểm ta cần biết nút đầu và cuối của danh sách.
- Hàng đợi được định nghĩa như sau:

```
struct Node
{
    Data info;
    struct Node *next;
    struct Node *prev;
};
struct Queue
{
Node *Head, *Tail;
}
Queue Q;
```



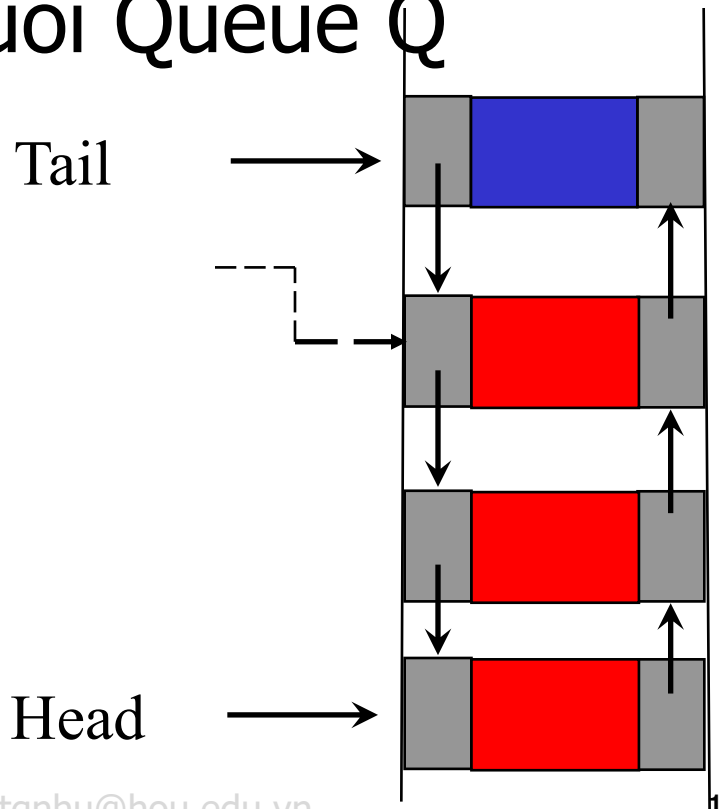
Các thao tác trên Queue

- Khởi tạo Queue rỗng:

```
void Initialize()
{
        Q.Head = NULL;
        Q.Tail = NULL;
}
```

Các thao tác trên Queue

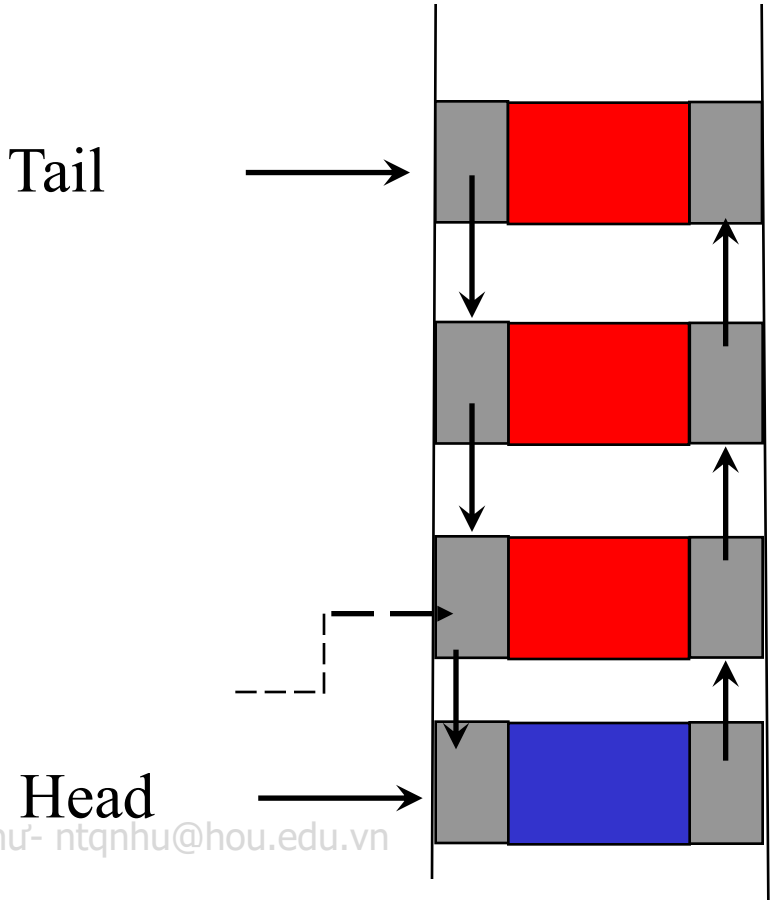
- Bổ sung phần tử vào Queue: Đưa phần tử có giá trị info vào phần tử cuối Queue Q



```
void insert(Queue &Q, Data x)
{    Node *p;
     p = (Node*) calloc ( 1,sizeof ( Node ));
     if ( p = = NULL )   {printf("Ko bo sung duoc"); exit(1);   }
     p -> info = x;
     if ( Q.Tail = = NULL)
     {
          p ->  next = NULL;
          p ->  pre =  NULL;
          Q.Head = p; Q.Tail = p;
     }
     else              //Chèn thêm phần tử mới vào cuối danh sách
     {
          p -> pre = Q.Tail;
          p -> next = NULL;
          Q.Tail->next=p;
          Q.Tail = p;
     }
}
```

Các thao tác trên Queue

- Hủy bỏ phần tử ra khỏi Queue: hủy bỏ phần tử đầu của Queue Q



```
void remove(Queue &Q, Data &x)
{
    Node *p;
    if ( Q.Head = = NULL )
    {   printf("\n Ko huy bo duoc");
        exit(1);
    }
    x = Q.Head -> info;
    p = Q.Head;
    Q.Head = p -> next;
    Q.Head -> pre = NULL;
    free(p);
}
```

Bài tập ngăn xếp

- Viết chương trình thực hiện các chức năng sau:
  - Nhập danh sách n sinh viên, mỗi sinh viên có các thông tin sau: Mã sinh viên, họ và tên, Lớp, Điểm lý thuyết, Điểm Thực hành, Điểm thi, Điểm Trung Bình. Danh sách sinh viên được tổ chức và lưu trữ dưới dạng ngăn xếp (sử dụng thuật toán chèn phần tử vào cuối danh sách khi tạo danh sách)
  - Nhập vào từ bàn phím thông tin của 1 sinh viên, chèn sinh viên này vào đầu danh sách
  - Nhập vào từ bàn phím thông tin của 1 sinh viên khác, chèn sinh viên này vào sau sinh viên có mã sinh viên là x, với x là mã sinh viên được nhập vào từ bàn phím

Bài tập hàng đợi

- Thông tin về nhân viên gồm: mã số nhân viên (MaNV - kiểu nguyên), họ tên (HoTen - kiểu xâu kí tự), ngày sinh (NS - kiểu xâu kí tự), tổng lương (TL - kiểu thực).
- Thực hiện các yêu cầu sau (bằng ngôn ngữ lập trình C):
  - Khai báo cấu trúc dữ liệu (dạng con trỏ) của hàng đợi để quản lý danh sách nhân viên (Kiểu danh sách là ListNV lưu các nhân viên có kiểu là NV).
  - Viết hàm thêm một nhân viên vào đầu danh sách liên kết đôi trên.
  - Viết hàm tính tổng lương của tất cả các nhân viên trong danh sách.
  - Viết hàm hủy nhân viên cuối cùng ra khỏi danh sách
  - Viết hàm tìm nhân viên có mức lương cao nhất
  - Viết hàm in ra danh sách Nhân viên có mức lương trên 2000000
  - Sắp xếp danh sách theo thứ tự tăng dần của MaNV