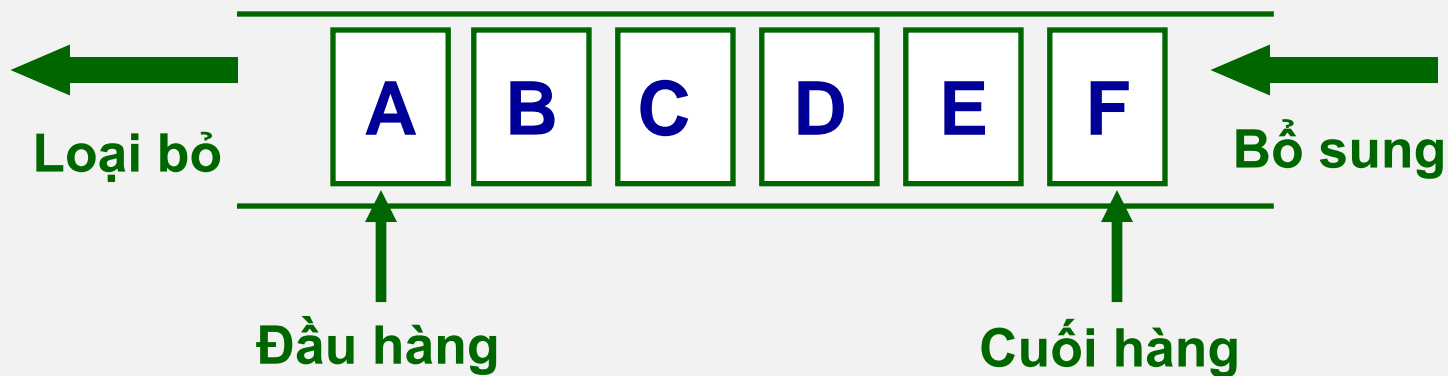


BÀI 4

DANH SÁCH TUYỂN TÍNH

4.6. HÀNG ĐỢI

4.6.1. Khái niệm hàng đợi



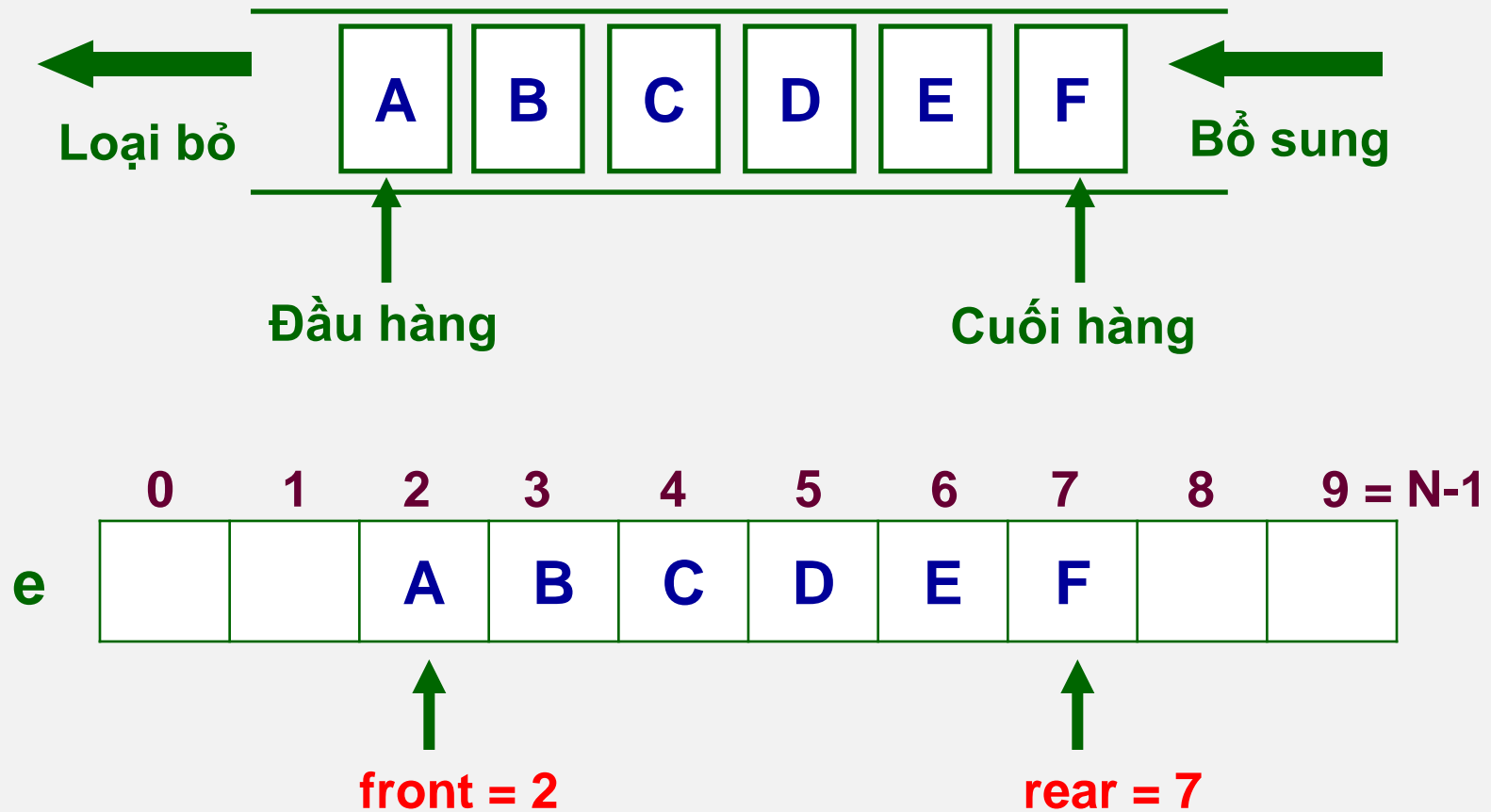
- **Hàng đợi là một danh sách tuyến tính, trong đó:**
 - Việc bổ sung một phần tử vào hàng đợi được thực hiện ở một đầu gọi là cuối hàng.
 - Việc loại bỏ một phần tử ra khỏi hàng đợi được thực hiện ở đầu kia gọi là đầu hàng.
- **Danh sách kiểu hàng đợi còn gọi là danh sách FIFO – First In First Out.**

4.6.2. Biểu diễn cấu trúc dữ liệu

- Giả sử các phần tử của hàng đợi có kiểu dữ liệu là `Item`, chiều dài cực đại của hàng đợi là **N**.
- Hàng đợi được lưu trong bộ nhớ máy tính bởi mảng 1 chiều (lưu trữ kế tiếp).
- Mỗi phần tử của mảng lưu một phần tử của hàng đợi.
- Hàng đợi được biểu diễn là một cấu trúc có 3 thành phần:
 - Mảng **e** lưu các phần tử của hàng đợi.
 - Biến **front** lưu chỉ số của phần tử đầu hàng.
 - Biến **rear** lưu chỉ số của phần tử cuối hàng.

Biểu diễn cấu trúc dữ liệu (tt)

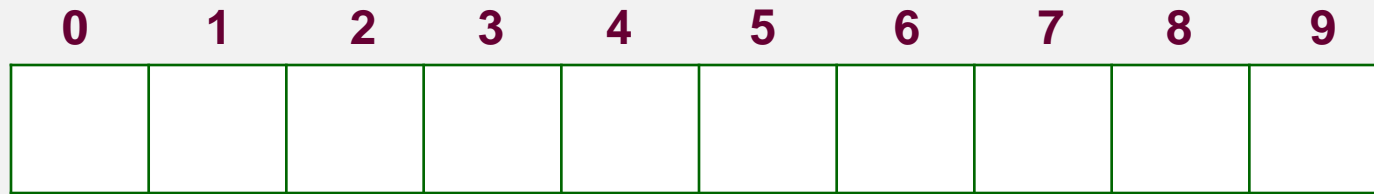
- Lưu trữ hàng đợi bằng danh sách kế tiếp (lưu trữ bằng mảng một chiều)



Biểu diễn cấu trúc dữ liệu (tt)

```
#define MAX N
//Định nghĩa kiểu Item
    struct Queue{
        int front;
        int rear;
        Item e[MAX];
    };
Queue Q;
/* Q.rear = -1 -> hàng đợi rỗng,
Q.rear = MAX-1 -> Hàng đợi đầy */
```

4.6.3. Các phép toán trên hàng đợi



front = 0

rear = -1

1. Khởi tạo hàng đợi rỗng

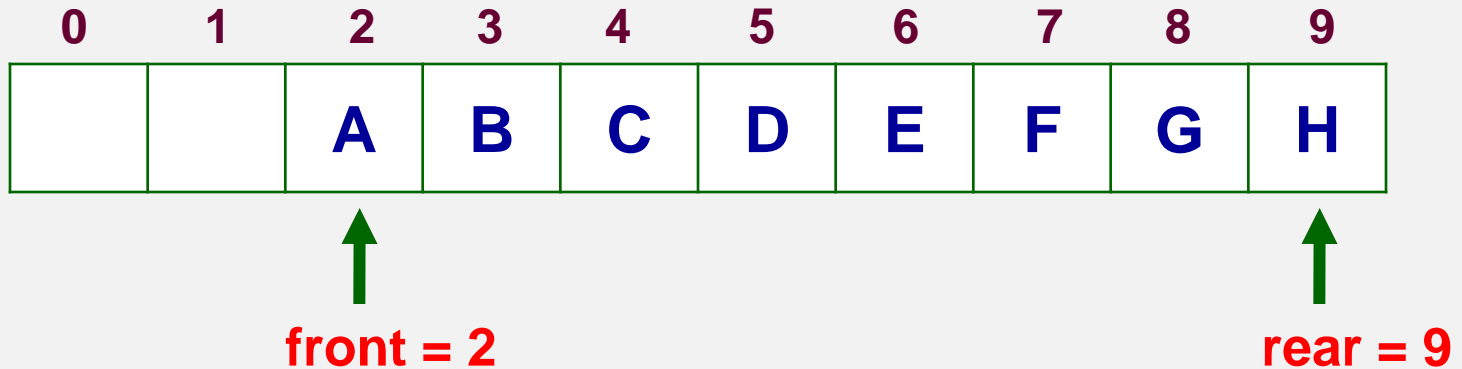
```
void initQueue (Queue &Q) {  
    Q.front = 0;  
    Q.rear = -1;  
}
```

2. Kiểm tra hàng đợi rỗng

```
int empty (Queue Q)  
{  
    return (Q.rear == -1);  
}
```

Các phép toán trên hàng đợi (tt)

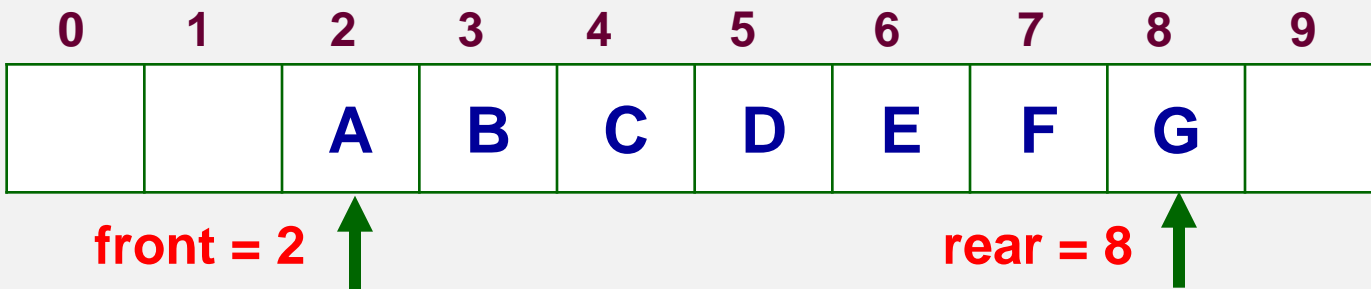
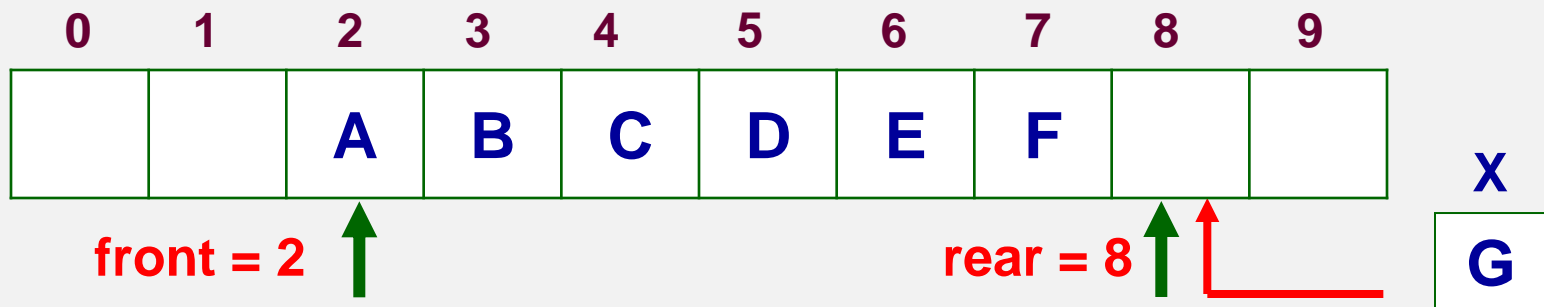
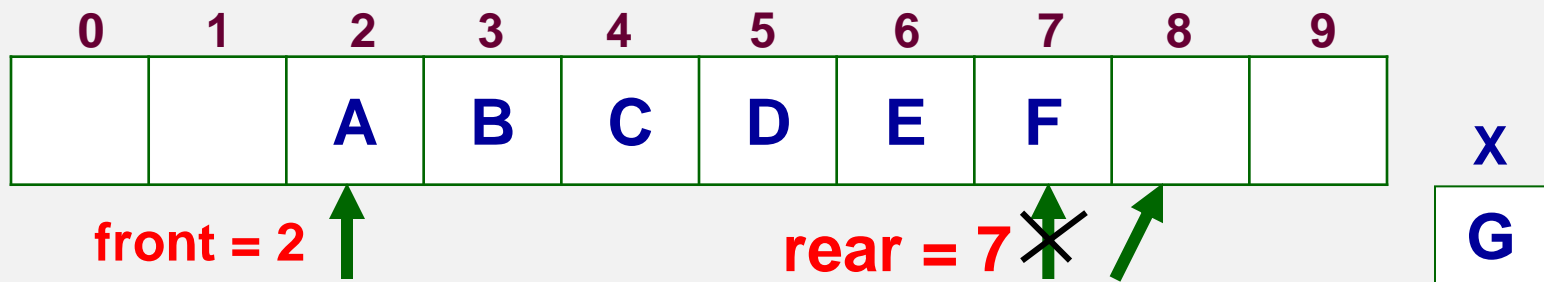
3. Kiểm tra hàng đợi đầy



```
int full (Queue Q)
{
    return (Q.rear == MAX-1) ;
}
```

Các phép toán trên hàng đợi (tt)

4. Bổ sung một phần tử vào cuối hàng đợi



Các phép toán trên hàng đợi (tt)

4. Bổ sung một phần tử vào cuối hàng đợi

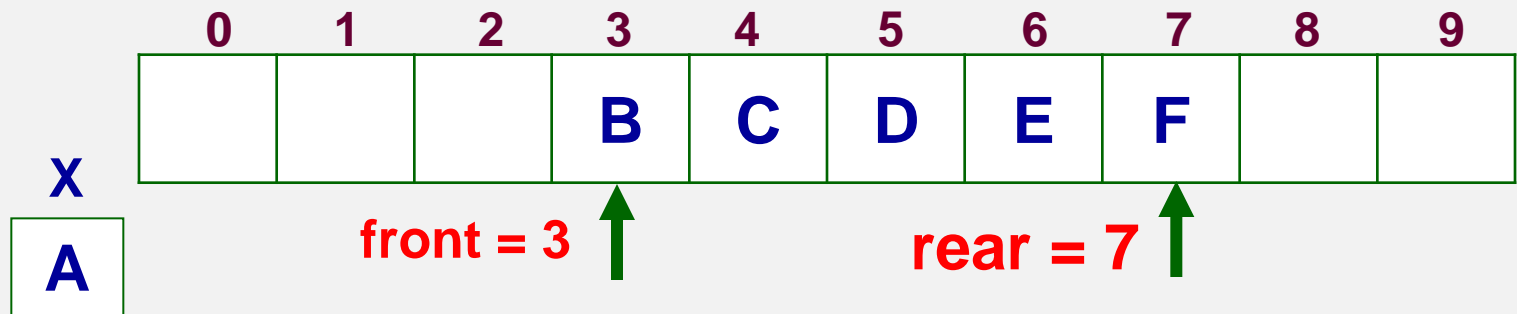
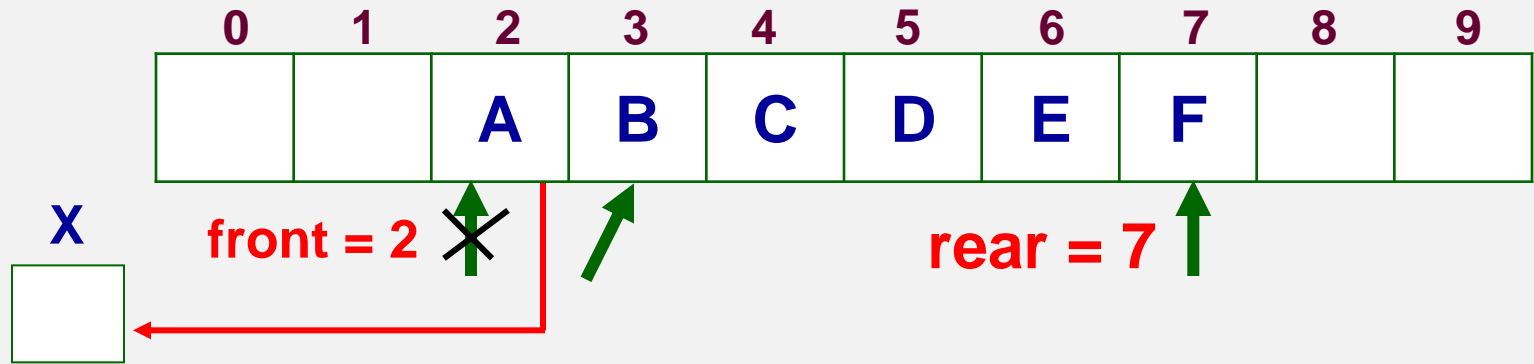
```
int add (Queue &Q, Item X)
{
    if (full(Q))
        return 0;
    else
    {
        Q.rear++;
        Q.e[Q.rear] = X;
        return 1;
    }
}
```

Hàm add thực hiện bổ sung một phần tử vào cuối hàng, hàm trả về 1 nếu bổ sung thành công, ngược lại hàm trả về 0

Các phép toán trên hàng đợi (tt)

5. Lấy ra một phần tử ở đầu hàng đợi

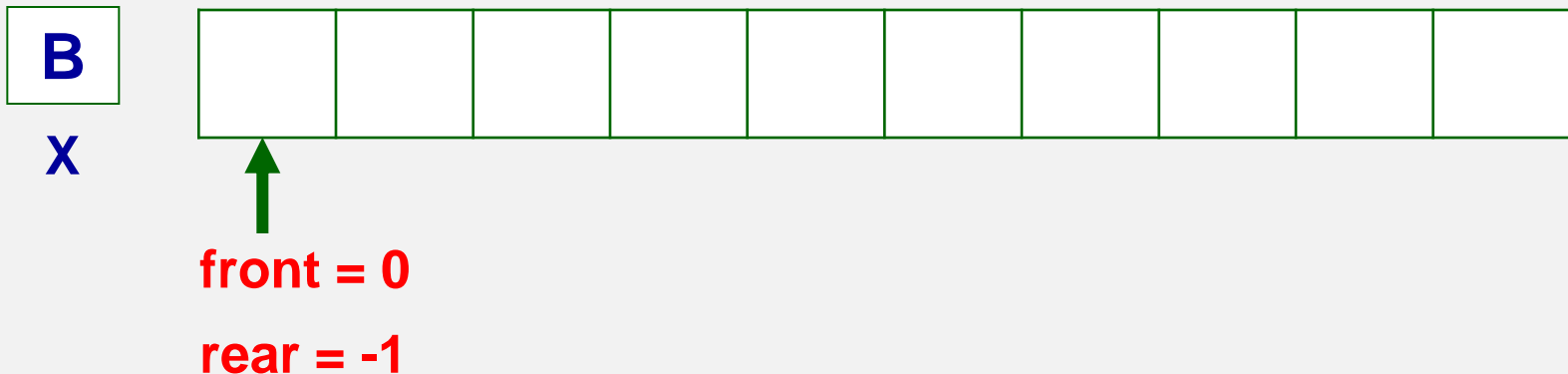
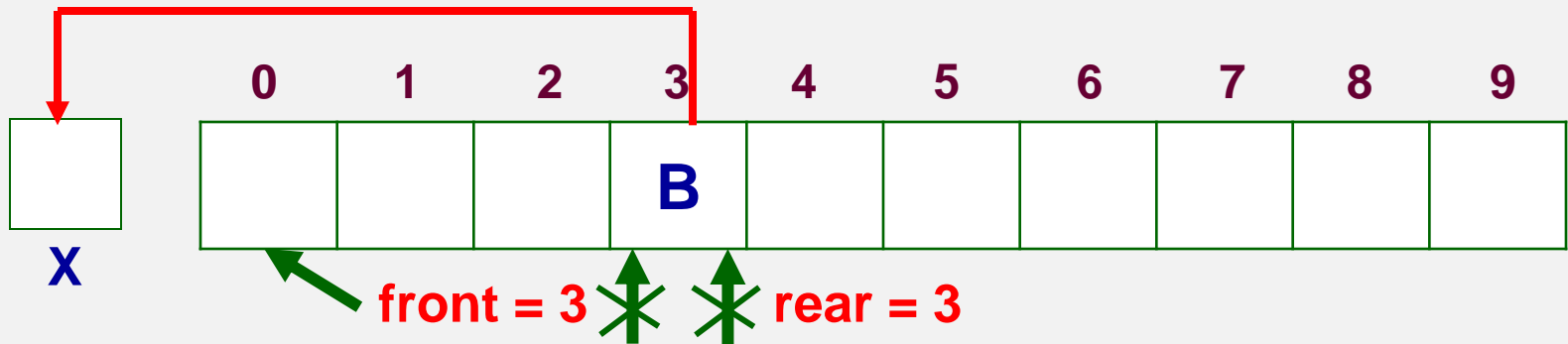
- Trường hợp hàng đợi có nhiều hơn 1 phần tử.



Các phép toán trên hàng đợi (tt)

5. Lấy ra một phần tử ở đầu hàng đợi

- Trường hợp hàng đợi có 1 phần tử



Các phép toán trên hàng đợi (tt)

5. Lấy ra một phần tử ở đầu hàng đợi

```
int removeQ(Queue &Q, Item &X) {  
    if (empty(Q)) return 0;  
    else{  
        X = Q.e[Q.front];  
        if (Q.front == Q.rear)  
            initQueue(Q);  
        else  
            Q.front = Q.front + 1;  
        return 1;  
    }  
}
```

Hàm removeQ thực hiện lấy ra một phần tử ở đầu hàng đợi, hàm trả về 1 nếu lấy ra thành công, ngược lại hàm trả về 0.

4.6.4. Ứng dụng hàng đợi

- **Viết chương trình:**
 - Nhập một chuỗi ký tự, sử dụng hàng đợi, việc nhập kết thúc khi gõ enter.
 - In chuỗi ra màn hình.
 - Đếm các nguyên âm trong chuỗi.