

## Chương 3 DANH SÁCH TUYẾN TÍNH

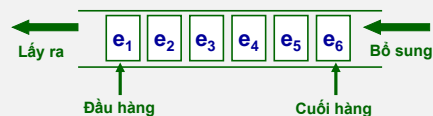
### 3.8. HÀNG ĐỢI

Design by Minh An

Email: anvanminh.hau@gmail.com

1

#### 3.8.1. Khái niệm hàng đợi



- Hàng đợi là một danh sách tuyến tính, trong đó:
  - Việc bỏ sung một phần tử vào hàng đợi được thực hiện ở một đầu gọi là cuối hàng.
  - Việc lấy ra 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.

Design by Minh An

2

#### 3.8.2. Biểu diễn cấu trúc dữ liệu

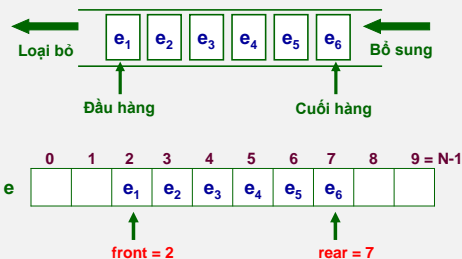
- Giả sử các phần tử của hàng đợi có kiểu dữ liệu là **DataType**, chiều dài cực đại của hàng đợi là **N**.
- Hàng đợi được lưu trong bộ nhớ máy tính bởi mảng 1 chiều (lưu trữ kế tiếp).
- Mỗi phần tử của mảng lưu một phần tử của hàng đợi.
- Hàng đợi được biểu diễn là một cấu trúc có 3 thành phần:
  - 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.

Design by Minh An

3

#### Biểu diễn cấu trúc dữ liệu (tt)

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



Design by Minh An

4

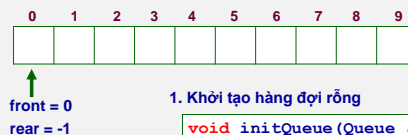
#### Biểu diễn cấu trúc dữ liệu (tt)

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

Design by Minh An

5

#### 3.8.3. Các phép toán trên hàng đợi



##### 1. Khởi tạo hàng đợi rỗng

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

##### 2. Kiểm tra hàng đợi rỗng

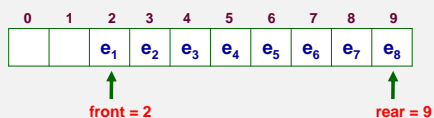
```
int empty(Queue Q)
{
    return (Q.rear == -1);
}
```

Design by Minh An

6

### Các phép toán trên hàng đợi (tt)

#### 3. Kiểm tra hàng đợi đầy



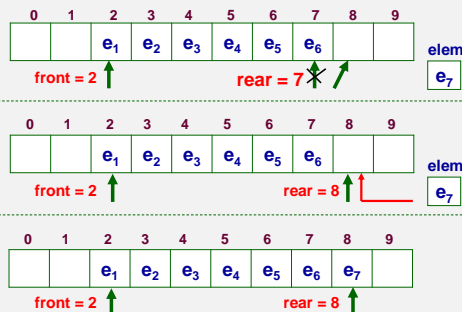
```
int full (Queue Q)
{
    return (Q.rear == MAX-1);
}
```

Design by Minh An

7

### Các phép toán trên hàng đợi (tt)

#### 4. Bỏ sung một phần tử vào cuối hàng đợi



Design by Minh An

8

### Các phép toán trên hàng đợi (tt)

#### 4. Bỏ sung một phần tử vào cuối hàng đợi

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

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

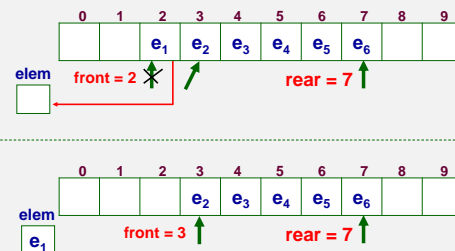
Design by Minh An

9

### 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.



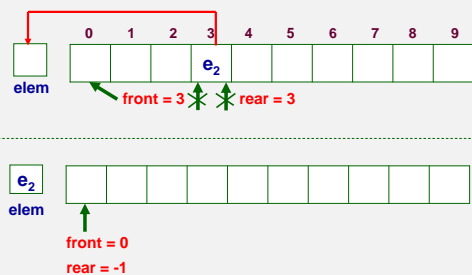
Design by Minh An

10

### 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



Design by Minh An

11

### Các phép toán trên hàng đợi (tt)

#### 5. Lấy ra một phần tử ở đầu hàng đợi

```
int removeQ (Queue &Q, DataType &elem) {
    if (empty(Q)) return 0;
    else {
        elem = 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.

Design by Minh An

12

### 3.8.4. 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.

Design by Minh An