Cấu trúc dữ liệu và thuật toán

PGS. TS. Phạm Tuấn Minh
Trường Công nghệ Thông tin, Đại học Phenikaa
minh.phamtuan@phenikaa-uni.edu.vn
https://sites.google.com/site/phamtuanminh/

Chương 2: Mảng và danh sách liên kết

- Cấu trúc lưu trữ mảng
- Danh sách liên kết
- **Hàng đợi**
- Ngăn xếp

Đọc thêm: Cài đặt hàng đợi bằng mảng dùng C

- Khai báo, khởi tạo
- Các thao tác hàng đợi
  - isEmptyQueue()
  - isFullQueue()
  - enqueue()
  - dequeue()
  - peek()

Khai báo, khởi tạo

```
#define MAX_SIZE 10
typedef struct {
    int data[MAX_SIZE];
    int front; //  vị trí đầu hàng đợi.
    int rear; //  vị trí cuối hàng đợi.
    int size; // số lượng phần tử trong hàng đợi
} Queue;

void initQueue(Queue *q) {
    q->front = 0;
    q->rear = -1;
    q->size = 0;
}
```

1-4

Kiểm tra hàng đợi rỗng, đầy

```
bool isEmptyQueue(Queue *q) {
    return q->size == 0;
}

bool isFullQueue(Queue *q) {
    return q->size == MAX_SIZE;
}
```

- isEmptyQueue: O(1)
- isFullQueue: O(1)

1-5

Thêm phần tử vào hàng đợi

```
void enqueue(Queue *q, int value) {
    if (isFullQueue(q)) {
        printf("Queue is full. Cannot enqueue %d\n", value);
        return;
    }
    q->rear = (q->rear + 1) % MAX_SIZE;
    q->data[q->rear] = value;
    q->size++;
}
```

- O(1)

1-6

Xóa phần tử khỏi hàng đợi

```
int dequeue(Queue *q) {
    if (isEmptyQueue(q)) {
        printf("Queue is empty. Cannot dequeue.\n");
        return -1;
    }
    int value = q->data[q->front];
    q->front = (q->front + 1) % MAX_SIZE;
    q->size--;
    return value;
}
```

- O(1)

1-7

Lấy phần tử đầu hàng đợi mà không xóa

```
int peek(Queue *q) {
    if (isEmptyQueue(q)) {
        printf("Queue is empty. Cannot peek.\n");
        return -1;
    }
    return q->data[q->front];
}
```

- O(1)

1-8

Cấu trúc dữ liệu và giải thuật

- Nội dung bài giảng được biên soạn bởi PGS. TS. Phạm Tuấn Minh.

1-9