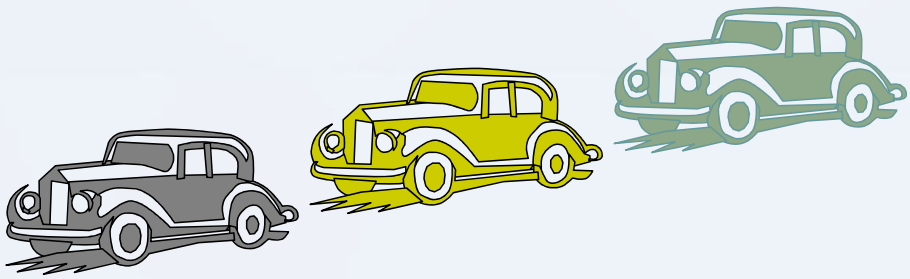
Bài 9. Cấu trúc dữ liệu hàng đợi



Danh sách kiểu Hàng đợi (Queue)

❖ Queue là cách tổ chức lưu trữ các đối tượng dưới dạng một danh sách tuyến tính mà việc bổ sung đối tượng được thực hiện ở đầu danh sách và việc lấy đối tượng ra được thực hiện ở cuối của danh sách.

❖ Queue còn được gọi là danh sách kiểu FIFO (First In First Out - vào trước ra trước)

Cấu trúc dữ liệu trừu tượng Queue (The Queue ADT)

- Queue ADT lưu trữ các đối tượng bất kỳ
- Thêm vào và xóa đi (lấy ra) theo kiểu FIFO
- Thêm vào thực hiện ở cuối của queue và lấy ra thực hiện ở đầu queue
- Các phép toán chính thực hiện trên queue:
  - enqueue(Object o): bổ sung một phần tử o vào cuối của queue.
  - dequeue(Object &o): Xóa đi phần tử đầu của queue
- Các phép toán bổ trợ
  - front(): trả lại phần tử đầu của queue nhưng không xóa nó đi
  - size(): trả lại số phần tử hiện đang được lưu trữ trong queue
  - isEmpty(): trả lại giá trị kiểu boolen để xác định có phần tử được lưu trữ trong queue không?
- Ngoại lệ: thực hiện dequeue hoặc enqueue trong khi queue rỗng hoặc đầy

➔ ta cần phải chuyển đến ngoại lệ

Một số ứng dụng của queue

- Các ứng dụng trực tiếp
  - Danh sách hàng đợi
  - Truy nhập các nguồn dùng chung (ví dụ máy in trong mạng cục bộ)
  - Đa lập trình

Các ứng dụng không trực tiếp

- Cấu trúc dữ liệu hỗ trợ cho các thuật toán
- Làm thành phần của các cấu trúc dữ liệu khác

Cài đặt queue bằng mảng

- Sử dụng một mảng kiểu vòng có kích thước N
- Sử dụng 2 biến lưu trữ chỉ số của phần tử trước và phần tử sau:
  - f lưu chỉ số phần tử trước
  - r lưu trữ chỉ số phần tử chuẩn bị được đưa vào
- Vị trí r của mảng là rỗng

Cấu hình bình thường

Cấu hình vòng lại

Các phép toán trên queue

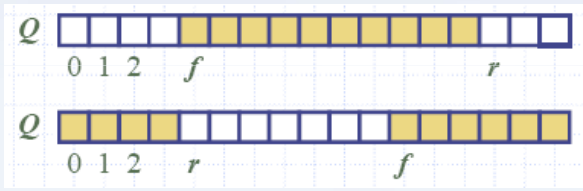
- Chúng ta sử dụng phép toán modulo để xác định số phần tử còn lại của queue

Algorithm *size*()
  return $(N - f + r) \bmod N$

Algorithm *isEmpty*()
  return $(f = r)$

Các phép toán trên queue (tiếp)

- Phép toán ensqueue dẫn đến ngoại lệ khi mảng đầy

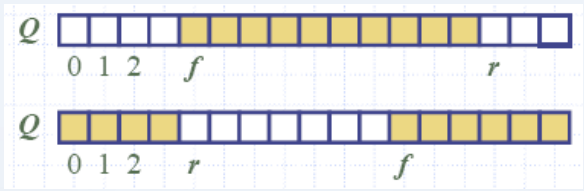


Algorthim enqueue(Object o)

```
if size()=N-1 then
    return 0
else
        Q[r]←o
        r←(r+1) mod N
        return 1;
```

Các phép toán trên queue (tiếp)

- Phép toán dequeue dẫn đến ngoại lệ khi mảng rỗng



Algorthim dequeue(Object &o)

```
if isEmpty() then
    return 0
else
        o←Q[f]
        f←(f+1) mod N
        return 1
```

Phát triển queue dựa trên mảng

- Khi thêm phần tử vào mảng, có thể xảy ra ngoại lệ. Để tránh điều đó ta có thể sử dụng một mảng có kích thước lớn hơn
- Tương tự phát triển stack dựa trên mảng
- Thời gian thực hiện của thuật toán là:
  - O(n) với chiến lược gia tăng
  - O(1) với chiến lược gấp đôi

Cài đặt queue bằng C++

- ADT queue không phù hợp khi cài đặt bằng C++ (trên DOS) vì nó yêu cầu định nghĩa lớp có cho phép dẫn đến ngoại lệ. Tuy nhiên chúng ta vẫn có thể sử dụng C++ để cài đặt queue

```
#define N   //const integer
template <class Object>
class Queue{
    private:
      Object Q[N];
       int f, r;
public:
    Queue();
    int isEmpty();
    int size();
    Object front();
    int enqueue(Object o);
    int dequeue(Object &o);
};
```

Cài đặt queue bằng C++

```
int Queue<Object>::enqueue(Object o){
    if (size()=N-1)
     return 0;
   else{
        Q[r]=o;
        r =(r+1)%N;
        return 1;
   }
}
```

Bài tập

1. Cài đặt lớp Queue mẫu bằng cách sử dụng mảng
2. Cài đặt Queue mẫu bằng cách sử dụng danh sách liên kết
3. Cài đặt lớp ứng dụng sử dụng lớp Queue để tổ chức lưu trữ các đối tượng là các số nguyên. Lớp có các chức năng:
    1. Thêm vào Queue 1 phần tử
    2. Lấy phần tử ra khỏi queue và hiển thị lên màn hình
    3. Cho biết số phần tử hiện có của Queue
    4. Cho biết Queue rỗng hay đầy