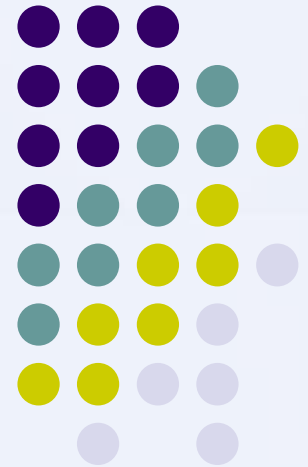
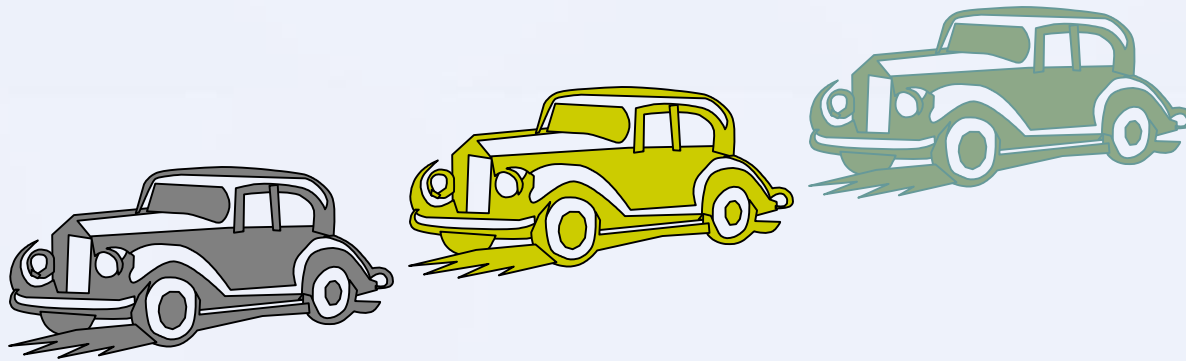


Bài 9. Cấu trúc dữ liệu hàng đợi

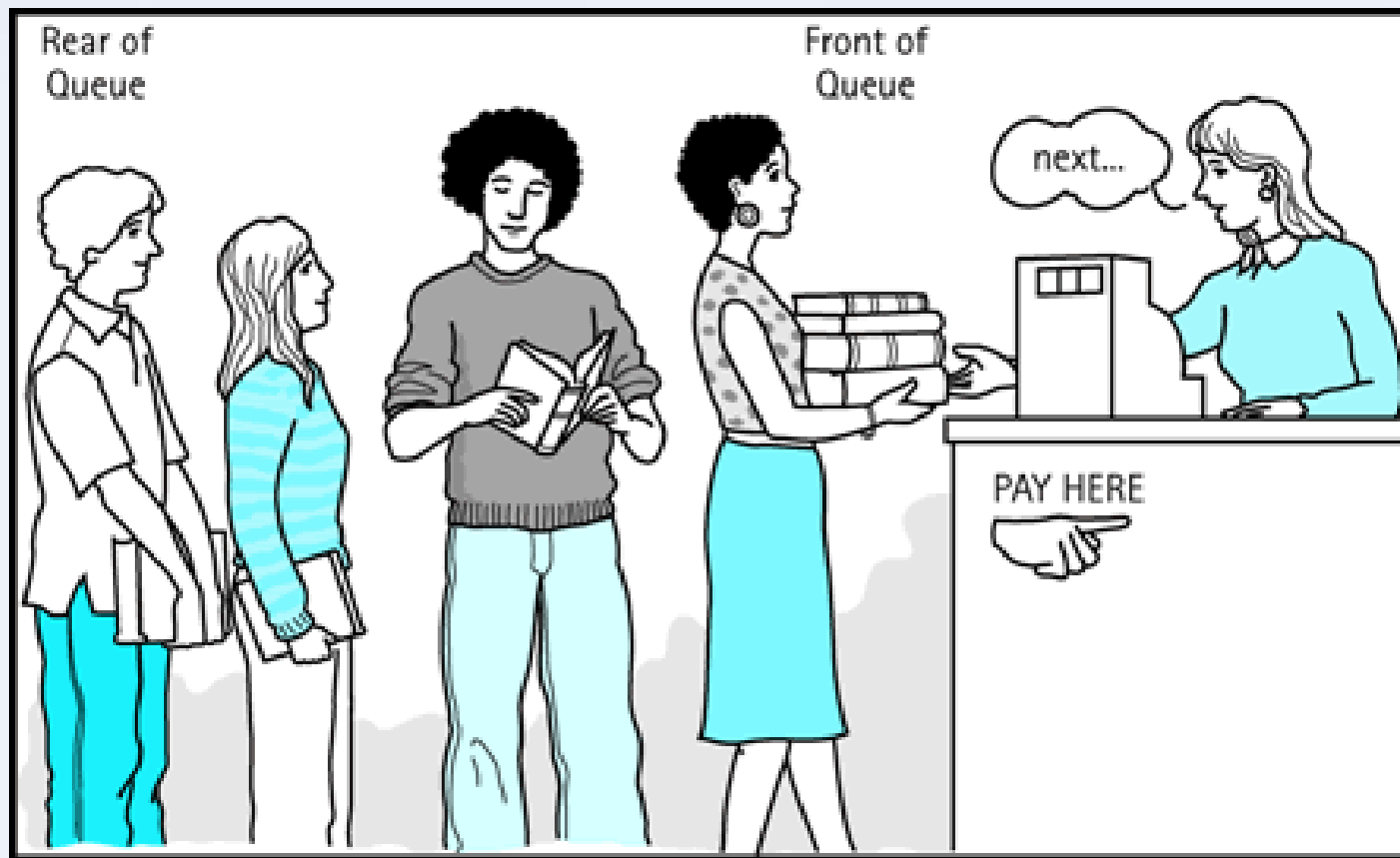


Danh sách kiểu Hàng đợi (Queue)



- ❖ **Queue** là một cấu trúc dữ liệu trừu tượng, 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 này danh sách và việc lấy đối tượng ra được thực hiện đầu kia của danh sách (FIFO – first in first out).
- ❖ **Queue** còn được gọi là danh sách kiểu **FIFO (First In First Out** - vào trước ra trước)

Danh sách kiểu Hàng đợi (Queue)



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:
 - **push(Object o)**: bổ sung một phần tử o vào cuối của queue.
 - **pop()**: Xóa đi phần tử đầu của queue
- ❖ Các phép toán hỗ trợ
 - **front()**: trả lại phần tử đầu queue nhưng không xóa
 - **back()**: trả lại phần tử cuối queue nhưng không xóa
 - **size()**: trả lại số phần tử hiện đang được lưu trữ trong queue
 - **empty()**: trả lại giá trị kiểu boolean để xác định có phần tử được lưu trữ trong queue không?
- ❖ Ngoại lệ: thực hiện **push** hoặc **pop** trong khi queue rỗng, khi đó ta cần phải chuyển nó đến ngoại lệ

Một số ứng dụng của queue



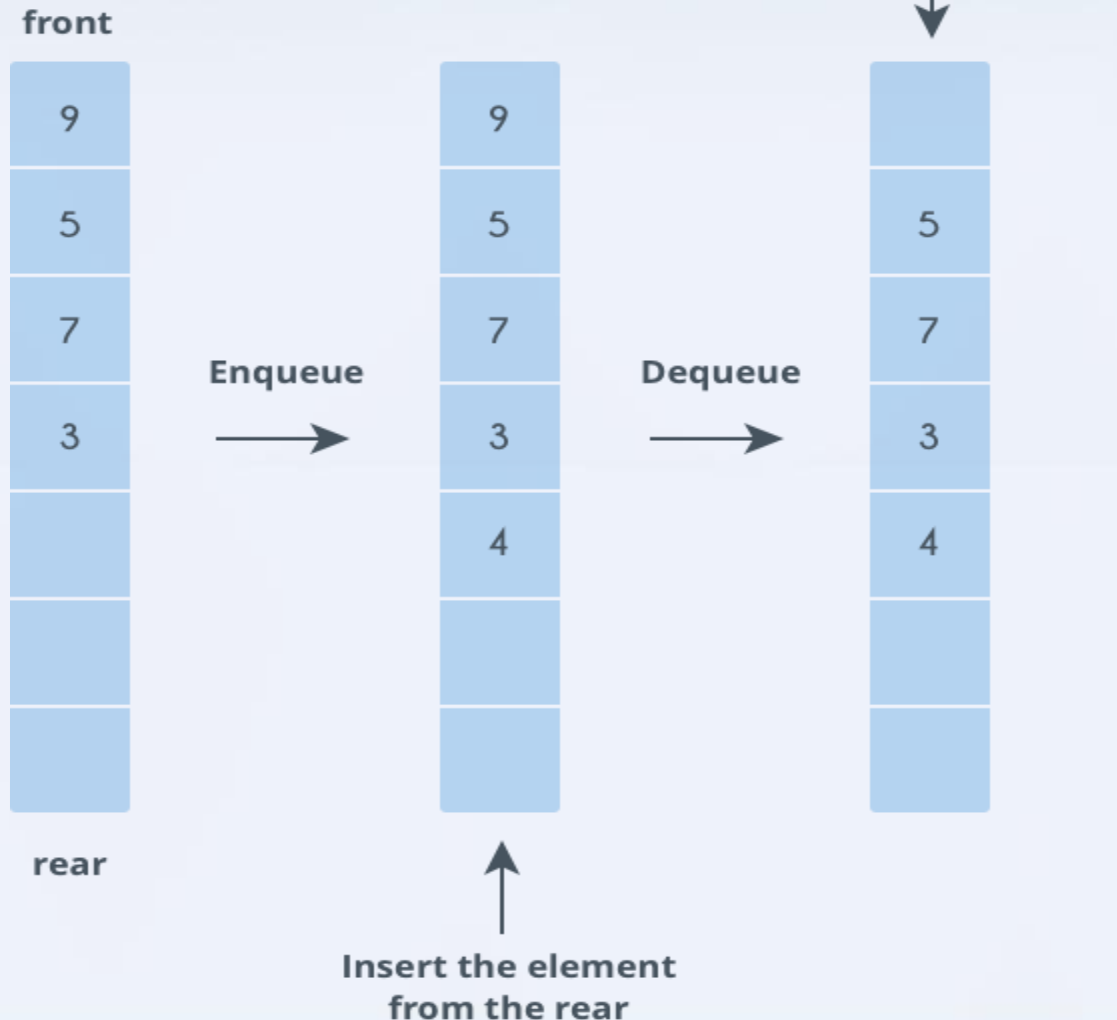
- Các ứng dụng trực tiếp
 - Danh sách hàng đợi
 - Truy nhập các nguồn dùng chung (ví dụ máy in trong mạng cục bộ)
 - Đa lập trình

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

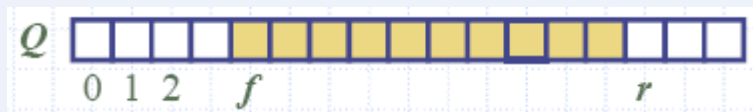
Sử dụng mảng thông thường



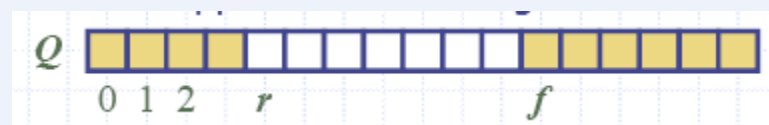
Cài đặt queue bằng mảng

- 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

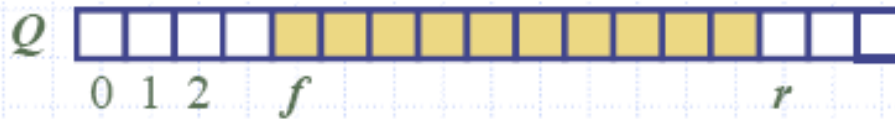
- Chúng ta sử dụng phép toán modulo để xác định số phần tử còn lại của queue

Algorithm *size()*

return $(N - f + r) \bmod N$

Algorithm *isEmpty()*

return $(f = r)$



Các phép toán trên queue(tiếp)



- Phép toán enqueue dẫn đến ngoại lệ khi mảng đầy

Algorithm `push(Object o)`

if `size()=N-1` then

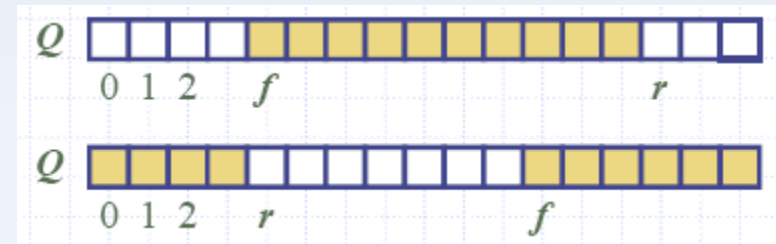
return 0

else

$Q[r] \leftarrow o$

$r \leftarrow (r+1) \bmod N$

return 1;



Các phép toán trên queue(tiếp)



- Phép toán **dequeue** dẫn đến ngoại lệ khi mảng rỗng

Algorithm **pop()**

if isEmpty() then

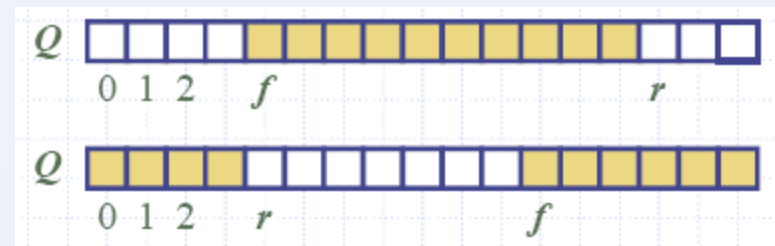
return 0

else

$o \leftarrow Q[f]$

$f \leftarrow (f+1) \bmod N$

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 như phát triển stack dựa trên mảng
- Thời gian thực hiện của thuật toán là:
 - $O(n)$ với chiến lược ra 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();
        bool empty();
        int size();
        Object front();
        void push(Object o);
        void pop();
};
```

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

Cài đặt Queue trong C++



- Cài đặt queue bằng mảng
<https://ideone.com/RwGifP>
- Cài đặt queue bằng node liên kết đơn
<https://ideone.com/4flzlm>

Bài tập 1. Cánh cửa thần kỳ



Đề bài

<http://laptrinhonline.club/problem/themisutc>

Code tham khảo

<https://ideone.com/5EAzNo>

Bài tập 2. Tập tam giác



Đề bài

<http://laptrinhonline.club/problem/tichpxsotamgiac>

Code tham khảo

<https://ideone.com/1eIIHZ>

Bài tập 3. Búp bê Nga



Đề bài

<http://laptrinhonline.club/problem/tichpxbupbenga>

Code tham khảo

<https://ideone.com/9X8mLC>

Bài tập 4. Thuật toán BFS bài toán thang máy



Đề bài

<http://laptrinhonline.club/problem/tichpxthangmay>

Code tham khảo

<https://ideone.com/ukfQwy>

Cấu trúc dữ liệu hàng đợi hai đầu



- Khái niệm
 - Dữ liệu có thể thêm hoặc xóa ở cả đầu (front) và cuối (rear) của hàng đợi → sự kết hợp của cả stack và queue
- Cài đặt
 - Cài đặt bằng mảng
 - Cài đặt bằng danh sách móc nối kép

Cấu trúc dữ liệu hàng đợi ưu tiên



- Khái niệm

- Giống hàng đợi thông thường
- Thứ tự các phần tử phụ thuộc vào độ ưu tiên của phần tử đó
- Ưu tiên nhất \rightarrow đầu dãy để được lấy ra trước nhất
- Ví dụ: $\{4, 1, 8, 7, 3\}$ với số càng lớn độ ưu tiên càng cao

