

HK I, 2012-2013

Bài 7: Hàng đợi

Giảng viên: Hoàng Thị Điệp

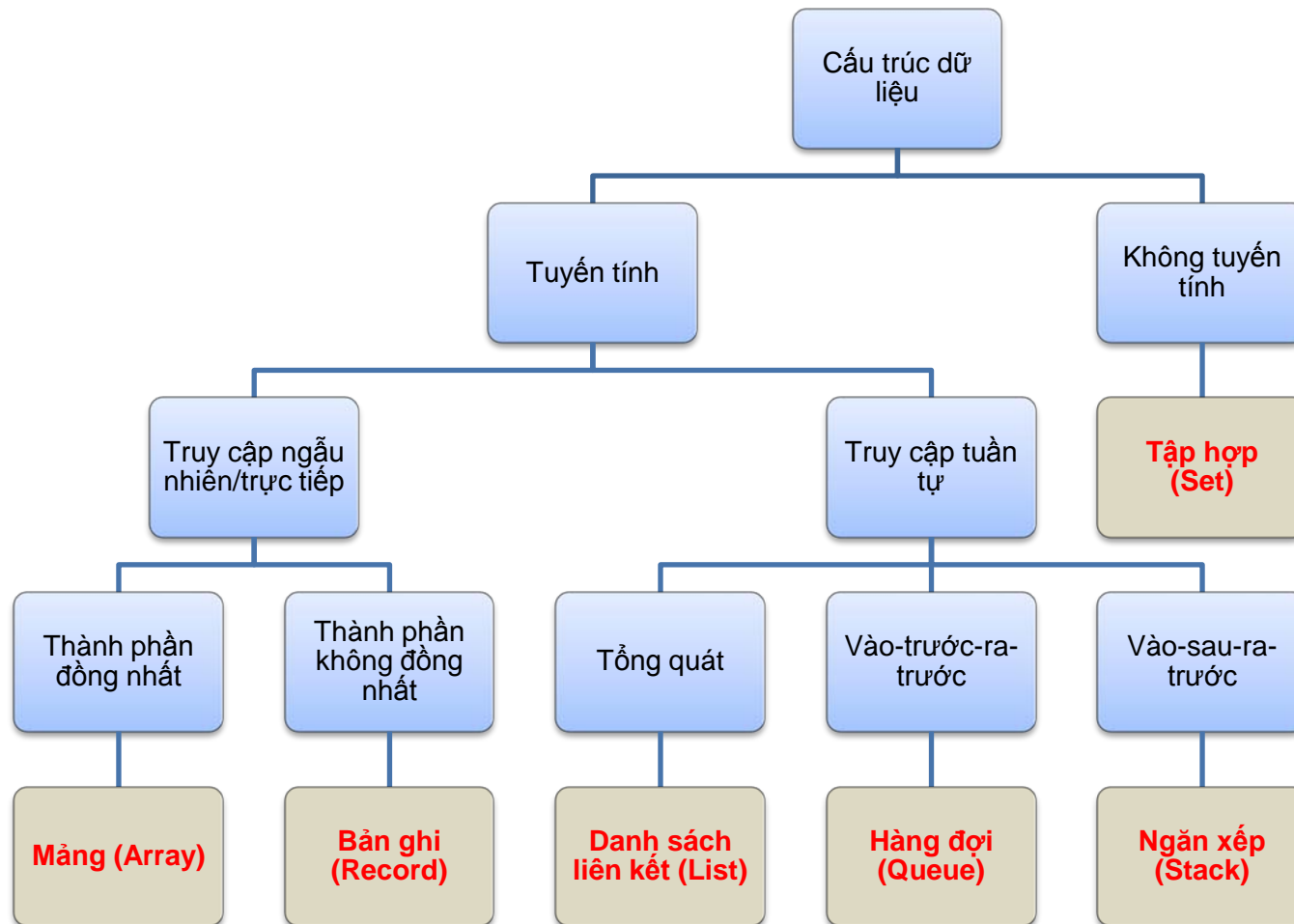
Khoa Công nghệ Thông tin – Đại học Công Nghệ

Nguồn tham khảo chính:

<http://www.cs.nyu.edu/~melamed/courses/102/lectures/>

<http://users.encs.concordia.ca/~dssouli/COEN352.html>

Tổng quan



Hàng đợi

- Hàng đợi là gì?
 - Là một danh sách nhưng các phép toán chỉ được thực hiện ở hai đỉnh của danh sách. Một đỉnh gọi là đầu hàng, đỉnh còn lại gọi là cuối hàng.
- Tính chất
 - Vào trước ra trước (First In First Out: FIFO)



KDLTT hàng đợi

- Trừu tượng hóa cấu trúc hàng đợi
 - Đặc tả dữ liệu
$$A = (a_0, a_1, \dots, a_n)$$
trong đó a_0 là đầu hàng đợi, a_n là cuối hàng đợi
 - Đặc tả các phép toán
 1. Thêm phần tử x vào cuối hàng đợi: `enqueue(x)`
 2. Loại phần tử ở đầu hàng đợi: `dequeue()`
 3. Kiểm tra hàng đợi có rỗng hay không: `isEmpty()`
 4. Kiểm tra hàng đợi hết chỗ hay chưa: `isFull()`
 5. Đếm số phần tử của hàng đợi: `size()`
 6. Trả về phần tử ở đầu hàng đợi: `front()`

Giao diện C++ của KDLETT hàng đợi

```
template <typename Object>
class Queue {
public:
    int size();
    bool isEmpty();
    Object& front()
        throw(EmptyQueueException);
    void enqueue(Object o);
    Object dequeue()
        throw(EmptyQueueException);
};
```

Minh họa các thao tác

thao tác	output	hàng đợi
enqueue(10)		(10)
enqueue(5)		(10, 5)
front()	10	(10, 5)
dequeue()		(5)
size()	1	(5)
dequeue()		()
front()	lỗi: hàng đợi rỗng	()
dequeue()	lỗi: hàng đợi rỗng	()
isEmpty()	true	()
enqueue(8)		(8)

Ứng dụng của hàng đợi

- Trực tiếp
 - Danh sách hàng đợi
 - Quản lý truy cập tới các tài nguyên dùng chung (ví dụ máy in)
 - Multiprogramming
- Gián tiếp
 - Cấu trúc dữ liệu phụ trợ cho các thuật toán
 - Một phần của CTDL khác

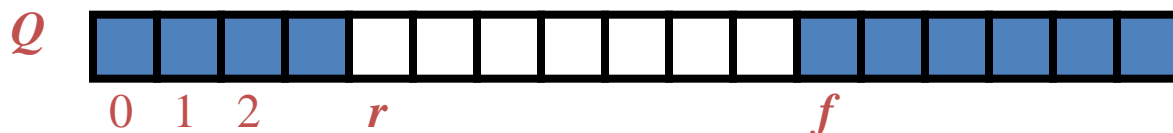
Cài đặt hàng đợi bởi mảng

- Dùng một mảng cỡ N theo kiểu vòng tròn
- Dùng 2 biến để theo dõi đầu (front) và đuôi (rear) hàng đợi
 - f là chỉ số của phần tử front
 - r là chỉ số của ô liền sau phần tử rear
- Ô r trong mảng sẽ luôn rỗng

cấu hình bình thường



cấu hình bọc vòng quanh



Các thao tác

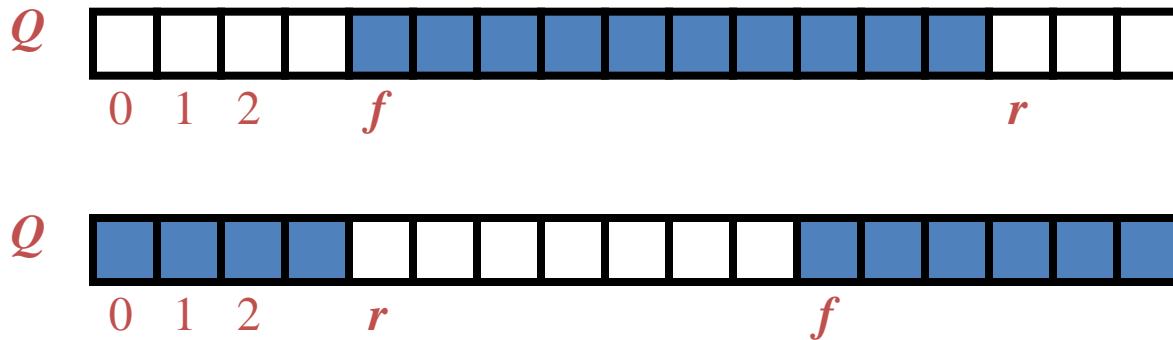
- Ta sử dụng phép chia lấy dư

Algorithm *size()*

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

Algorithm *isEmpty()*

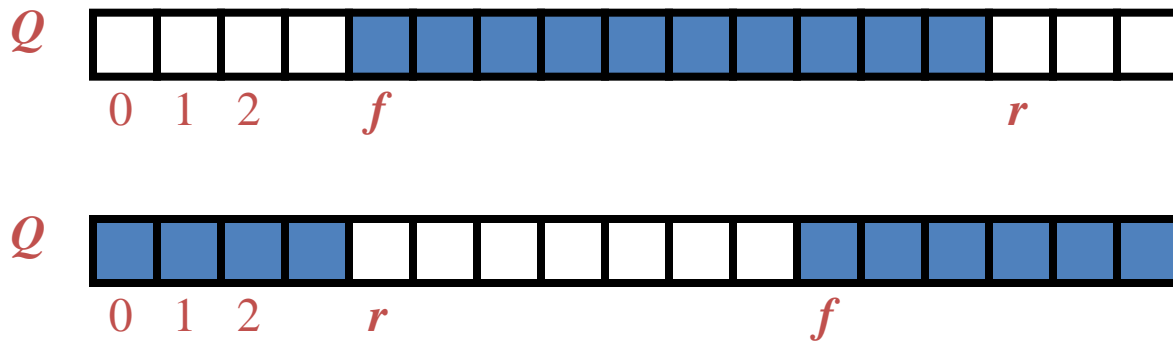
return $(f = r)$



Các thao tác (2)

- Thao tác enqueue ném một ngoại lệ nếu mảng đã đầy
- Đây là ngoại lệ do cài đặt

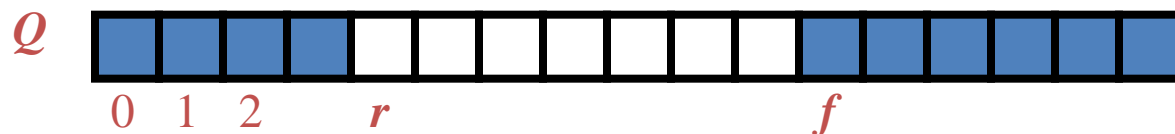
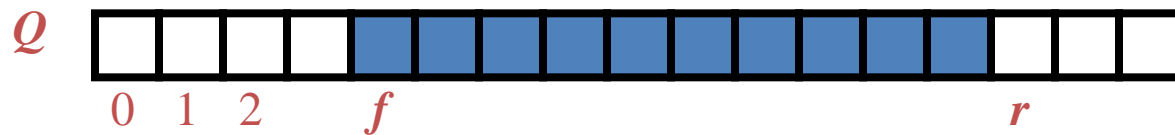
```
Algorithm enqueue(o)  
  if size() =  $N - 1$  then  
    throw FullQueueException  
  else  
     $Q[r] \leftarrow o$   
     $r \leftarrow (r + 1) \bmod N$ 
```



Các thao tác (3)

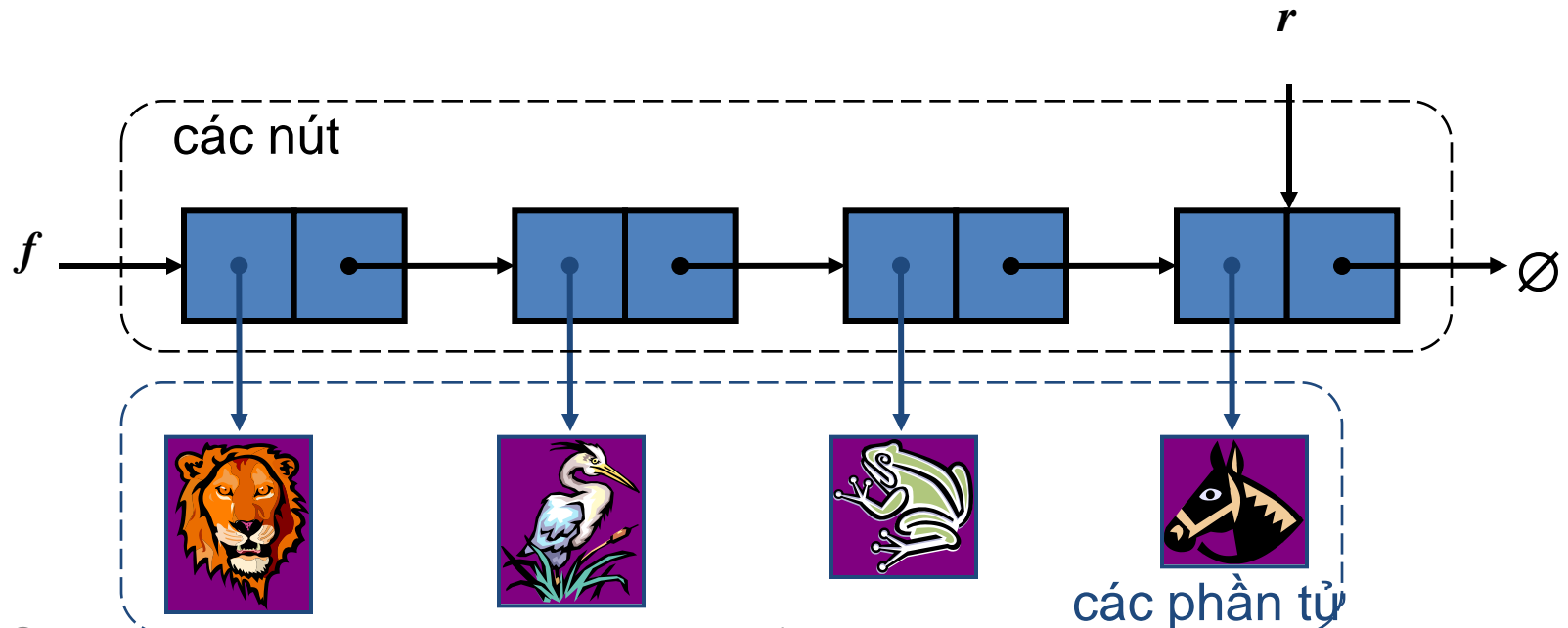
- Thao tác dequeue ném ngoại lệ nếu hàng đợi rỗng
- Đây là ngoại lệ xác định cho KDLTT hàng đợi

```
Algorithm dequeue()  
  if isEmpty() then  
    throw EmptyQueueException  
  else  
     $o \leftarrow Q[f]$   
     $f \leftarrow (f + 1) \bmod N$   
    return  $o$ 
```



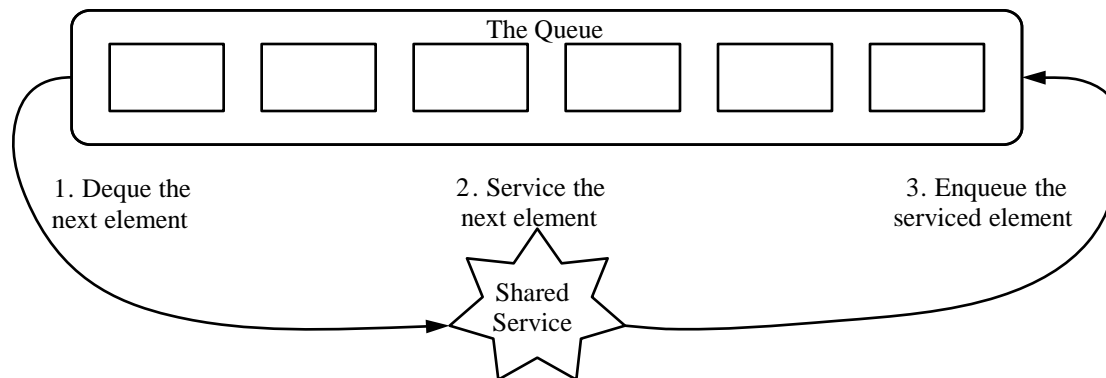
Cài đặt hàng đợi bởi DSLK đơn

- Ta có thể cài đặt hàng đợi bởi một danh sách liên kết đơn
 - Phần tử front được lưu ở nút đầu
 - Phần tử rear được lưu ở nút cuối
- Không gian sử dụng là $O(n)$ và mỗi thao tác thực hiện trong thời gian $O(1)$



Ứng dụng: Lập lịch quay vòng (Round Robin Schedulers)

- Có thể cài đặt một bộ lập lịch quay vòng bằng một hàng đợi, Q , bằng việc lặp lại các bước sau:
 1. $e = Q.dequeue()$
 2. Service element e
 3. $Q.enqueue(e)$



Bài tập

1. Viết chương trình cài đặt cấu trúc hàng đợi bằng mảng.
2. Viết chương trình cài đặt cấu trúc hàng đợi bằng danh sách liên kết đơn.
3. Tính độ phức tạp cho cài đặt ở câu 1, 2
4. Cài đặt hàng đợi bằng mảng vòng.

Chuẩn bị bài tới

- Đọc Chương 8 giáo trình (Cây)