

Chương 4. HÀNG ĐỘI

ThS. Nguyễn Chí Hiếu

2017

Giới thiệu hàng đợi

Cài đặt hàng đợi

Ứng dụng của hàng đợi

Hàng đợi (Queue)

- ▶ Thực hiện theo cơ chế FIFO (*First In, First Out*) vào trước ra trước.
- ▶ Dùng để lưu trữ các phần tử có thứ tự truy xuất **đúng** với thứ tự lưu trữ (vào trước, ra trước).



Hình 1: Hình minh họa hàng đợi.

Giới thiệu hàng đợi

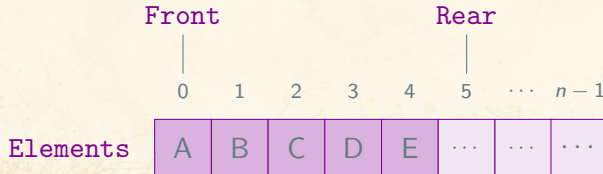
Các thao tác cơ bản

- ▶ EnQueue: *thêm* phần tử vào *cuối* hàng đợi.
- ▶ DeQueue: *lấy và xóa* phần tử tại *đầu* hàng đợi.
- ▶ GetFront: *xem thông tin* phần tử tại *đầu* hàng đợi.
- ▶ Kiểm tra hàng đợi rỗng, đầy.

Cài đặt hàng đợi bằng mảng

- Biến Elements là mảng 1 chiều kích thước n : lưu trữ phần tử từ vị trí $[0, \dots, n - 1]$.
- Biến Front, Rear kiểu số nguyên: cho biết vị trí đầu và cuối.
- Mặc định, hàng đợi vừa khởi tạo $\text{Front} = \text{Rear} = 0$ và tất cả phần tử của mảng Elements gán bằng NULL_DATA .

```
1 struct Queue
2 {
3     Data    Elements[MAX_SIZE];
4     int     Front;
5     int     Rear;
6 };
```



Cài đặt hàng đợi bằng mảng

Thuật toán 1: IsEmpty(q)

- Đầu vào: hàng đợi q.
- Đầu ra: true/false.

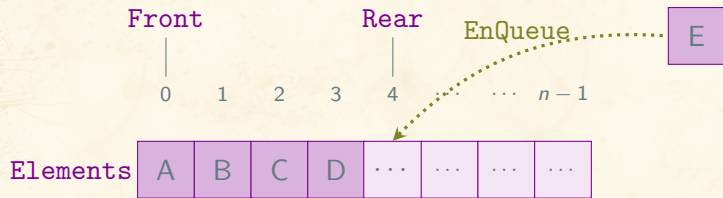
```
1     if Elements[Front] = NULL_DATA
2         return true
3     return false
```

Thuật toán 2: IsFull(q)

- Đầu vào: hàng đợi q.
- Đầu ra: true/false.

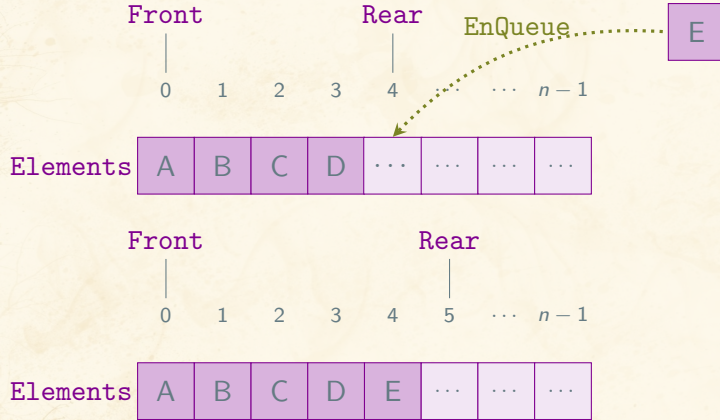
```
1     if Elements[Rear] ≠ NULL_DATA
2         return true
3     return false
```

Cài đặt hàng đợi bằng mảng



Cài đặt hàng đợi bằng mảng

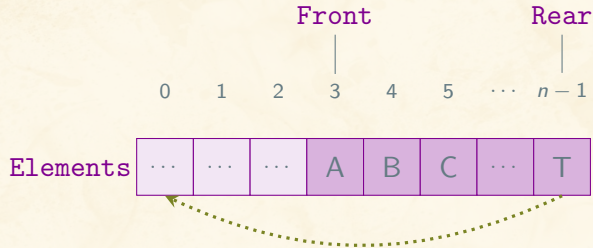
duyetho



Cài đặt hàng đợi bằng mảng

Xử lý vấn đề tràn giả

Sử dụng mảng như danh sách vòng.



- EnQueue: nếu đến cuối mảng, cập nhật $\text{Rear} = 0$.

Cài đặt hàng đợi bằng mảng

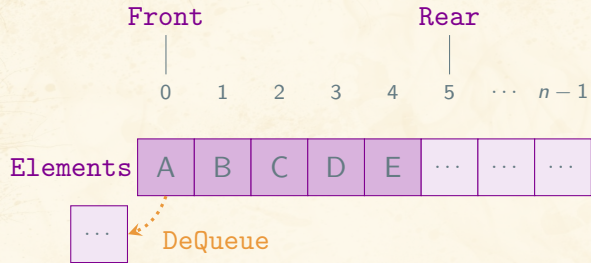
Thuật toán 3: EnQueue(q, x)

- Đầu vào: hàng đợi q và phần tử x cần thêm.
- Đầu ra: hàng đợi q sau khi thêm x.

```
1  if hàng đợi chưa đầy
2      Elements[Rear] ← x
3      Rear ← Rear + 1
4      if Rear = MAX_SIZE
5          Rear ← 0
```

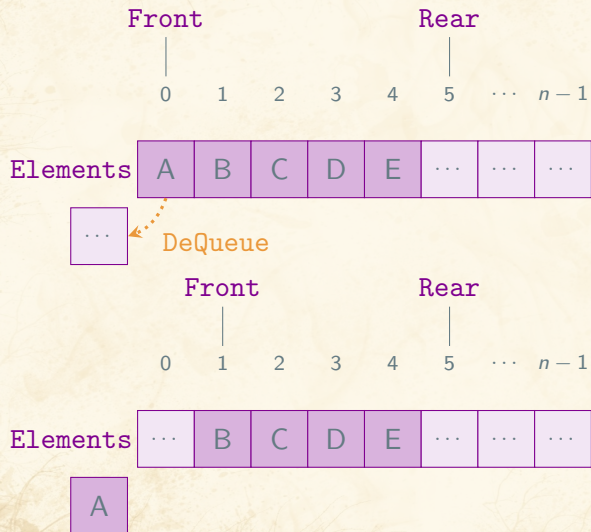
Cài đặt hàng đợi bằng mảng

duyetho



Cài đặt hàng đợi bằng mảng

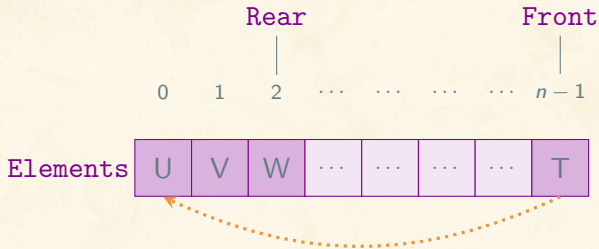
duyethoang



Cài đặt hàng đợi bằng mảng

Xử lý vấn đề tràn giả

Sử dụng mảng như danh sách vòng.



- DeQueue: nếu đến cuối mảng, cập nhật $\text{Front} = 0$.

Cài đặt hàng đợi bằng mảng

Thuật toán 4: DeQueue(q)

- Đầu vào: hàng đợi q.
- Đầu ra: phần tử đầu hàng đợi hay NULL_DATA (hàng đợi rỗng).

```
1  if hàng đợi khác rỗng
2      x ← Elements[Front]
3      Elements[Front] ← NULL_DATA
4      Front ← Front + 1
5      if Front = MAX_SIZE
6          Front ← 0
7      return x
8  return NULL_DATA
```

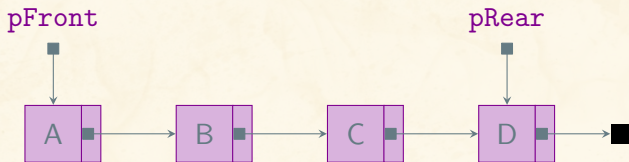
Cài đặt hàng đợi bằng mảng

Thuật toán 5: GetFront(q)

```
1   if hàng đợi khác rỗng
2       return Elements[Front]
3   return NULL_DATA
```

Cài đặt hàng đợi bằng danh sách liên kết

- ▶ Cấu trúc dữ liệu một phần tử của hàng đợi chứa thành phần dữ liệu và thành phần liên kết (*tương tự danh sách liên kết*).
- ▶ Cấu trúc dữ liệu hàng đợi chứa hai con trỏ pFront trỏ đến phần tử **đầu** và con trỏ pRear trỏ đến phần tử **cuối** của hàng đợi.
- ▶ Thao tác thêm thực hiện ở cuối và thao tác xóa thực hiện ở đầu hàng đợi.



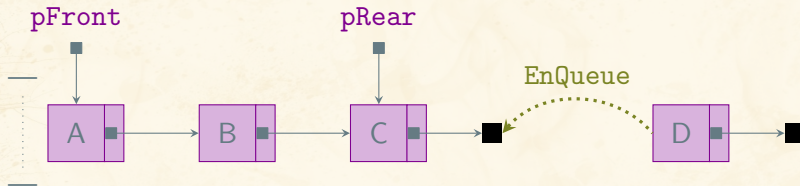
Cài đặt hàng đợi bằng danh sách liên kết

```
1 typedef int Data;  
2 struct Node  
3 {  
4     Data Info;  
5     Node *pNext;  
6 };
```

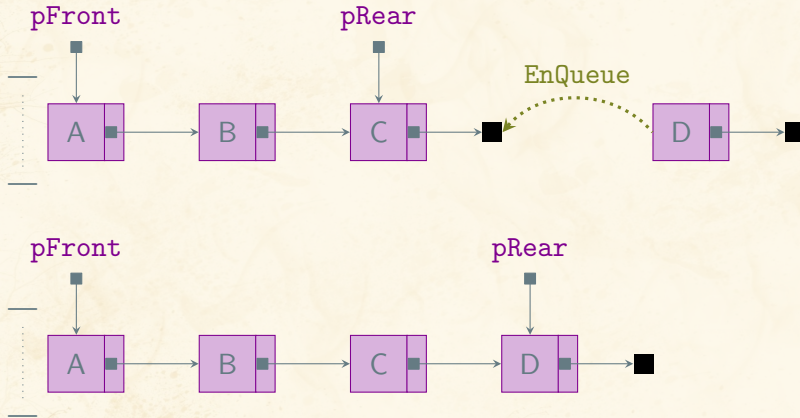
```
1 struct Queue  
2 {  
3     int Count; // Dem so phan tu trong hang doi  
4     Node *pFront; // pHead  
5     Node *pRear; // pTail  
6 };
```

Cài đặt hàng đợi bằng danh sách liên kết

duyethoang



Cài đặt hàng đợi bằng danh sách liên kết

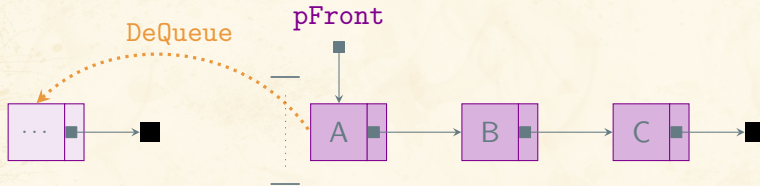


Thao tác thêm phần tử

```
1 // Thêm phần tử vào hàng đợi (thêm cuối)
2 void EnQueue(Queue *q, Data x)
3 {
4     Node *p = InitNode(x);
5     if (q->pFront == NULL) // TH1. Hàng đợi rỗng
6         q->pFront = q->pRear = p;
7     else // TH2. Hàng đợi khác rỗng
8     {
9         q->pRear->pNext = p;
10        q->pRear = p;
11    }
12    q->Count++;
13 }
```

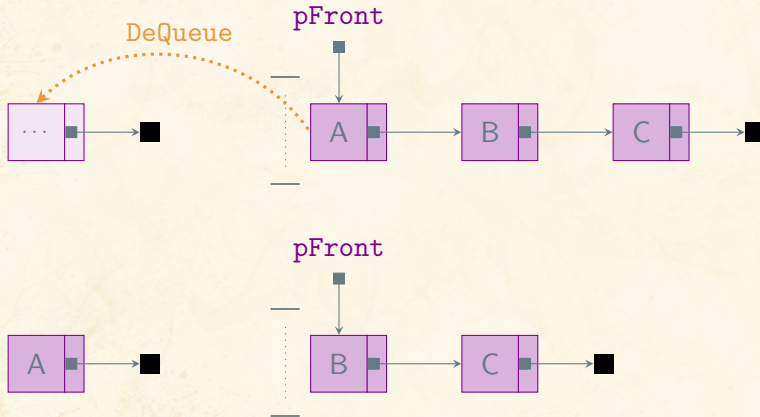
Cài đặt hàng đợi bằng danh sách liên kết

duyethoang



Cài đặt hàng đợi bằng danh sách liên kết

duyetho



Thao tác lấy phần tử

```
1 // Lấy phần tử ở đầu hàng đợi (xóa đầu)
2 Data DeQueue(Queue *q)
3 {
4     Node *p = new Node();
5     if (IsEmpty(q) == true) // TH1. Hàng đợi rỗng
6         return NULL_DATA;
7     // TH2. Hàng đợi khác rỗng
8     p = q->pFront;
9     q->pFront = q->pFront->pNext;
10    q->Count--;
11    Data x = p->Info;
12    delete p;
13    return x;
14 }
```


Thao tác lấy phần tử

```
1 Data GetFront(Queue *q)
2 {
3     Node *p = new Node();
4     if (IsEmpty(q) == true) // TH1. Hàng đôi rỗng
5         return NULL_DATA;
6     // TH2. Hàng đôi khác rỗng
7     p = q->pFront;
8
9     return p->Info;
10 }
```

Ứng dụng của hàng đợi

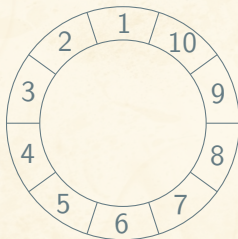
Một số ứng dụng của hàng đợi

- ▶ Trong một số thuật toán của lý thuyết đồ thị, hàng đợi được sử dụng để lưu dữ liệu khi thực hiện.
- ▶ Bài toán sản xuất và tiêu thụ.
- ▶ Quản lý bộ đệm (ví dụ: nhấn phím \rightarrow bộ đệm \rightarrow CPU xử lý).
- ▶ Xử lý các lệnh/tiến trình trong máy tính (ví dụ: hàng đợi máy in)

Ứng dụng của hàng đợi

Ví dụ 1 (Bài toán Josephus)

Cho n người đứng thành vòng tròn và một số nguyên m , với $m < n$.



► Giả sử

► $n = 10$

► $m = 3$

- Bắt đầu vị trí s , bài toán sẽ đếm từng người theo một chiều nhất định. Sau khi có $m - 1$ người được bỏ qua, người thứ m sẽ bị xử tử.
- Quy luật lặp lại đến khi còn $m - 1$ người sống sót.

Câu hỏi $m - 1$ người còn sống đứng vị trí nào ?

Bài toán Josephus

Thuật toán 6: Josephus(n , m)

- Đầu vào: n là số người và m là một số nguyên
- Đầu ra: in ra thứ tự người bị xử tử

```
1  for i ← 1 to n
2      EnQueue(q, i);
3
4  while q ≠ ∅
5      for i ← 1 to m - 1
6          EnQueue(q, DeQueue(q));
7      x ← DeQueue(q)
8      Print x
```

Giải thích

- ▶ Dòng 1 \rightarrow 2: đưa tất cả người tham gia vào hàng đợi.
- ▶ Dòng 4 \rightarrow 7: hàng đợi khác rỗng, bắt đầu đếm và thực hiện
 - ▶ Dòng 5 \rightarrow 6: đưa $m - 1$ người vào hàng đợi.
 - ▶ Dòng 7: chọn người vị trí m .
- ▶ Dòng 8: in thứ tự người bị chọn (trong đó, hai người ở vị trí cuối cùng sẽ sống sót)

1. Cài đặt hàng đợi sử dụng mảng và danh sách liên kết.
2. Cài đặt phiên bản hàng đợi sử dụng hai thao tác *thêm đầu* và *xóa cuối* danh sách.
3. Áp dụng hàng đợi cài đặt thuật toán Josephus.



Dương Anh Đức, Trần Hạnh Nhi.

Nhập môn Cấu trúc dữ liệu và Thuật toán.

Đại học Khoa học tự nhiên TP Hồ Chí Minh, 2003.



Donald E. Knuth.

The Art of Computer Programming, Volume 3.

Addison-Wesley, 1998.



Niklaus Wirth.

Algorithms + Data Structures = Programs.

Prentice-Hall, 1976.



Robert Sedgewick.

Algorithms in C.

Addison-Wesley, 1990.