

## Tìm hiểu một cấu trúc dữ liệu

- 1) Đặc điểm, tổ chức
- 2) Cấu trúc lưu trữ
- 3) Phép toán

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

3.1

1

## CHƯƠNG 2 MẢNG VÀ DANH SÁCH

### 1. Mảng

- Mảng là một tập hợp có thứ tự gồm một số cố định các phần tử cùng kiểu.
- Mỗi phần tử mảng được chỉ ra bởi chỉ số, thể hiện thứ tự của phần tử trong mảng. Điều này cho phép truy nhập trực tiếp phần tử qua chỉ số.
- Các phần tử của mảng có thể tổ chức thành mảng 1 chiều, 2 chiều, 3 chiều... Mảng 1 chiều có 1 chỉ số, mảng 2 chiều có 2 chỉ số, ... mảng n chiều có n chỉ số.

2

# 1. Mảng

- Mảng chỉ dùng được cấu trúc lưu trữ kế tiếp, để cho phép truy nhập trực tiếp các phần tử.
- Dùng vec tơ lưu trữ  $V$  có  $n$  ô nhớ liên tiếp với chỉ số từ 1 đến  $n$  để lưu trữ các phần tử dữ liệu của mảng.
- Với mảng 1 chiều, phần tử  $a_i$  được lưu trữ ở ô nhớ  $V[i]$
- Với mảng 2 chiều, các phần tử được lưu trữ lần lượt, hết hàng 1 đến hàng 2... Phần tử  $a_{ij}$  được lưu trữ ở ô nhớ  $V[k]$ ,  $k = (i-1)*n + j$ ,  $n$  là số phần tử trên hàng.

3

# 1. Mảng

V	x	x	x					x
k	1	2	3					n

4	5	9
7	10	1

$$k = (i-1)*n + j$$

$$a_{13} \Rightarrow V[k],$$

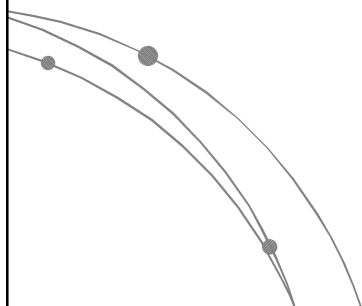
$$k = (1-1)*3 + 3 = 3$$

V	4	5	9	7	10	1
k	1	2	3	4	5	6

4

# 1. Mảng

- Có các phép tạo lập mảng, tìm kiếm 1 phần tử từ mảng, truy nhập một phần tử mảng.
- Không có phép bổ sung và loại bỏ một phần tử mảng.

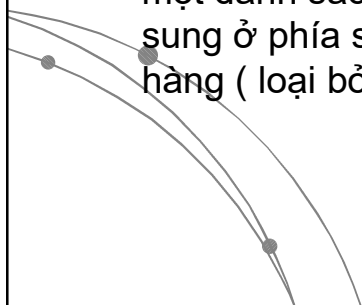


5

# 2. Danh sách

## 2.1. Khái niệm

- Danh sách là một tập hợp có thứ tự gồm một số biến động các phần tử cùng kiểu.
- Ví dụ: Tập hợp người đến khám bệnh cho ta một danh sách. Người đến xếp hàng khám bổ sung ở phía sau, người được khám sẽ ra khỏi hàng ( loại bỏ ) ở phía trước.



6

## 2.1. Khái niệm

- Danh sách tuyến tính là danh sách mà mỗi quan hệ lân cận giữa các phần tử được xác định rõ ràng. Ví dụ: Véc tơ là một danh sách tuyến tính.
- Danh sách tuyến tính có dạng  $(a_1, a_2, \dots, a_n)$ , trong đó  $a_1$  là phần tử đầu,  $a_n$  là phần tử cuối, phần tử thứ  $i$  là  $a_i$ . Với  $a_i$  bất kỳ  $1 \leq i \leq n$  thì  $a_{i+1}$  gọi là phần tử sau  $a_i$ ;  $2 \leq i \leq n$  thì phần tử  $a_{i-1}$  là phần tử trước của  $a_i$ .
- Danh sách có thể rỗng (không có phần tử nào).

7

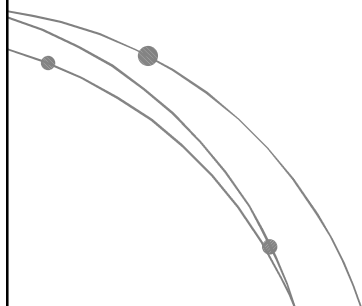
## 2.1. Khái niệm

- $n$  là độ dài (kích thước) của danh sách,  $n$  có thể thay đổi.
- Một phần tử trong danh sách thường là một bản ghi (gồm một hoặc nhiều trường).
- Ví dụ 1: Danh mục điện thoại là một danh sách tuyến tính, mỗi phần tử của nó là một thuê bao gồm 3 trường: Họ tên chủ hộ, địa chỉ, số điện thoại.
- Ví dụ 2: Tập(File) là một danh sách có kích thước lớn được lưu trữ ở bộ nhớ ngoài.

8

## 2.2. Lưu trữ kế tiếp cho danh sách tuyến tính

- Danh sách có thể sử dụng cấu trúc lưu trữ kế tiếp hoặc phân tán.



9

## 2.3. Các phép toán trên danh sách

- Danh sách luôn có phép toán bổ sung, loại bỏ phần tử dữ liệu.
- Phép bổ sung: Có thể bổ sung phần tử vào danh sách.
- Phép loại bỏ: có thể loại bỏ một phần tử ra khỏi danh sách.
- Phép ghép: có thể ghép hai hay nhiều danh sách thành một danh sách.
- Phép tách: có thể tách một danh sách thành nhiều danh sách.
- Phép cập nhật: cập nhật giá trị cho các phần tử của danh sách.

10

## 2.3. Các phép toán trên danh sách

- Phép sao chép: có thể sao chép một danh sách.
- Phép sắp xếp: Có thể sắp xếp các phần tử của danh sách theo một thứ tự nhất định.
- Phép tìm kiếm: Tìm kiếm trong danh sách một phần tử mà một trường nào đó có giá trị ấn định.

Ví dụ 1: Minh họa cho các phép toán trên danh sách được cài đặt trên mảng. Cho danh sách các số nguyên, thêm vào 1 số nguyên và loại bỏ một số nguyên.

11

## 3. Cấu trúc ngăn xếp (Stack)

### 3.1. Định nghĩa

- Ngăn xếp là một loại danh sách tuyến tính đặc biệt mà phép bổ sung và phép loại bỏ luôn luôn thực hiện ở một đầu gọi là đỉnh (Top).
- Phép bổ sung và loại bỏ phần tử của ngăn xếp được thực hiện theo nguyên tắc 'Vào sau ra trước' (Last in - First out, viết tắt là LIFO).
- Ngăn xếp có thể rỗng.

12

## 3.2. Ngăn xếp lưu trữ kế tiếp

- Dùng vector lưu trữ  $S$  có  $n$  ô nhớ kế tiếp nhau với chỉ số từ 1 đến  $n$  để lưu các phần tử dữ liệu.
- Dùng biến  $T$  chứa chỉ số phần tử đỉnh của ngăn xếp.  $T$  sẽ có giá trị thay đổi khi ngăn xếp hoạt động. Khi bổ sung 1 phần tử  $T$  sẽ tăng lên 1. Khi loại bỏ 1 phần tử thì  $T$  giảm đi 1.
- Khi ngăn xếp rỗng  $T=0$ .



13

## 3.3. Các phép toán trên ngăn xếp

- Bổ sung một phần tử vào stack
  - Vào: ngăn xếp  $(S, T)$ , phần tử  $x$
  - Ra: không có

{Thủ tục này bổ sung phần tử  $x$  vào ngăn xếp được lưu trữ bởi véc tơ  $S$  có kích thước là  $n$ , có chỉ số đỉnh là  $T$ .}

14

## Thủ tục bổ sung một phần tử vào stack

Procedure push(Var S, T; x)

1) {Kiểm tra đầy}

    If T = n then Begin Write('Ngăn xếp đã đầy')

        Return

    End;

2) {Tăng T lên 1}

    T := T+1;

3) {Đưa x vào ngăn xếp tại vị trí T}

    S[T] := x;

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 03

3.15

15

## 3.3. Các phép toán trên ngăn xếp

- Loại bỏ một phần tử ra khỏi stack

- Vào: Ngăn xếp (S, T)

- Ra: giá trị phần tử loại bỏ (đỉnh)

{Hàm này thực hiện việc loại bỏ phần tử ở đỉnh ngăn xếp (S,T) và trả về phần tử này.}

16



## Hàm loại bỏ phần tử khỏi ngăn xếp

```

Function pop(Var S, T)
  1) {Kiểm tra rỗng}
    If T = 0 then Begin
      Write('Ngăn xếp rỗng')
      Return;
    End
  2) tg := S[T]; {Lưu lại phần tử đỉnh}
  3) {Giảm T đi 1}
    T := T-1
  4) {Trả về phần tử đỉnh đã loại}
    pop := tg;

```

Return  
Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

2.17

17

## Hàm kiểm tra ngăn xếp rỗng/ đầy

```

Function isEmpty(S,T)
  If T = 0 then isEmpty := TRUE
  Else isEmpty := FALSE;

```

Return

```

Function isFull(S,T)
  If T = n then isFull := TRUE
  Else isFull := FALSE;

```

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

2.18

18

## Hàm trả về phần tử đỉnh ngăn xếp

Function top(S,T)

1) {Kiểm tra rỗng}

If T = 0 then Begin

Write('Ngăn xếp rỗng')

Return;

End

2) {Trả về phần tử đỉnh}

top := S[T];

Return

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

2.19

19

## Cài đặt cấu trúc dữ liệu

- Khi cài đặt một cấu trúc dữ liệu ta cài đặt những gì?

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

3.20

20

## Cài đặt cấu trúc dữ liệu

Cấu trúc dữ liệu + Giải thuật = Chương trình

$\begin{array}{c} || \\ \text{Cấu trúc lưu trữ} \\ (1) \end{array}$ 
 $\begin{array}{c} || \\ \text{Phép toán} \\ (2) \end{array}$

Cài đặt (Lập trình)

- 1) Cài đặt cấu trúc lưu trữ: Khai báo biến và khởi tạo giá trị ban đầu cho biến (nếu cần)
- 2) Cài đặt phép toán: Khai báo và định nghĩa hàm, mỗi hàm là một phép toán

21

## Ví dụ về ứng dụng của ngăn xếp

- (ctdlgt-nganxep1.cpp): Cài đặt cấu trúc dữ liệu ngăn xếp lưu trữ kế tiếp có phần tử là số nguyên. Ứng dụng ngăn xếp đã cài đặt cho bài toán chuyển đổi số nguyên hệ 10 sang hệ 2.

22

## Ví dụ về ứng dụng của ngăn xếp

```

- Vào: n
- Ra: số nhị phân
Procedure chuyen_doi (n);
  While n <> 0 do Begin
    R:=n mod 2
    Call Push(S,T,R);
    n= n div 2
  End;
  While S <> NULL do Begin
    R:=POP(S,T); {lay R tu dinh T cua Stack S }
    Write(R)
  End;
Return;

```

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

3.23

23

## Bài tập

- Ứng dụng ngăn xếp chuyển biểu thức trung tố hành hậu tố. Biết rằng biểu thức trung tố có dấu ngoặc đầy đủ.

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

3.24

24

## 4. Cấu trúc hàng đợi (Queue)

### 4.1. Định nghĩa

- Hàng đợi (queue) là loại danh sách tuyến tính đặc biệt mà phép bổ sung được thực hiện ở một đầu, gọi là lối sau (rear) và phép loại bỏ thực hiện ở một đầu khác, gọi là lối trước (front).
- Phép bổ sung và loại bỏ phần tử của hàng đợi được thực hiện theo nguyên tắc vào trước ra trước (First in - First out, viết tắt là FIFO).
- Hàng đợi có thể rỗng.

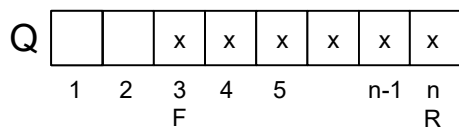
25

### 4.2. Hàng đợi lưu trữ kế tiếp

- Dùng vector lưu trữ  $Q$  có  $n$  ô nhớ kế tiếp với chỉ số từ 1 đến  $n$  để lưu trữ các phần tử dữ liệu của hàng đợi.
- Dùng biến  $F$  chứa chỉ số phần tử đầu hàng (lối trước), biến  $R$  chứa chỉ số phần tử cuối hàng (lối sau).  $F$  (Front),  $R$  (Rear).
- Khi bổ sung 1 phần tử vào hàng đợi thì  $R$  sẽ tăng lên 1, còn khi loại bỏ một phần tử khỏi hàng đợi thì  $F$  tăng lên 1.
- Khi hàng đợi rỗng thì  $R=F=0$ .

26

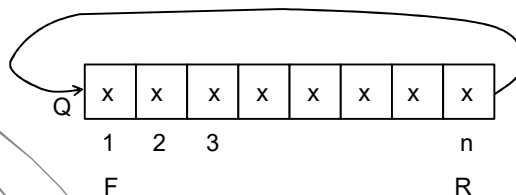
## 4.2. Hàng đợi lưu trữ kế tiếp



27

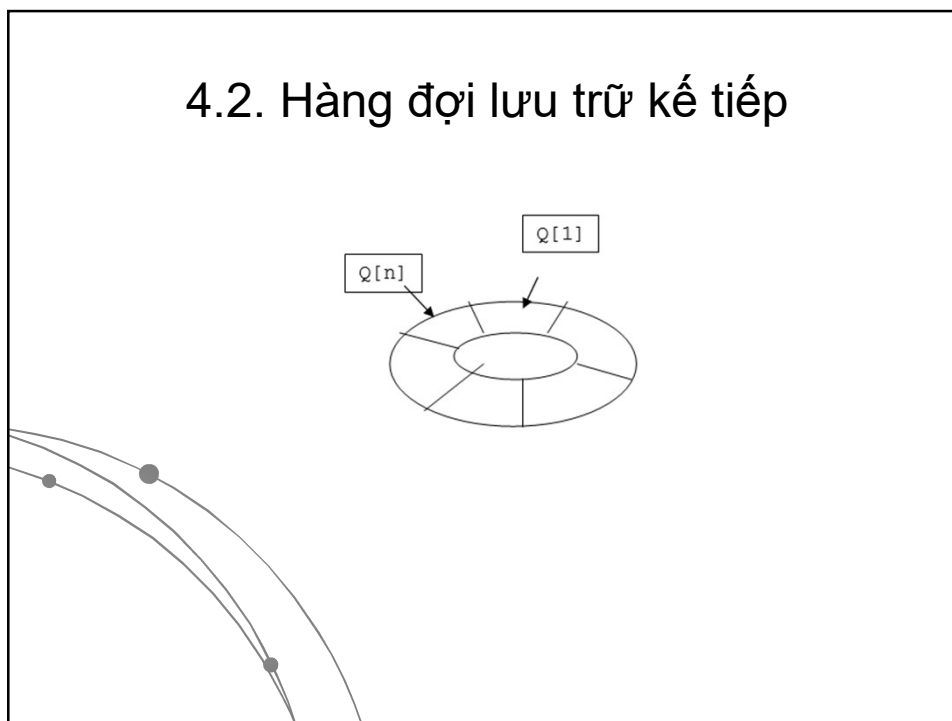
## 4.2. Hàng đợi lưu trữ kế tiếp

- Trong cấu trúc lưu trữ kế tiếp của hàng đợi người ta sử dụng ô nhớ theo kiểu quay vòng để sử dụng lại các ô nhớ chứa các phần tử dữ liệu đã loại bỏ, tức là tiếp theo ô nhớ n là ô nhớ 1.



28

## 4.2. Hàng đợi lưu trữ kế tiếp



29

## 4.3. Các phép toán trên Queue

a) Bổ sung một phần tử vào queue

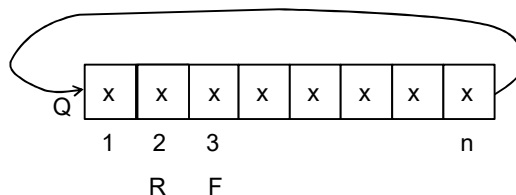
- Vào:  $(Q, F, R), x$

- Ra: Không có

{Thủ tục này bổ sung phần tử  $x$  vào hàng đợi lưu trữ bởi vector  $Q$  có  $n$  ô nhớ theo kiểu vòng tròn, có biến chỉ số  $F, R$ .}

30

## Trường hợp hàng đợi đầy



Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

3.31

31

```

Procedure CQInsert (Var Q,F,R; x)
  1) {Kiểm tra đầy}
    If (F=1)and(R=n) or (R+1=F) then Begin
      Write( 'Hàng đợi đã đầy');
      Return;
    End;
  2) {Tăng R lên 1}
    If F=R=0 Then F := R :=1
    Else If R=n Then R := 1
    Else R := R+1;
  3. {Đưa x vào hàng đợi tại vị trí R}
    Q[R] := x;
  Return
  
```

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

3.32

32



### 4.3. Các phép toán trên hàng đợi

b) Loại bỏ phần tử ra khỏi hàng đợi

- Vào: Hàng đợi (Q,F,R)

- Ra: Trả về phần tử loại bỏ

{Hàm này loại bỏ phần tử ở lối trước của hàng đợi (Q,F,R) và trả về phần tử loại bỏ}

33

Thủ tục loại bỏ phần tử khỏi hàng đợi

Function CQDelete(Var Q,F,R)

1) {Kiểm tra rỗng}

If F=R=0 then Begin

Write('Hàng đợi đã rỗng');

Return;

End;

2) {Lưu lại phần tử loại bỏ}

tg:=Q[F];

Ngô Công Thắng

Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02

3.34

34

Thủ tục loại bỏ phần tử khỏi hàng đợi

Function CQDelete(Var Q,F,R)

3) {Tăng F lên 1}

    If F=R then F:=R:=0

    Else If F=n then F:=1

    Else F:=F+1;

4) CQDelete:=tg;

Return

Ngô Công Thắng      Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02      3.35

35

Phép toán kiểm tra hàng đợi rỗng

- Vào: (Q,F,R)

- Ra: TRUE nếu rỗng còn không là FALSE

Function CQIsEmpty(Q,F,R)

    If F=R=0 then CQIsEmpty:=True

    Else CQIsEmpty:=False;

Return

Ngô Công Thắng      Bài giảng Cấu trúc dữ liệu và giải thuật - Chương 02      3.36

36

## Bài tập

Cài đặt cấu trúc dữ liệu hàng đợi lưu trữ kế tiếp có phần tử dữ liệu là số nguyên. Ứng dụng hàng đợi cho bài toán: Đọc dãy số nguyên dương từ tệp văn bản 'daysonguyen.txt', trên tệp không có thông tin về số lượng số, tách thành dãy số lẻ và dãy số chẵn theo đúng thứ tự xuất hiện trên tệp, đưa 2 dãy số ra màn hình.