

Cấu trúc dữ liệu và giải thuật

CÁC CẤU TRÚC DỮ LIỆU CƠ BẢN

Giảng viên:

Văn Chí Nam – Nguyễn Thị Hồng Nhung – Đặng Nguyễn Đức Tiến

Nội dung trình bày

2

Danh sách liên kết

Ngăn xếp

Hàng đợi

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

3

Danh sách liên kết

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Nội dung

4

- ◉ Giới thiệu
- ◉ Các loại danh sách liên kết
- ◉ Các thao tác trên danh sách liên kết
- ◉ So sánh danh sách liên kết và mảng
- ◉ Ứng dụng

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Giới thiệu

5

- ◉ **Mảng: cấu trúc dữ liệu quen thuộc**
 - ▣ Tập có thứ tự
 - ▣ Số lượng phần tử cố định (tĩnh)
 - ▣ Cấp phát vùng nhớ liên tục
 - ▣ Truy xuất phần tử thông qua chỉ số

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Giới thiệu

6

- ◉ **Đánh giá thao tác trên mảng:**
 - ▣ Truy xuất phần tử?
 - ▣ Cập nhật?
 - ▣ Chèn phần tử?
 - ▣ Xoá phần tử?

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Giới thiệu

7

◉ Thực tế:

- ▣ Không xác định được chính xác số lượng phần tử
 - Danh sách bệnh nhân: tăng/giảm.
 - Danh sách sinh viên: tăng/giảm.
- ▣ Vùng nhớ thay đổi trong quá trình sử dụng
=> Không đủ vùng nhớ cấp phát liên tục.

=> Cấu trúc dữ liệu động đáp ứng nhu cầu

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Các loại danh sách liên kết

8

◉ Danh sách liên kết đơn

- ▣ singly linked list
- ▣ uni-directional linked list

◉ Danh sách liên kết kép

- ▣ doubly linked list
- ▣ bi-directional linked list

◉ Danh sách liên kết vòng

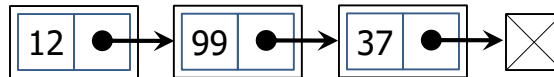
- ▣ circularly linked list
- ▣ ring list

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Danh sách liên kết đơn

9

- ◉ Mỗi phần tử có MỘT liên kết đến phần tử phía sau nó.

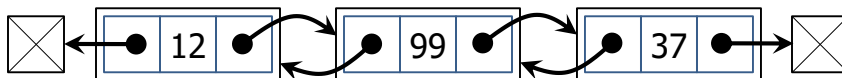


Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Danh sách liên kết kép

10

- ◉ Mỗi phần tử có HAI liên kết đến phần tử đứng sau và trước nó.

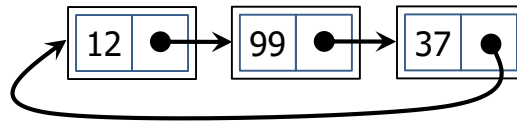


Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Danh sách liên kết vòng

11

- ◉ Có mỗi liên kết giữa phần tử cuối và phần tử đầu



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

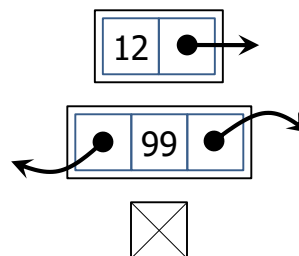
Phần tử trên danh sách liên kết

12

- ◉ Phần tử (Node, Element)
 - ▣ Phần tử = Dữ liệu + Liên kết

▣ Ví dụ:

- Phần tử có 1 liên kết
- Phần tử có 2 liên kết
- Phần tử rỗng



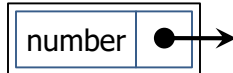
Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Ví dụ

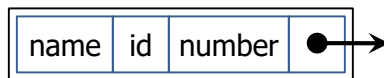
13

◉ Ví dụ:

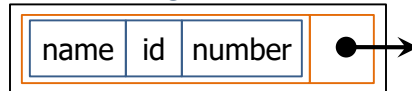
- ▣ Phần tử có dữ liệu gồm 1 thành phần



- ▣ Phần tử có dữ liệu gồm 3 thành phần



- ▣ Phần tử có dữ liệu gồm 1 cấu trúc



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Cài đặt

14

- ◉ Phần cài đặt cho các ví dụ trên

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Tổ chức

15

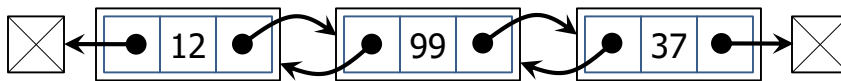
- ◉ Mỗi danh sách liên kết bao gồm:

- ▣ Con trỏ đến phần tử đầu (hoặc/và cuối) danh sách.

- ▣ (Các) phần tử trên danh sách

- Dữ liệu

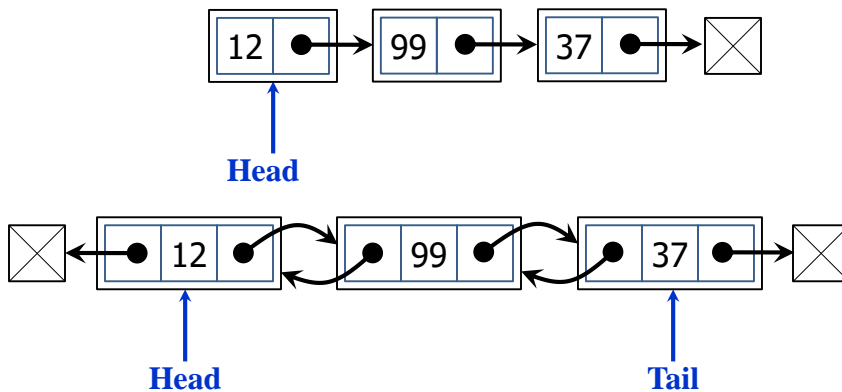
- Các mối liên kết



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Tổ chức

16



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Các thao tác trên danh sách liên kết

17

- ◉ Thêm phần tử
- ◉ Duyệt danh sách
- ◉ Xoá phần tử
- ◉ Truy xuất phần tử
- ◉ Xoá danh sách

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Thêm phần tử

18

- ◉ Vào đầu danh sách
- ◉ Sau một phần tử
- ◉ Vào cuối danh sách

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Thêm phần tử

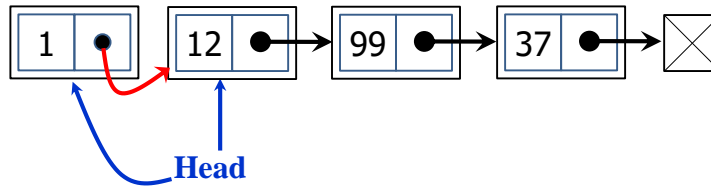
19

◉ Vào đầu danh sách:

▣ Nếu danh sách rỗng

- Phần tử vừa thêm là phần tử đầu danh sách

▣ Ngược lại,



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Thêm phần tử

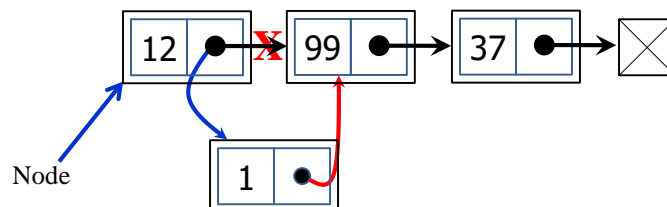
20

◉ Sau một phần tử (Node):

▣ Nếu danh sách rỗng?

▣ Nếu danh sách khác rỗng?

- Tạo node mới có dữ liệu là **Data**.
- Cập nhật lại liên kết của **Node** và node vừa tạo.



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Duyệt danh sách

21

- ◉ Đảm bảo việc truy xuất đến tất cả các phần tử trên danh sách
- ◉ Thuật toán:
 - ▣ Bắt đầu từ phần tử đầu tiên
 - ▣ Trong khi chưa hết danh sách
 - Xử lý phần tử hiện hành
 - Di chuyển đến phần tử kế tiếp

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Xoá phần tử

22

- ◉ Đầu danh sách
- ◉ Cuối danh sách
- ◉ Sau một phần tử
- ◉ Theo khóa k

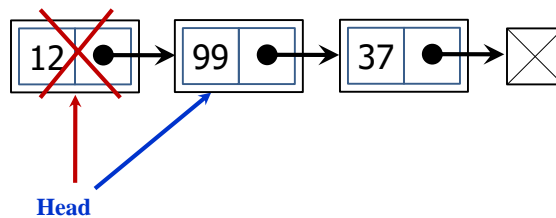
Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Xoá phần tử

23

◉ Đầu danh sách

- ▣ Nếu danh sách rỗng:
- ▣ Nếu danh sách khác rỗng:
 - Cập nhật lại Head
 - Xóa con trỏ Head cũ



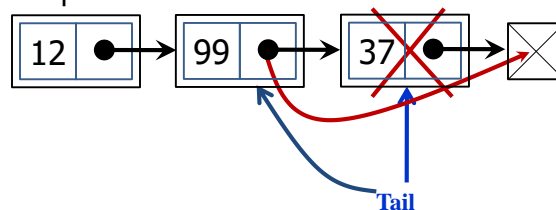
Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Xoá phần tử

24

◉ Cuối danh sách:

- ▣ Danh sách rỗng?
- ▣ Danh sách khác rỗng:
 - tìm con trỏ cuối danh sách pTail
 - Cập nhật lại pTail
 - Xóa pTail cũ



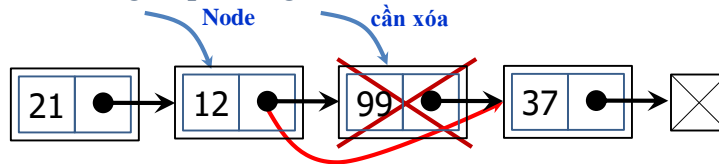
Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Xoá phần tử

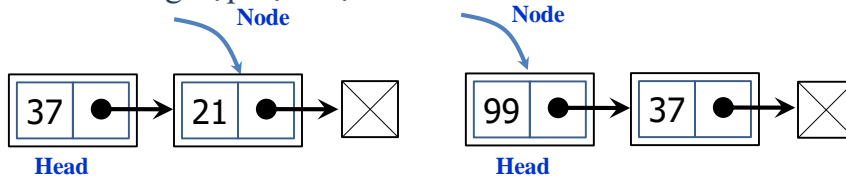
25

- ◉ Sau một phần tử (pNode)

- Trường hợp chung:



- Trường hợp đặc biệt:



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Xoá danh sách

26

- ◉ Danh sách liên kết bao gồm các phần tử được cập phát động.
- ◉ Phải xoá các phần tử trên danh sách sau khi đã sử dụng xong danh sách.
- ◉ Thuật toán:
 - ▣ Duyệt qua các phần tử trên danh sách
 - ▣ Xoá từng phần tử

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Danh sách liên kết là gì?

27

- Một dãy tuần tự các phần tử (node)
- Giữa hai phần tử có liên kết với nhau.
- Các phần tử không cần phải lưu trữ liên tiếp nhau trong bộ nhớ
- Có thể mở rộng tùy ý (chỉ giới hạn bởi dung lượng bộ nhớ)
- Thao tác Chèn/Xóa không cần phải dịch chuyển phần tử
- Có thể truy xuất đến các phần tử khác thông qua các liên kết

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

So sánh danh sách liên kết và mảng

28

Danh sách liên kết

- Số phần tử không cần xác định trước.
- Cấp phát vùng nhớ riêng lẻ cho từng phần tử.
- Truy xuất tuần tự, danh sách liên kết đơn chỉ có thể duyệt 1 chiều.
- Cần nhiều bộ nhớ hơn để lưu trữ các liên kết
- Thêm/xóa phần tử cuối: $O(1)$
- Thêm/xóa phần tử giữa: $O(1)$

Mảng

- Cần xác định trước số phần tử.
- Cần cấp phát vùng nhớ liên tục đủ lớn để lưu trữ mảng → lãng phí nếu không dùng hết.
- Truy xuất ngẫu nhiên, đơn giản, nhanh chóng.
- Không cần
- Thêm/xóa phần tử cuối: $O(1)$
- Thêm/xóa phần tử giữa: $O(n)$

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Ứng dụng

29

- Là cấu trúc dữ liệu chính cho ngôn ngữ lập trình LISP (List Processing Language) – ngôn ngữ lập trình hàm.
- Giúp nâng cao hiệu quả của một số thuật toán sắp xếp: Quick Sort, Radix Sort

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

30

Ngăn xếp - stack

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Nội dung

31

- Giới thiệu
- Các thao tác cơ bản
- Ký pháp Ba Lan ngược

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Giới thiệu

32

- Một số hình ảnh thông dụng:

- ▣ Một chồng sách vở ở trên bàn



- ▣ Một chồng đĩa



- ▣ Cơ cấu của một hộp chứa đạn súng trường.



Nhận xét gì từ các ví dụ trên?

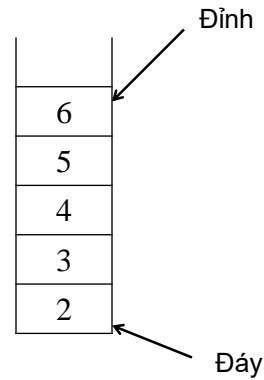
Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Giới thiệu

33

◉ Định nghĩa:

- ▣ Ngăn xếp là vật chứa các đối tượng làm việc theo cơ chế “vào sau ra trước” (**Last In First Out**)
- ▣ Đối tượng có thể được thêm vào bất kì lúc nào, nhưng chỉ có *đối tượng vào sau cùng mới được phép lấy ra khỏi ngăn xếp*.



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Các thao tác trên ngăn xếp

34

◉ Các thao tác cơ bản:

- ▣ Push: Thêm 1 phần tử vào ngăn xếp
- ▣ Pop: Lấy 1 phần tử ra khỏi ngăn xếp

◉ Các thao tác khác:

- ▣ Lưu trữ ngăn xếp
- ▣ Kiểm tra ngăn xếp rỗng
- ▣ Lấy thông tin của phần tử đầu ngăn xếp

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lưu trữ ngăn xếp

35

◉ Lưu trữ bằng mảng

- ▣ Khai báo mảng 1 chiều với kích thước tối đa N.
- ▣ t là địa chỉ của phần tử đỉnh của ngăn xếp → t sẽ thay đổi khi ngăn xếp hoạt động.
 - *Ngăn xếp rỗng thì giá trị của t là 0*

- ▣ Tạo ngăn xếp S và quản lý ngăn xếp bằng biến t:

Data S[N];

int t;

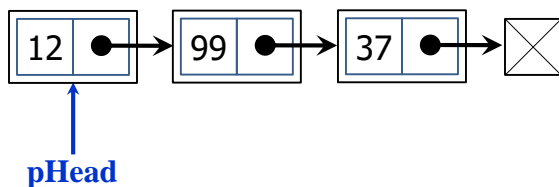
Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lưu trữ ngăn xếp

36

◉ Lưu trữ bằng DSLK:

- ▣ Dùng con trỏ pHead lưu địa chỉ của đỉnh ngăn xếp
- ▣ Ngăn xếp rỗng khi pHead = NULL



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Kiểm tra ngăn xếp rỗng

37

- Input:
- Output:
 - ▣ TRUE nếu ngăn xếp rỗng
 - ▣ FALSE nếu ngăn xếp không rỗng
- Ngăn xếp rỗng:
 - ▣ Mảng: số lượng phần tử mảng là 0
 - ▣ DSLK: pHead = NULL

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Kiểm tra ngăn xếp đầy

38

- Input:
- Output:
 - ▣ TRUE nếu ngăn xếp đầy
 - ▣ FALSE nếu ngăn xếp còn chỗ trống
- Ngăn xếp đầy:
 - ▣ Mảng: đã lưu hết các phần tử mảng
 - ▣ DSLK: không cấp phát được vùng nhớ mới cho ngăn xếp

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Thêm phần tử vào ngăn xếp (push)

39

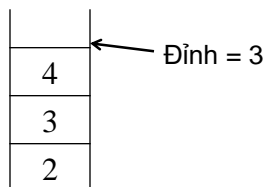
- Input: phần tử cần thêm
- Output:
- Giải thuật:
 - ▣ Kiểm tra ngăn xếp có đầy không?
 - ▣ Nếu không
 - Bổ sung phần tử mới vào
 - Cập nhật địa chỉ của con trỏ đến đỉnh ngăn xếp

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

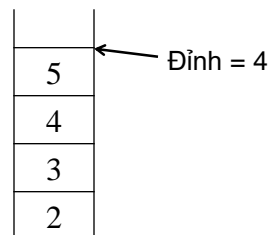
Thêm phần tử vào ngăn xếp (push)

40

- Ví dụ:



Ngăn xếp ban đầu



Ngăn xếp sau khi thêm push(5)

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lấy phần tử ra khỏi ngăn xếp (pop)

41

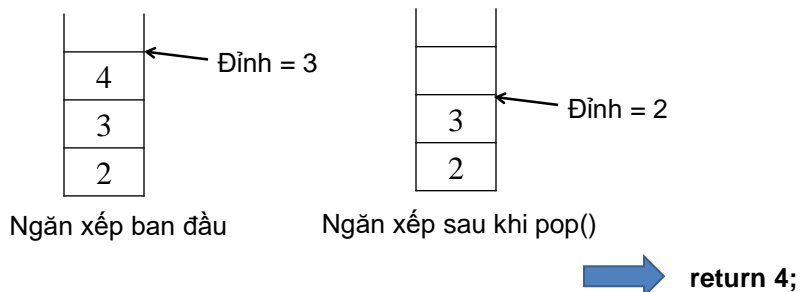
- Input:
- Output: giá trị của đối tượng đầu ngăn xếp
- Thuật toán:
 - ▣ Kiểm tra ngăn xếp rỗng hay không?
 - ▣ Nếu không:
 - Cập nhật địa chỉ của con trỏ đến đỉnh ngăn xếp
 - Xóa phần tử ở đỉnh khỏi ngăn xếp
 - Trả về giá trị của phần tử ở đỉnh

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lấy phần tử ra khỏi ngăn xếp (pop)

42

- Ví dụ:



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lấy thông tin đỉnh ngăn xếp

43

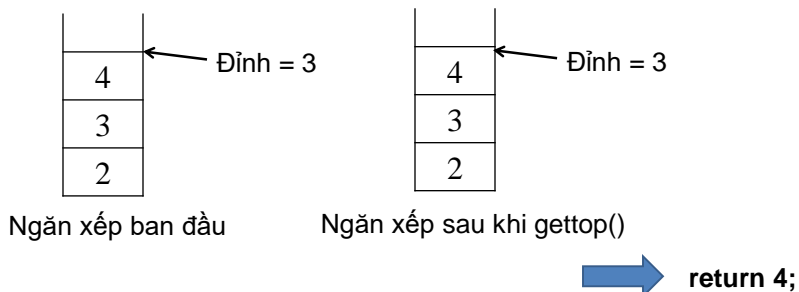
- Chỉ lấy giá trị của phần tử đầu mà *không hủy nó* khỏi ngăn xếp.
- Input:
- Output: giá trị tại đỉnh ngăn xếp
- Giải thuật:
 - ▣ Kiểm tra xem ngăn xếp có rỗng không?
 - ▣ Trả về giá trị của phần tử ở đỉnh ngăn xếp

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lấy thông tin đỉnh ngăn xếp

44

- Ví dụ



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Bài tập

45

- Cho biết nội dung của stack sau khi thực hiện các thao tác trong dãy (từ trái sang phải):

EAS*Y**QUE***ST***I*ON

Mỗi chữ cái hoặc dấu * tương ứng một thao tác trên stack trong đó:

- ▣ Một chữ cái tượng trưng cho thao tác thêm chữ cái đó vào stack
- ▣ Dấu * tượng trưng cho thao tác lấy nội dung một phần tử trong stack ra rồi in lên màn hình.

Cho biết kết quả xuất ra màn hình sau khi hoàn tất chuỗi trên?

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Bài tập

46

- Cho biết nội dung của stack và giá trị của các biến sau khi thực hiện liên tiếp các thao tác sau:
- (Ban đầu các biến được khởi tạo: $A = 5$, $B = 3$, $C = 7$)
 - Tạo stack
 - push A
 - push $C * C$
 - pop rồi lưu trữ vào biến B
 - push $B + A$
 - pop rồi lưu trữ vào biến A
 - pop rồi lưu trữ vào biến B

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Ký pháp Ba Lan

47

- ◉ Biểu thức dạng *trung tố*: dấu của các phép toán hai ngôi luôn được *đặt giữa* 2 toán hạng

▣ Ví dụ: $A + B * C$ $A + B * C - D$
 $(A+B) * C$ $(A + B) * (C - D)$

⇒ Qui định thứ tự ưu tiên của các phép toán

⇒ Dùng dấu ngoặc để phân biệt thứ tự thực hiện.

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Ký pháp Ba Lan

48

- ◉ Biểu thức dạng *tiền tố*:

Trung tố	Tiền tố
$A + B$	$+ A B$
$(A+B)*C$	$* + A B C$
$(A + B) * (C - D)$	$* + A B - C D$

- ◉ Biểu thức dạng *hậu tố*:

Trung tố	Hậu tố
$A + B$	$A B +$
$(A+B)*C$	$A B + C *$
$(A + B) * (C - D)$	$A B + C D - *$

Không cần
thiết phải
dùng dấu
ngoặc

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

◉ $10\ 7\ 3\ * + 2\ 4\ 5 + * - 18 - 7 /$

Cấu trúc dữ liệu và giải thuật – HCMUS 2016



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Chuyển trung tố sang hậu tố

51

- ◉ Mã giả: P là biểu thức trung tố ban đầu, Q là biểu thức kết quả dạng hậu tố

```
0.1 push('(');
0.2 Thêm ')' vào P
while (chưa hết biểu thức P)
{
    1. đọc 1 kí tự x trong P (từ trái qua phải)
    2. if (x là toán hạng)
        Thêm x vào Q
    3. if (x là dấu ngoặc mở)
        push(x);
```

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Chuyển trung tố sang hậu tố

52

- ◉ Mã giả:

```
4. if (x là toán tử)
    4.1 while( thứ tự ưu tiên tại đỉnh >= x)
        4.1.1 w = pop();
        4.1.2 Thêm w vào Q
    4.2 push(x);
5. if (x là dấu ngoặc đóng)
    5.1 while( chưa gặp ngoặc mở)
        4.1.1 w = pop();
        4.1.2 Thêm w vào Q
    5.2 pop(); //đẩy ngoặc mở ra khỏi ngăn xếp
}
```

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Chuyển trung tố sang hậu tố

53

- Ví dụ 1: $P = (A + B) * (C - (D + A))$

Kí tự đọc được

$$(A + B) * (C - (D + A))$$

Trạng
thái của
ngăn xếp

Diagram illustrating a neural network architecture with 16 layers. The layers are represented by vertical rectangles. The first 14 layers are grouped into pairs, with the second layer of each pair containing a '+' sign. The 15th layer contains a '-' sign. The 16th layer is empty. The layers are labeled with mathematical symbols: (, *, +, -, and empty.

$$Q = A \cdot B + C \cdot D \cdot A + - *$$

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Chuyển trung tố sang hậu tố

54

- ◉ Ví dụ 2: đổi biểu thức trung tố

$$P = A + (B * C - (D / E \wedge F) * G) * H$$

sang biểu thức dạng hậu tố

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Ứng dụng khác của ngăn xếp

55

- Dùng biến đổi cơ số
- Lượng giá biểu thức hậu tố
- Trong trình biên dịch, ngăn xếp được sử dụng để lưu môi trường các thủ tục.
- Dùng trong một số bài toán của lý thuyết đồ thị.
- Khử đệ qui đuôi.

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

56

Hàng đợi - Queue

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Nội dung

57

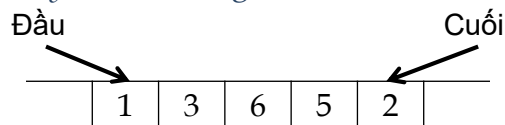
- Giới thiệu
- Các thao tác cơ bản
- Ứng dụng

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Giới thiệu

58

- Giới thiệu:
 - ▣ Là vật chứa các đối tượng làm việc theo qui tắc *vào trước ra trước* (FIFO).
 - ▣ Các đối tượng có thể được thêm vào hàng đợi bất kì lúc nào nhưng chỉ có đối tượng *thêm vào đầu tiên* mới được *lấy ra khỏi hàng đợi*



- ▣ Việc thêm vào diễn ra ở cuối, việc lấy ra diễn ra ở đầu

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Các thao tác trên hàng đợi

59

◉ Thao tác cơ bản:

- ▣ Enqueue: Thêm 1 đối tượng vào cuối hàng đợi
- ▣ Dequeue: Lấy đối tượng ở đầu ra khỏi hàng đợi

◉ Thao tác khác:

- ▣ Lưu trữ hàng đợi
- ▣ Kiểm tra hàng đợi rỗng
- ▣ Kiểm tra hàng đợi đầy
- ▣ Lấy thông tin của đối tượng ở đầu hàng đợi

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lưu trữ hàng đợi

60

◉ Lưu trữ bằng mảng:

- ▣ Khai báo mảng 1 chiều với kích thước tối đa N.
- ▣ f là địa chỉ của phần tử nằm ở đầu, r là địa chỉ của phần tử nằm ở cuối

	0	1	2	3	4	5	6	7	8
Hàng đợi			1	3	6	5	2		
			$f = 2$				$r = 6$		

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lưu trữ hàng đợi

61

◉ Lưu trữ bằng mảng:

- ▣ Các phần tử của hàng đợi sẽ di chuyển khắp các ô nhớ
→ coi không gian dành cho hàng đợi theo dạng xoay vòng.
- ▣ Hàng đợi khi xoay vòng:

0	1	2	3	4	5	6	7	8
		1	3	6	5	2		
$r = 2$				$f = 6$				

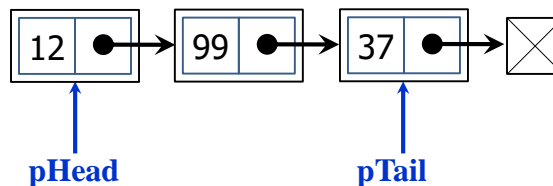
Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lưu trữ hàng đợi

62

◉ Lưu trữ hàng đợi bằng danh sách liên kết đơn

- ▣ Phần tử đầu DSLK sẽ là phần tử đầu hàng đợi.
- ▣ Phần tử cuối DSLK sẽ là phần tử cuối hàng đợi



Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Kiểm tra hàng đợi rỗng

63

- Input:
- Output:
 - ▣ TRUE nếu hàng đợi rỗng
 - ▣ FALSE nếu hàng đợi không rỗng
- Hàng đợi rỗng:
 - ▣ Mảng: ô nhớ đầu tiên không chứa dữ liệu
 - ▣ DSLK: pHead = NULL

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Kiểm tra hàng đợi đầy

64

- Input:
- Output:
 - ▣ TRUE nếu hàng đợi đầy
 - ▣ FALSE nếu hàng đợi không đầy
- Hàng đợi đầy:
 - ▣ Mảng: ô nhớ cuối hàng đợi đã chứa dữ liệu
 - ▣ DSLK: không cấp phát được vùng nhớ cho phần tử mới

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Thêm phần tử vào cuối hàng đợi

65

- Input: giá trị cần thêm
- Output:
- Giải thuật thêm phần tử (EnQueue)
 - ▣ Kiểm tra hàng đợi đã đầy chưa?
 - ▣ Trong trường hợp lưu trữ bằng mảng: kiểm tra điều kiện xoay vòng.
 - ▣ Thêm phần tử vào cuối hàng đợi
 - ▣ Cập nhật địa chỉ phần tử cuối hàng đợi

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Thêm phần tử vào cuối hàng đợi

66

- Ví dụ:

	0	1	2	3	4	5	6	7	8
Hàng đợi ban đầu			1	3	6	5	2		
			f = 2				r = 6		
EnQueue(9)			1	3	6	5	2	9	
			f = 2				r = 7		

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lấy phần tử đầu ra khỏi hàng đợi

67

- Input:
- Output: giá trị của phần tử đầu hàng đợi
- Giải thuật lấy phần tử ở đầu (DeQueue)
 - ▣ Kiểm tra hàng đợi có rỗng không?
 - ▣ Xóa phần tử đầu ra khỏi hàng đợi
 - ▣ Cập nhật địa chỉ phần tử đầu hàng đợi
 - ▣ Trong trường hợp lưu trữ bằng mảng: kiểm tra điều kiện xoay vòng

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lấy phần tử đầu ra khỏi hàng đợi

68

- Ví dụ:

	0	1	2	3	4	5	6	7	8
Hàng đợi ban đầu			1	3	6	5	2		
			f = 2				r = 6		
DeQueue() = 1				3	6	5	2		
				f = 3			r = 6		

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lấy thông tin đầu hàng đợi

69

- Chỉ lấy thông tin của đối tượng đầu hàng đợi mà *không hủy* đối tượng khỏi hàng đợi.
- Input: hàng đợi
- Output: giá trị của đối tượng đầu hàng đợi
- Giải thuật:
 - Kiểm tra hàng đợi rỗng?
 - Trả về giá trị của phần tử đầu hàng đợi

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Lấy thông tin đầu hàng đợi

70

- Ví dụ:

	0	1	2	3	4	5	6	7	8
Hàng đợi ban đầu			1	3	6	5	2		
			f = 2				r = 6		
GetFront() = 1			1	3	6	5	2		
			f = 2				r = 6		

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Ứng dụng của hàng đợi

71

- Bộ đệm bàn phím của máy tính
- Xử lý các lệnh trong máy tính: hàng đợi thông điệp trong Windows, hàng đợi tiến trình ...
- Thường dùng trong các hệ mô phỏng.

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Bài tập 1

72

- Cho một DSLK đơn, mỗi node trong DSLK lưu thông tin là 1 số nguyên và con trỏ đến node kế. Tạo 2 DSLK đơn mới (không phá hủy DSLK đã cho).
 - ▣ Một danh sách chứa các số lẻ của danh sách đã cho.
 - ▣ Một danh sách chứa các số chẵn của danh sách đã cho.

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Bài tập 1

73

▣ In ra các *đường chạy* tự nhiên từ DSLK đã cho:

VÍ DỤ: DSLK ban đầu biểu diễn các số: 1 5 6 4 8 3
7

In ra các dãy số:

1	5	6
4	8	
3	7	

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Bài tập 2

74

⦿ Cho danh sách liên kết đơn L, lập giải thuật thực hiện các phép sau đây:

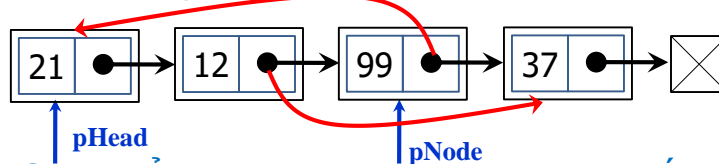
- ▣ Tính số lượng các nút của danh sách.
- ▣ Tìm tới nút thứ k trong danh sách, nếu có nút thứ k thì cho biết địa chỉ của nút đó, ngược lại trả về null.
- ▣ Bổ sung một nút vào sau nút k.
- ▣ Loại bỏ nút đứng trước nút k.
- ▣ Đảo ngược danh sách đã cho.

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Bài tập 3

75

- Hàm **MoveToFront** có tác dụng di chuyển 1 node trong xâu lên đầu xâu, như hình sau:



- Chọn kiểu khai báo hàm phù hợp và viết code
`void MoveToFront(NODE □□ pHead, NODE □□ pTail,
 NODE □□ pNode)`

Lưu ý: các kí hiệu □ có thể là *, & hoặc khoảng trắng

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Bài tập

76

Cho hàng đợi ban đầu như sau: (hàng đợi có tối đa 6 phần tử)

0	1	2	3	4	5
	A	B	C		

f = 1

r = 3

Vẽ tình trạng của hàng đợi, cho biết giá trị f, r tương ứng với mỗi lần thực hiện thao tác sau:

- Bổ sung E vào hàng đợi
- Loại 2 phần tử khỏi hàng đợi
- Bổ sung I, J, K vào hàng đợi
- Loại 2 phần tử khỏi hàng đợi
- Bổ sung O vào hàng đợi
- Loại 2 phần tử khỏi hàng đợi

Cấu trúc dữ liệu và giải thuật – HCMUS 2016

Hỏi và Đáp

Cấu trúc dữ liệu và giải thuật – HCMUS 2016