

CẤU TRÚC DỮ LIỆU NGĂN XẾP VS HÀNG ĐỢI

Bùi Tiến Lên

01/01/2017



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

NGĂN XẾP

Ngăn xếp

Định nghĩa 1

Ngăn xếp (**stack**) là một cấu trúc dữ liệu dùng để lưu trữ một tập hợp các phần tử

- ▶ Hoạt động theo cơ chế “vào sau - ra trước” (*last in, first out - LIFO*); nghĩa là, ta chỉ thấy và truy cập của đỉnh của ngăn xếp
- ▶ Cấu trúc dữ liệu này được đề xuất bởi hai nhà khoa học người Đức [Bauer and Samelson, 2001]

Ngăn xếp (cont.)

Một lớp cấu trúc dữ liệu ngăn xếp sẽ bao gồm những thao tác cơ bản sau

- ▶ Xóa ngăn xếp
- ▶ Kiểm tra ngăn xếp rỗng
- ▶ Thêm một phần tử vào ngăn xếp
- ▶ Lấy một phần tử ra khỏi ngăn xếp
- ▶ Lấy thông tin phần tử ở đỉnh ngăn xếp

Minh họa hoạt động của ngăn xếp

- ▶ Cho một ngăn xếp s rỗng
- ▶ Thêm một phần tử 3 vào ngăn xếp



- ▶ Thêm một phần tử 2 vào ngăn xếp



- ▶ Thêm phần tử 4 vào ngăn xếp



- ▶ Lấy một phần tử ra khỏi ngăn xếp



Minh họa hoạt động của ngăn xếp

- ▶ Cho một ngăn xếp s rỗng
- ▶ Thêm một phần tử 3 vào ngăn xếp



- ▶ Thêm một phần tử 2 vào ngăn xếp



- ▶ Thêm phần tử 4 vào ngăn xếp



- ▶ Lấy một phần tử ra khỏi ngăn xếp

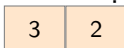


Minh họa hoạt động của ngăn xếp

- ▶ Cho một ngăn xếp s rỗng
- ▶ Thêm một phần tử 3 vào ngăn xếp



- ▶ Thêm một phần tử 2 vào ngăn xếp



- ▶ Thêm phần tử 4 vào ngăn xếp



- ▶ Lấy một phần tử ra khỏi ngăn xếp



Minh họa hoạt động của ngăn xếp

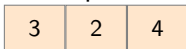
- ▶ Cho một ngăn xếp s rỗng
- ▶ Thêm một phần tử 3 vào ngăn xếp



- ▶ Thêm một phần tử 2 vào ngăn xếp



- ▶ Thêm phần tử 4 vào ngăn xếp



- ▶ Lấy một phần tử ra khỏi ngăn xếp

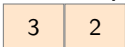


Minh họa hoạt động của ngăn xếp

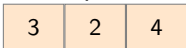
- ▶ Cho một ngăn xếp s rỗng
- ▶ Thêm một phần tử 3 vào ngăn xếp



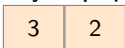
- ▶ Thêm một phần tử 2 vào ngăn xếp



- ▶ Thêm phần tử 4 vào ngăn xếp



- ▶ Lấy một phần tử ra khỏi ngăn xếp



Cài đặt ngăn xếp

Kiểu dữ liệu stack có thể cài đặt bằng

- ▶ Mảng một chiều
- ▶ Danh sách liên kết

Cài đặt ngăn xếp (cont.)

Cài đặt lớp cho cấu trúc dữ liệu trừu tượng ngăn xếp Stack

```
1  template <class T>
2  class Stack
3  {
4      private:
5          // data
6
7      public:
8          void clear();
9          bool isEmpty();
10         void push(T data);
11         T pop();
12         T top();
13 };
```

Ứng dụng của ngăn xếp

Kiểu dữ liệu ngăn xếp được dùng trong nhiều thuật toán

- ▶ Thuật toán Balan ngược *Reverse Polish Notation* để tính giá trị một biểu thức toán học
- ▶ Thuật toán tìm đường đi - quay lui, như: mê cung, mã đi tuần, 8 hoàng hậu
- ▶ Thuật toán về xử lý đồ họa
- ▶ Thuật toán quản lý việc gọi hàm

Các ví dụ

Ví dụ 1

Cho một chuỗi ký tự

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

- ▶ Một **ký tự alphabet** tượng trưng cho thao tác thêm chữ cái đó vào **stack**
- ▶ Ký tự ***** tượng trưng cho thao tác lấy nội dung một phần tử trong **stack** ra rồi in lên màn hình.
- ▶ Cho biết nội dung của **stack** sau mỗi thao tác
- ▶ Cho biết kết quả xuất ra màn hình sau khi hoàn tất chuỗi trên?

Các ví dụ (cont.)

Ví dụ 2

Các biến có giá trị ban đầu $A = 5$, $B = 3$, $C = 7$. Hãy thực hiện các thao tác sau và cho biết nội dung của stack và giá trị của các biến

1. Xóa **stack**
2. push A
3. push $C * C$
4. pop rồi lưu trữ vào biến B
5. push $B + A$
6. pop rồi lưu trữ vào biến A
7. pop rồi lưu trữ vào biến B

Biểu thức toán học

Có 3 cách viết biểu thức toán học

- ▶ Trung tố

`<toán hạng1><toán tử><toán hạng 2>`

- ▶ Tiền tố

`<toán tử><toán hạng1><toán hạng 2>`

- ▶ Hậu tố

`<toán hạng1><toán hạng 2><toán tử>`

Biểu thức toán học (cont.)

Ví dụ 3

Một số biểu thức toán học

trung tố	tiền tố	hậu tố
$A+B*C$	$+*BCA$	$BC*A+$
$(A-B)/C$	$/-ABC$	$AB-C/$
$(A+B)*(C-D)$	$*+AB-CD$	$AB+CD-*$

Nhận xét

- ▶ Biểu thức trung tố *quen thuộc*. Tuy nhiên, khó cài đặt tính
- ▶ Biểu thức tiền tố *hơi lạ*. Tuy nhiên, đây là cách viết khá quen thuộc trong lập trình
- ▶ Biểu thức hậu tố *lạ*. Tuy nhiên, dễ cài đặt tính toán

Thuật toán Ba Lan ngược

- ▶ Đây là thuật toán chuyển một biểu thức ở dạng trung tố P sang dạng hậu tố Q
- ▶ Giả sử biểu thức được viết ở dạng đơn giản nhất. Toán hạng và toán tử được biểu diễn bằng một ký tự
- ▶ Thuật toán cần một cấu trúc dữ liệu stack

Thuật toán Ba Lan ngược (cont.)

```
stack.push('(')
P.append(')')
while (P.end())
    c ← P.read() (từ trái qua phải)
    if (c là toán hạng) Q.append(c)
    if (c là dấu ngoặc mở) stack.push(c)
    if (c là toán tử)
        while (độ ưu tiên của stack.top() cao hơn c)
            Q.append(stack.pop())
        stack.push(c)
    if (c là dấu ngoặc đóng)
        while (stack.top() không phải ngoặc mở)
            Q.append(stack.pop())
        stack.pop()
```

Áp dụng thuật toán

Ví dụ 4

Chuyển biểu thức trung tố $P = (A+B) * (C - (D+A))$ sang biểu thức hậu tố Q bằng thuật toán Ba Lan ngược

HÀNG ĐỢI

Định nghĩa 2

Hàng đợi là một cấu trúc dữ liệu để

- ▶ Dùng để lưu trữ một tập các dữ liệu
- ▶ Hoạt động theo cơ chế “vào trước - ra trước” (*first in, first out - FIFO*); cũng như cấu trúc dữ liệu ngăn xếp chúng ta chỉ có thể truy xuất đến phần tử đầu tiên của ngăn xếp

Cài đặt hàng đợi

Kiểu dữ liệu queue có thể cài đặt bằng

- ▶ Mảng một chiều
- ▶ Danh sách liên kết

Cài đặt hàng đợi (cont.)

Cài đặt lớp cho cấu trúc dữ liệu hàng đợi queue, về cơ bản nó giống như cấu trúc dữ liệu ngăn xếp

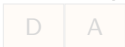
```
1  template <class T>
2  class Queue
3  {
4      private:
5          // data
6
7      public:
8          void clear();
9          bool isEmpty();
10         void enqueue(T data);
11         T dequeue();
12         T top();
13 };
```

Minh họa hoạt động của hàng đợi

- ▶ Cho một hàng đợi s rỗng
- ▶ Thêm một phần tử A vào hàng đợi



- ▶ Thêm một phần tử D vào hàng đợi



- ▶ Thêm phần tử C vào hàng đợi



- ▶ Lấy một phần tử ra khỏi hàng đợi

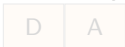


Minh họa hoạt động của hàng đợi

- ▶ Cho một hàng đợi s rỗng
- ▶ Thêm một phần tử A vào hàng đợi



- ▶ Thêm một phần tử D vào hàng đợi



- ▶ Thêm phần tử C vào hàng đợi



- ▶ Lấy một phần tử ra khỏi hàng đợi



Minh họa hoạt động của hàng đợi

- ▶ Cho một hàng đợi s rỗng
- ▶ Thêm một phần tử A vào hàng đợi



- ▶ Thêm một phần tử D vào hàng đợi



- ▶ Thêm phần tử C vào hàng đợi



- ▶ Lấy một phần tử ra khỏi hàng đợi



Minh họa hoạt động của hàng đợi

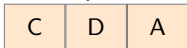
- ▶ Cho một hàng đợi s rỗng
- ▶ Thêm một phần tử A vào hàng đợi



- ▶ Thêm một phần tử D vào hàng đợi



- ▶ Thêm phần tử C vào hàng đợi



- ▶ Lấy một phần tử ra khỏi hàng đợi



Minh họa hoạt động của hàng đợi

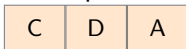
- ▶ Cho một hàng đợi s rỗng
- ▶ Thêm một phần tử A vào hàng đợi



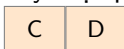
- ▶ Thêm một phần tử D vào hàng đợi



- ▶ Thêm phần tử C vào hàng đợi



- ▶ Lấy một phần tử ra khỏi hàng đợi



Ứng dụng của hàng đợi

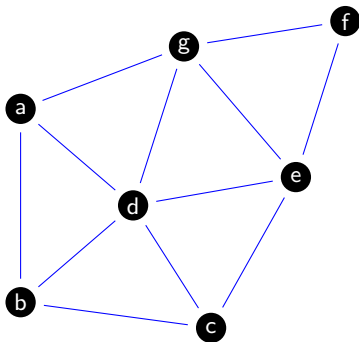
Kiểu dữ liệu hàng đợi cũng được dùng trong rất nhiều thuật toán

- ▶ Thuật toán quản lý việc xếp hàng (theo số thứ tự) trong ngân hàng, bệnh viện
- ▶ Thuật toán tìm đường đi - quay lui, như: mê cung, mã đi tuần, 8 hoàng hậu
- ▶ Thuật toán về xử lý đồ họa

Thuật toán tìm đường đi theo chiều rộng

Algorithm 1 Tìm đường đi từ đỉnh v_s đến v_e

```
1: procedure BFS_FIND_PATH( $v_s, v_e$ )
2:    $queue \leftarrow v_s$ 
3:   while  $queue \neq \emptyset$  do
4:      $v \leftarrow queue$ 
5:     Duyệt đỉnh  $v$ 
6:     if  $v == v_e$  then
7:       In ra đường và kết thúc
8:     for mỗi đỉnh  $u$  kề với đỉnh  $v$  do
9:       if đỉnh  $u$  chưa duyệt và không có trong  $queue$  then
10:         $pre[u] = v$ 
11:         $queue \leftarrow u$ 
```



Hình 1: Tìm đường đi

Đánh giá

Hãy so sánh đánh giá các kiểu cài đặt

- ▶ Mảng
- ▶ Danh sách liên kết

Cho kiểu dữ liệu stack, queue

Tài liệu tham khảo



Bauer, F. L. and Samelson, K. (2001).

Verfahren zur automatischen verarbeitung von kodierten daten und rechenmaschine zur ausübung des verfahrens.

In Pioneers and Their Contributions to Software Engineering, pages 29–40. Springer.