



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY

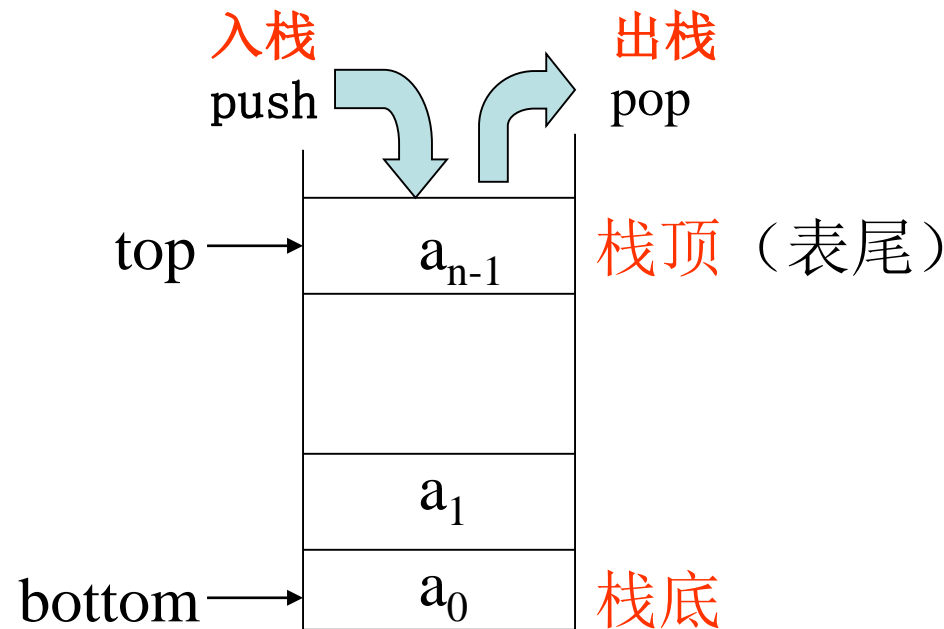
# 第3章 栈



## 3.1 栈的定义

栈是限制**仅在表的一端**进行插入和删除运算的线性表

- 栈底（bottom）：结构的首部(结点最早到达的部分)
- 栈顶（top）：结构的尾部(结点最晚到达的部分)
- 出栈（Pop）：结点从栈顶删除
- 入栈（Push）：结点在栈顶位置插入
- 空栈：栈中结点个数为零时



# 例题

设 a、b、c、d、e 5 个元素依次进栈（进栈后可立刻出栈），问能否得到下列的出栈序列。若能，写出相应的 push 和 pop 序列；若不能，说明理由

a, b, c, d, e

b, c, a, d, e

a, b, e, c, d



a, c, e, b, d





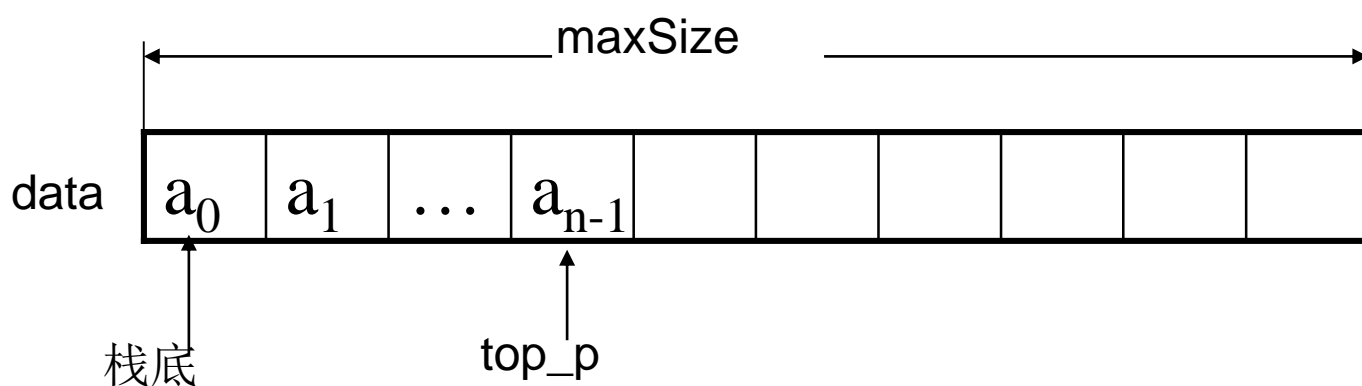
栈在计算机中主要有两种基本的存储结构：  
顺序存储结构和链式存储结构。

- 顺序存储的栈为顺序栈
- 链式存储的栈为链接栈



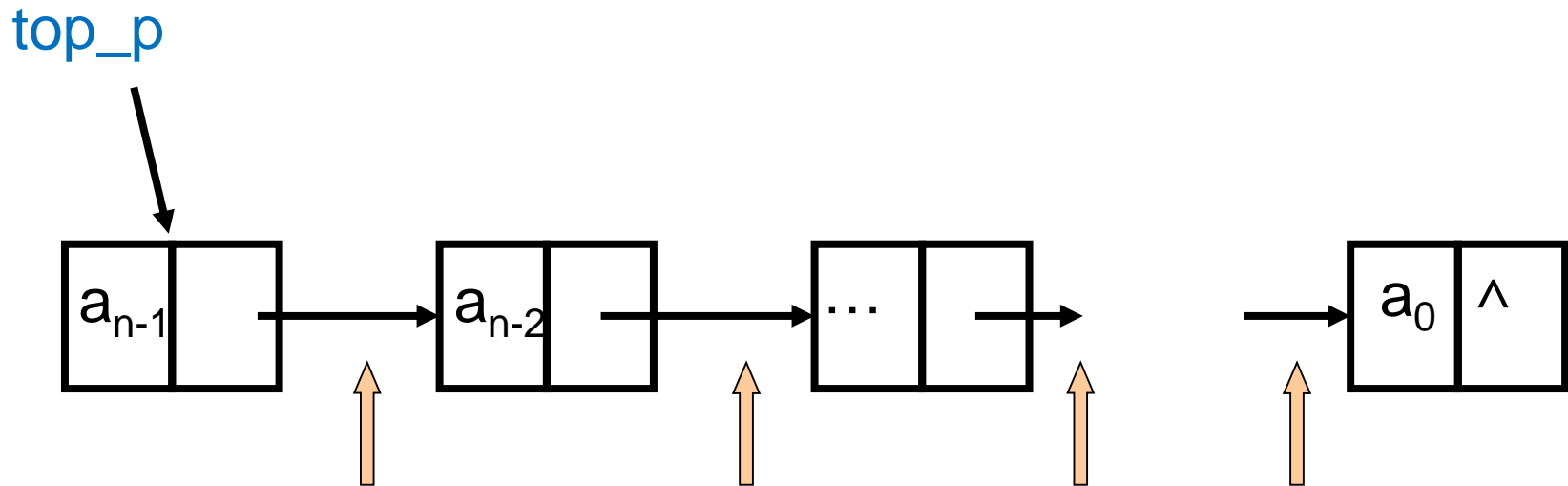
## 3.2 顺序栈

- 顺序栈用连续的空间存储栈中的结点，即数组
- 用数组的后端表示栈顶。进栈和出栈总是在栈顶一端进行，不会引起类似顺序表中的大量数据的移动。
- 栈顶位置是随着进栈和退栈操作而变化的，用一个整型量 **top** 来指示当前栈顶位置



- **链栈**是采用**链表**作为存储结构实现的栈，是线性链表的特例。
- 栈的操作都是在栈顶进行的，因此不需要双链表，用单链表就足够了，而且不需要头结点
- 对栈来讲，只需要考虑栈顶元素的插入删除。从栈的基本运算的实现方便性考虑，可将单链表的头指针指向栈顶。

# 栈的链接实现



**注意：链栈中指针的方向是从栈顶指向栈底方向**

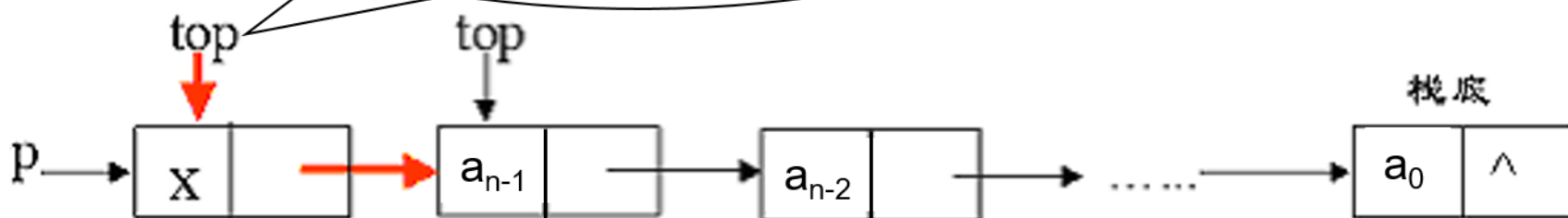
$top\_p$ 为栈顶指针，始终指向当前栈顶元素结点。  
若 $top\_p=NULL$ ，则代表空栈。

**注意：**链栈在使用完毕时，应该释放其空间。



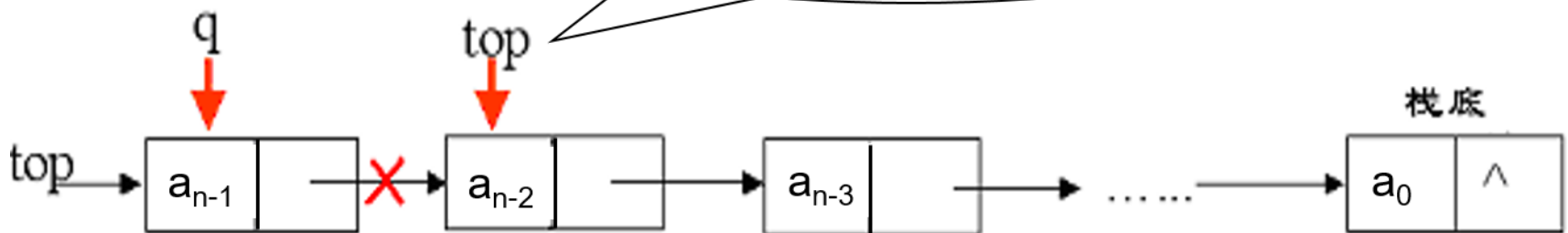
# 链栈的入栈操作和出栈操作

将 $x$ 入栈，修改栈顶  
指针： $top=p$



入栈示意图

$a_n$ 出栈，修改栈顶指  
针： $top=top \rightarrow next$



出栈示意图



# 栈的应用：表达式计算

- 中缀表达式：运算符位于两个运算数之间。如， $a + b * c - d / e$
- 后缀表达式：运算符位于两个运算数之后。如， $a b c * + d e / -$

后缀表达式求值策略：从左向右“扫描”，逐个处理运算数和运算符号

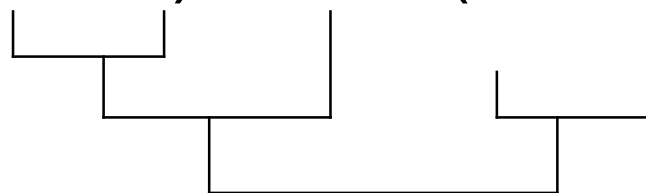
1. 遇到运算数怎么办？如何“记住”目前还未参与运算的数？
2. 遇到运算符号怎么办？对应的运算数是什么？

启示：需要有存储方法，能顺序存储运算数，并在需要时“倒序”输出！

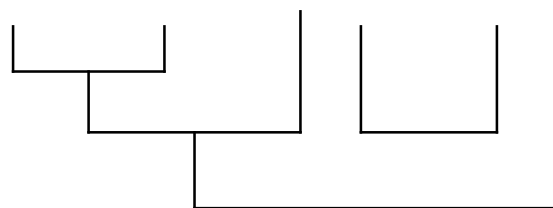
# 后缀式的优点

- 可以不考虑运算符的优先级

例如:  $(a + b) * c / (d - e)$



后缀式为:  $a b + c * d e - /$



1. 初始化一个栈。
2. 依次读入后缀式的操作数和运算符。
3. 若读到的是操作数，则将其进栈。
4. 若读到的是运算符，则将栈顶的两个操作数出栈，后弹出的操作数为被操作数，先弹出的为操作数，将得到的操作数完成运算符所规定的运算，并将结果进栈。
5. 回到2的读入操作，继续。
6. 当栈中只剩有一个操作数时，弹出该操作数，它就是表达式的值。

以 $5*(7-2*3)+8/2$ 为例，它的后缀式为5 7 2 3\*-\*8 2/+

步骤	读剩的后缀式	栈中内容	步骤	读剩的后缀式	栈中内容
1	<u>5 7 2 3</u> *-* <u>8 2</u> /+		10	/+	<u>5 8 2</u>
2	<u>7 2 3</u> *-* <u>8 2</u> /+	<u>5</u>	11	+	<u>5 4</u>
3	<u>2 3</u> *-* <u>8 2</u> /+	<u>5 7</u>	12		<u>9</u>
4	<u>3</u> *-* <u>8 2</u> /+	<u>5 7 2</u>	13		
5	*-* <u>8 2</u> /+	<u>5 7 2 3</u>	14		
6	-* <u>8 2</u> /+	<u>5 7 6</u>	15		
7	* <u>8 2</u> /+	<u>5 1</u>	16		
8	<u>8 2</u> /+	<u>5</u>	17		
9	<u>2</u> /+	<u>5 8</u>	18		



# 中缀式转换为后缀式的算法

- 若读入的是操作数，立即输出。
- 若读入的是闭括号，则将栈中的运算符依次出栈，并将其放在操作数序列之后。出栈操作一直进行到遇到相应的开括号为止。将开括号出栈。
- 若读入的是开括号，则进栈。
- 若读入的是运算符，如果栈顶运算符优先级高，则栈顶运算符出栈；出栈操作一直要进行到栈顶运算符优先级低为止，然后将新读入的运算符进栈保存。
- 在读入操作结束时，将栈中所有的剩余运算符依次出栈，并放在操作数序列之后，直至栈空为止。

序号	读剩的表达式	栈	输出
1	$(5 + 6 * (7 + 3) / 3) / 4 + 5$		
2	$5 + 6 * (7 + 3) / 3) / 4 + 5$	(	
3	$+ 6 * (7 + 3) / 3) / 4 + 5$	(	5
4	$6 * (7 + 3) / 3) / 4 + 5$	(+	5
5	$* (7 + 3) / 3) / 4 + 5$	(+	5 6
6	$(7 + 3) / 3) / 4 + 5$	(+*	5 6
7	$7 + 3) / 3) / 4 + 5$	(+*(	5 6
8	$+ 3) / 3) / 4 + 5$	(+*(	5 6 7
9	$3) / 3) / 4 + 5$	(+*(+	5 6 7
10	$) / 3) / 4 + 5$	(+*(+	5 6 7 3
11	$/ 3) / 4 + 5$	(+*	5 6 7 3 +
12	$3) / 4 + 5$	(+ /	5 6 7 3 + *
13	$) / 4 + 5$	(+ /	5 6 7 3 + * 3
14	$/ 4 + 5$		5 6 7 3 + * 3 / +
15	$4 + 5$	/	5 6 7 3 + * 3 / +
16	$+ 5$	/	5 6 7 3 + * 3 / + 4
17	5	+	5 6 7 3 + * 3 / + 4 /
18		+	5 6 7 3 + * 3 / + 4 / 5
19			5 6 7 3 + * 3 / + 4 / 5 +

1. 一个栈的输入序列是1 2 3 4 5, 则下列序列中不可能是栈的输出序列是 D

A. 2 3 4 1 5      B. 5 4 3 2 1      C. 2 3 1 4 5      D. 1 5 4 2 3

2. 下面程序段的功能实现数据 x 进栈, 要求在下划线处填上正确的语句。

```
struct sqstack{int s[100]; int top; sqstack(){top = -1;}};  
void push(sqstack &stack, int x){  
    if(stack.top==99)cout<< “overflow” ;  
    else{stack.top++; stack.s[stack.top]=x;}  
}
```

3. 栈是将插入或删除操作限定在()处进行的线性表。  
A. 栈底      B. 栈顶      C. 中间      D. 不确定

参考答案： B

4. 向一个栈顶指针为HS的链表栈中插入一个s所指结点时,则执行()  
(不带空的头结点)  
A.  $HS \rightarrow next = s;$   
B.  $s \rightarrow next = HS \rightarrow next; \quad HS \rightarrow next = s;$   
C.  $s \rightarrow next = HS; \quad HS = s;$   
D.  $s \rightarrow next = HS; \quad HS = HS \rightarrow next;$

参考答案： C



5. 向一个栈顶指针为HS的链表栈中删除栈顶元素时,用x保存被删除节点的值, 则执行( )(不带空的头结点)

- A.  $x=HS; HS= HS \rightarrow next;$
- B.  $x=HS \rightarrow data;$
- C.  $HS= HS \rightarrow next; x=HS \rightarrow data;$
- D.  $x=HS \rightarrow data; HS= HS \rightarrow next;$

参考答案: D

6. 一个栈的输入序列为123....n,若输出序列的第一个元素是n,输出第i ( $1 < i \leq n$ )个元素是( )。

- A. 不确定
- B.  $n-i+1.$
- C.  $i$
- D.  $n-i$

参考答案: B

7. 算术表达式  $a + b * (c + d/e)$  转为后缀表达式后为( )

A.  $ab + cde/*$     B.  $abcde/+*+$     C.  $abcde/*++$     D.  $abcde*/++$

参考答案: B

8. 已知操作符包括+、-、\*、/、( 和 )。将中缀表达式  $a+b-a*((c+d)/e-f)+g$  转换为等价的后缀表达式  $ab+acd+e/f-* -g+$  时, 用栈来存放暂时还不能确定运算次序的操作符, 若栈初始时空, 则转换过程中同时保存在栈中的操作符的最大个数是()。

A. 5    B. 7    C. 8    D. 11

参考答案: A

9. 元素a, b, c, d, e依次进入初始为空的栈中, 若元素进栈后可停留、可出栈, 直到所有元素都出栈, 则在所有可能的出栈序列中, 以元素d开头的序列个数是()

A. 3    B. 4    C. 5    D. 6

参考答案: B