

小马加编信息学教案(十八)

栈

- 一. 课程内容
- 二. 知识讲解
 - 1. 栈概念
 - 2. 队列基本操作实现
 - * 2.1 判断栈是否为空 (empty)
 - * 2.2 清空 (clear)
 - * 2.3 进栈 (push)
 - * 2.4 栈顶查看 (get_top)
 - * 2.5 出栈 (pop)
- 三. 经典例题
- 四. 提高巩固

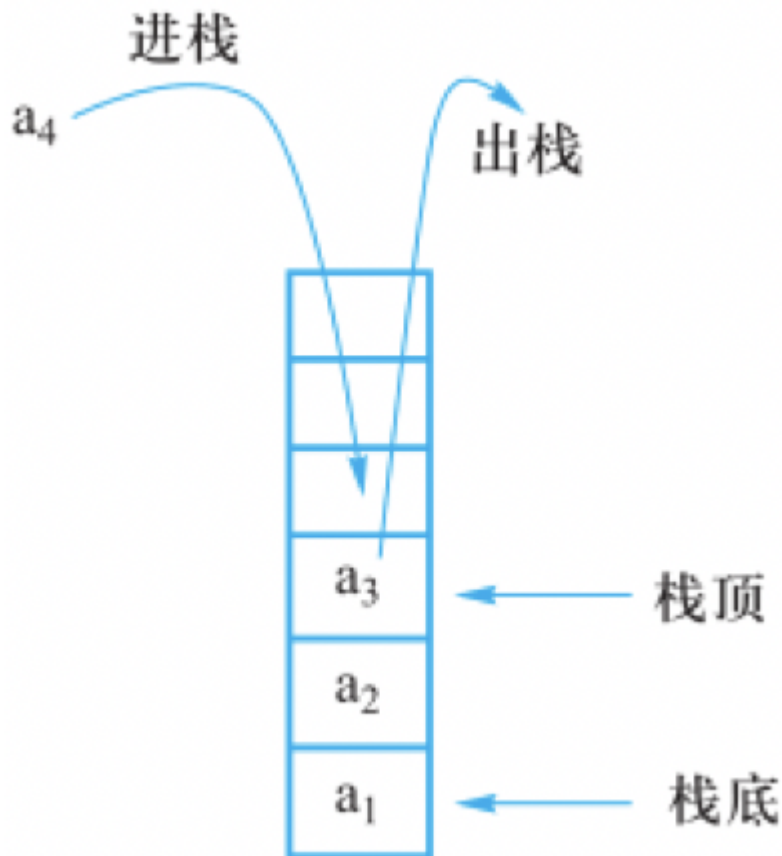
一. 课程内容

1. 栈概念
2. 栈基本操作实现

二. 知识讲解

1. 栈概念

栈是一种数据结构，可以理解为组织变量以存储数据的一种方式
可以把队列理解为一个规定了顶、底的变量序列。类似一个竖起来的数组
但规定只能对栈顶元素进行加入或删除操作，比如



我们规定只允许

查询栈顶元素（即查询 a_3 ）

删除栈顶元素（即删除 a_3 ,然后 a_2 成为栈顶）

在当前栈顶上加入元素（即在 a_3 上面加入 a_4 ,然后 a_4 成为栈顶）

这样的数据结构看起来很鸡肋是不是。。还没有数组好用

其实有些算法的实现只需要栈的功能，所以还是有必要学习的

2. 队列基本操作实现

我们采用一维数组stk来实现队列

令变量top指向栈顶位置（即栈顶为 $stk[top]$ ）

由于栈底位置不改变，令栈底为 $stk[1]$

初始时令 $top=0$

假设元素为int（其它类型同理）

2.1 判断栈是否为空（empty）

```
bool empty()
{
    if(top==0) return 1;
    return 0;
}
```

为空返回1，非空返回0

2.2 清空 (clear)

```
void clear()
{
    top=0;
}
```

栈元素为stk[1]~stk[top]，当top=0时队列为空

2.3 进栈 (push)

```
void push(int x)
{
    stk[++top]=x;
}
```

在栈顶上方加入元素x，并使元素x成为栈顶

2.4 栈顶查看 (get_top)

```
int get_top()
{
    return stk[top];
}
```

返回值为栈顶元素

在查看前先判断栈是否为空，栈为空时查看栈顶无意义

2.5 出栈 (pop)

```
void pop()
{
    top--;
}
```

在查看前先判断栈是否为空，栈为空时删除栈顶无意义，不执行

栈特点：先进后出

三. 经典例题

1. 在纸上模拟一个存储int的栈。
给出每个有返回值操作的返回值。
并画出每个操作后的栈。

```
push 1
push 7
push 5
get_top
pop
push 4
```

get_top
empty
clear
empty

2. 使用c++实现一个储存int的栈。
给出每个有返回值操作的返回值。

输入格式： 同上题

输出格式： 每行一个整数，表示对应返回值

样例输入	样例输出
empty push 5 push 7 get_top pop get_top empty	 1 7 5 0

3.溶液模拟器

实现一个溶液模拟器。溶液初始体积、浓度为v0， c0%。然后进行一系列操作。
操作有两种：

- 1、P v c 表示向当前溶液中混入体积、浓度分别为v， c的溶液
- 2、Z 撤销上一个P操作

每次操作后输出溶液当前体积、浓度

输入格式：
第一行两个整数c0,v0
第二行一个整数n， 表示n个操作
接下来n行， 每行格式为P v c或Z
输入都为整数
 $0 \leq c0, c \leq 100$
 $n \leq 10000$

输出格式
n行， 每行格式为操作后的体积、浓度
体积为整数， 浓度保留5位小数

样例输入	样例输出
------	------

样例输入	样例输出
100 100 2 P 100 0 Z	200 50.00000 100 100.00000

四. 提高巩固

1.括号匹配

规定表达式只包括小括号和中括号。

规定空表达式为合法表达式。

且若A为合法表达式，则(A)、[A]为合法表达式。

除了以用以上方式在有限步内可以生成的，都是不合法表达式

给出一个表达式，判断是否是合法的

输入格式: 一行，不超过255个字符

输出格式: 一行，“OK”或“Wrong”

样例输入	样例输出
[()]	Wrong

2.表达式求值（noip2013普及组）

给一个只包含加法和乘法的表达式，计算表达式的值对10000取余数的结果

输入格式:一行，仅包含数字，+，*。保证数字在int范围，保证表达式合法。长度不超过100000

输出格式:一行，一个数字表示表达式的值%10000

样例输入	样例输出
1+1*3+4	8