

数据结构

夏天

xiat@ruc.edu.cn

中国人民大学信息资源管理学院

栈和队列

栈和队列：两种特殊的线性表

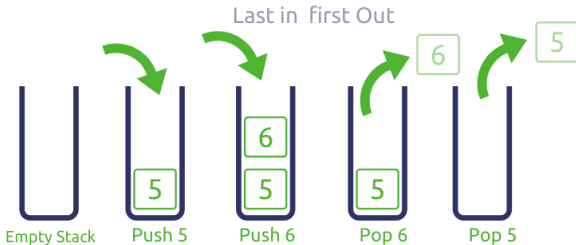
栈/Stack

- 限制仅在表的一端进行插入和删除。
- 通常称插入、删除的一端为栈顶 (Top)，另一端为栈底 (Bottom)
- LIFO : Last In First Out

队/Queue

- 限制仅在表的一端进行插入、在另一端进行删除。
- 允许插入的一端称队尾 (rear)，允许删除的一段称为队头 (front)。
- FIFO : First In First Out

Stack



- 操作系统中的栈

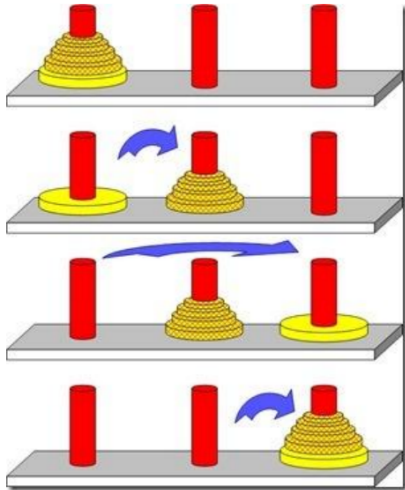
- * 由编译器自动分配释放，存放函数的参数值，局部变量的值等。栈使用的是一级缓存，被调用时处于存储空间中，调用完毕立即释放。

- 数据结构中的栈

- * 一种后进先出的数据结构

为什么设计栈、研究栈？

- 栈的一个重要应用是在程序设计中实现递归，从而使许多实际问题大大简化。

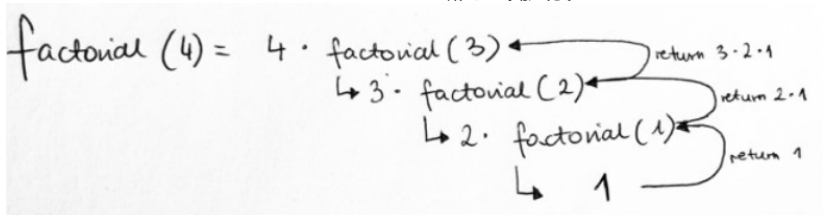


- 上帝创造世界的时候做了三根金刚石柱子，在一根柱子上从下往上按大小顺序摞着 64 片黄金圆盘。上帝命令婆罗门把圆盘从下面开始按大小顺序重新摆放在另一根柱子上，并且规定一次只能移动一个圆盘，在小圆盘上不能放大圆盘。
- 有预言说，这件事完成时宇宙会在一瞬间闪电式毁灭。也有人相信婆罗门至今还在一刻不停地搬动着圆盘。
- 18,446,744,073,709,551,615 次搬动才能挪完 64 片金盘！

举例：计算 n 的阶乘

```
int factorial (int n) {  
    int f;  
    if (n==1) f=1;  
    else f=n*fact (n-1);  
    return f;  
}
```

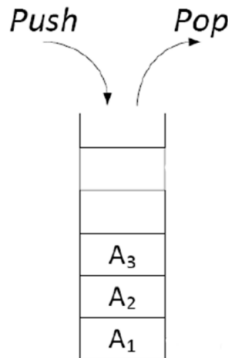
- ① 将调用函数的现场 (各寄存器的值, 中断时的程序地址等) 入栈, 转入被调函数;
- ② 执行被调函数, 如又调用其它函数, 则执行上述步骤;
- ③ 被调函数执行完, 取栈顶的值, 恢复调用函数时的现场, 根据现场中的指令地址, 恢复调用函数在中断处继续执行。



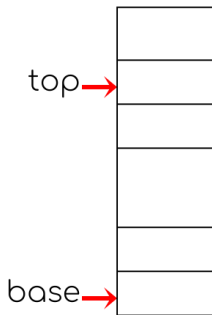
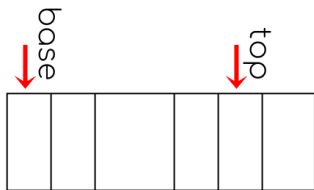
栈的存储表示方法

请考虑其常用操作。试想选用顺序存储还是链式存储？

- 特点：后进先出
 - * 1、经常性的在栈顶插入新元素，以及取栈顶元素；
 - * 2、无须访问非栈顶元素。



1. 顺序栈



- ① 顺序栈中元素用地址连续的存储单元依次存放；
- ② 栈底位置固定不变；
- ③ 栈顶位置 top 随着进栈和出栈的操作而变化。

顺序栈类型定义：

```
class stack<Elem>{  
    Object[] data;  
    int top;  
    int maxSize;  
} SeqStack;
```


- 动态分配

- * 先为栈分配一个初始容量，在栈的空间不够使用时再逐段扩大。

- 指针 base

- * 始终指向栈底位置，如果 base 为 NULL 表示栈不存在；

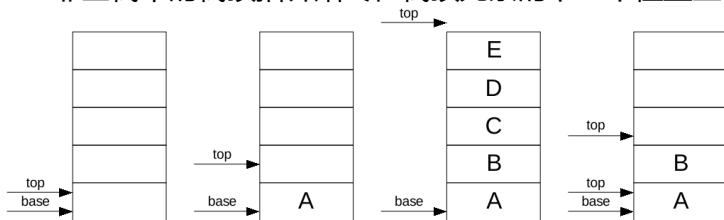
- 指针 top

- * 初值指向栈底，即 $top == base$ ，表示栈空；(不唯一)

- * 每当插入新的栈顶元素，指针 $top++$ ；

- * 每当删除栈顶元素，指针 $top--$ ；

- * 非空栈中的栈顶指针始终在栈顶元素的下一个位置上。



对于顺序栈，入栈需要判栈是否满，是则需要中止、或重新分配空间，否则出现空间溢出。

```
public Boolean push (Elem e) {  
    if ( top==maxSize) {  
        print(" 栈已满");  
        return false;  
    }  
    data[top++] = e;  
    return true;  
}
```

对于顺序栈，入栈需要判栈是否满，是则需要中止、或重新分配空间，否则出现空间溢出。

```
public Boolean push (Elem e) {  
    if ( top==maxSize) {  
        print(" 栈已满");  
        return false;  
    }  
    data[top++] = e;  
    return true;  
}
```

出栈首先要判断栈是否为空；否则栈空时进行操作将出现下溢错误。

```
public Elem pop() {  
    if(top==base) {  
        return null;  
    } else {  
        top = top-1  
        return data[top];  
    }  
}
```

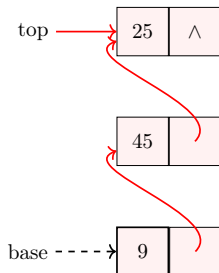
2. 链栈

- 链栈有无栈满溢出问题？
- 指针如何指？
 - * 从栈顶依次向后指，因为操作主要是在栈顶插入、删除，经常需要根据栈顶元素找次顶元素。
- 是否要加头结点？
 - * No. 因为头部插入不会出现处理不一致的问题。

写出链栈的类型定义：

```
class Node{  
    Elem data;  
    Node next;  
}
```

```
Node top; //链栈
```



请尝试写出入栈出栈的算法

```
public push (Elem e) {  
    //是否还要判断栈是否满?  
    top=new Node(e, top);  
    return OK;  
}
```

```
public pop(){  
    //是否还要判断栈是否空?  
    e=top;  
    top=top.next;  
    e.next=null;  
    return e.data;  
}
```

栈的应用

迷宫求解 (见 PPT)