

# 数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program



## 疑问解释

- 上节课我们讲解栈的结构，我们是这样声明的：

```
typedef int ElemType;  
typedef struct  
{  
    ElemType *base;  
    ElemType *top;  
    int stackSize;  
}sqStack;
```

- 有些朋友提出了疑问：怎么没有data元素存放数据？怎么会有两个ElemType元素？



## 疑问解释

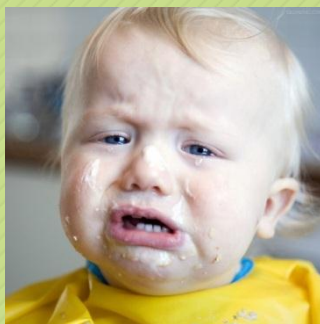
- 其实如果小甲鱼按照套路出牌，我们完全可以这样子声明：

```
typedef int ElemType;  
typedef struct  
{  
    ElemType data[MAXSIZE];  
    int top;        // 用于标注栈顶的位置  
    int stackSize;  
}
```



## 疑问解释

- 但是呢，小甲鱼要考虑到大家希望学习到的是“不变应万变”的能力，所以，这次有些朋友没反应过来^\_^
- 好吧，如果理解的朋友接下来的两分钟视频可以撸过，小甲鱼画两个图给可能还不理解的朋友解释下，因为比较重要，所以大家不要嫌弃我罗嗦哈，就两分钟T\_T



## 疑问解释

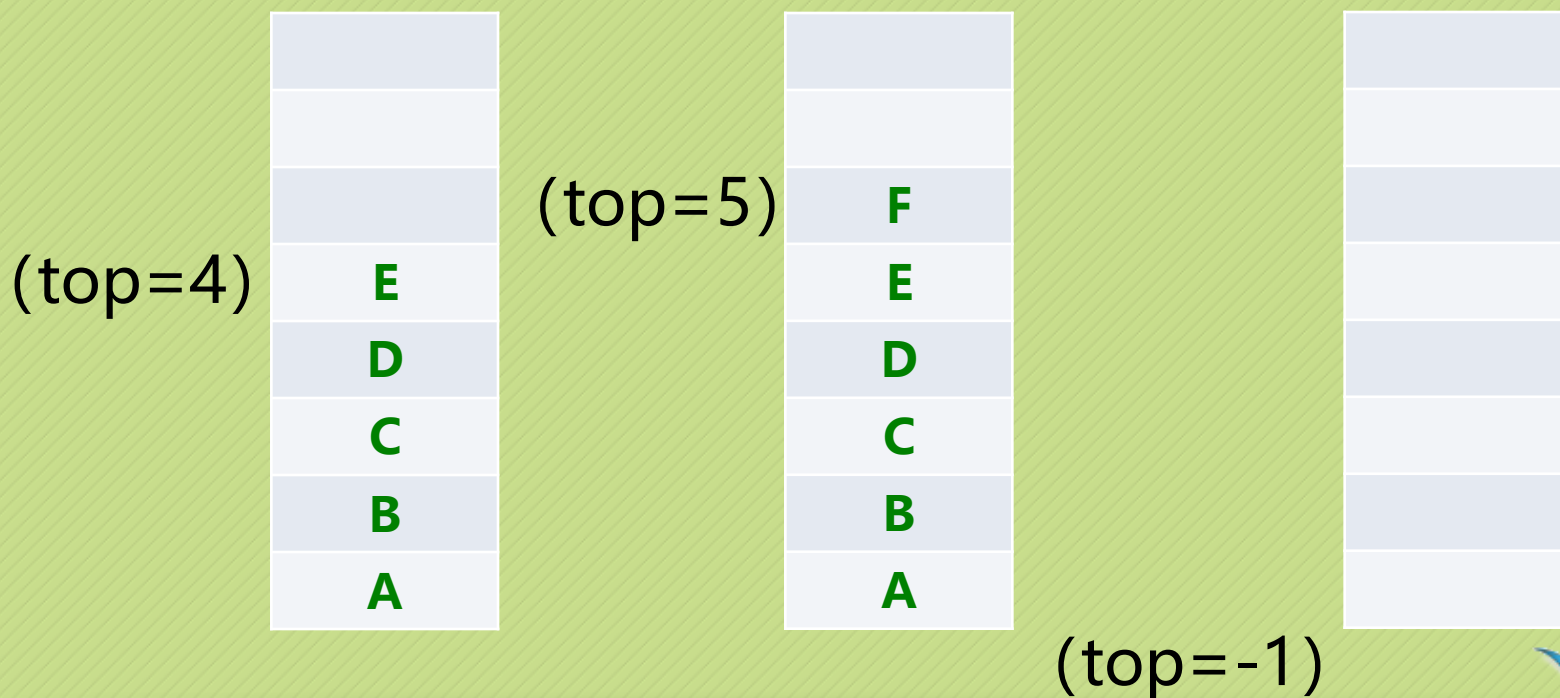
- 我们课堂的定义方法:
  - ( $\text{ElemType}^* \text{base}$  和  $\text{ElemType}^* \text{top}$ )





## 疑问解释

- 按照传统套路:
  - int top (用于索引数组)



## 栈的其他操作

- 上节课我们介绍了柯尔特半自动手枪，介绍了栈相关的基本操作：入栈和出栈。
- 除了以上介绍外，对栈还有一些其他的操作，例如清空一个栈，销毁一个栈，计算栈的当前容量等。
- 我们这节课的安排就这些！
- 哦还有，利用栈的原理，给大家讲解一道例题。
- 题目：利用栈的特点，将用户输入的二进制数转换为十进制数。（大家可以先思考）



## 清空一个栈

- 所谓清空一个栈，就是将栈中的元素全部作废，但栈本身物理空间并不发生改变（不是销毁）。
- 因此我们只要将s->top的内容赋值为s->base即可，这样s->base等于s->top，也就表明这个栈是空的了。这个原理跟高级格式化只是但单纯地清空文件列表而没有覆盖硬盘的原理是一样的。
- 代码清单：

```
ClearStack(sqStack *s){  
    s->top = s->base;  
}
```





## 销毁一个栈

- 销毁一个栈与清空一个栈不同，销毁一个栈是要释放掉该栈所占据的物理内存空间，因此不要把销毁一个栈与清空一个栈这两种操作混淆。
- 代码清单：

```
XXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOO
XXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOO
XXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOO
XXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOO
XXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOOXXOO
XXOOXXOOXXOOXXOO
```



## 销毁一个栈

```
DestroyStack(sqStack *s){  
    int i, len;  
    len = s->stackSize;  
    for( i=0; i < len; i++){  
        free( s->base );  
        s->base++;  
    }  
    s->base = s->top = NULL;  
    s->stackSize = 0;  
}
```



## 计算栈的当前容量

- 计算栈的当前容量也就是计算栈中元素的个数，因此只要返回`s.top-s.base`即可。
- 注意，栈的最大容量是指该栈占据内存空间的大小，其值是`s.stackSize`，它与栈的当前容量不是一个概念哦。
- 代码清单：

```
int StackLen(sqStack s)
{
    return(s.top - s.base); // 初学者需要重点讲解
}
```



## 实例分析

- 题目：利用栈的数据结构特点，将二进制转换为十进制数。
- 分析：地球人都知道，二进制数是计算机数据的存储形式，它是由一串0和1组成的，每个二进制数转换成相应的十进制数方法如下：

$$(X_n X_{n-1} \dots X_3 X_2 X_1)_2 = X_1 * 2^0 + X_2 * 2^1 + \dots + X_n * 2^{(n-1)}$$

- 一个二进制数要转换为相应的十进制数，就是从最低位起用每一位去乘以对应位的积，也就是说用第n位去乘以 $2^{(n-1)}$ ，然后全部加起来。





## 实例分析

- 由于栈具有后进先出的特性，例如我们输入11001001这样的二进制数，如图：

top →

1
0
0
1
0
0
1
1

base →

