

数据结构和算法

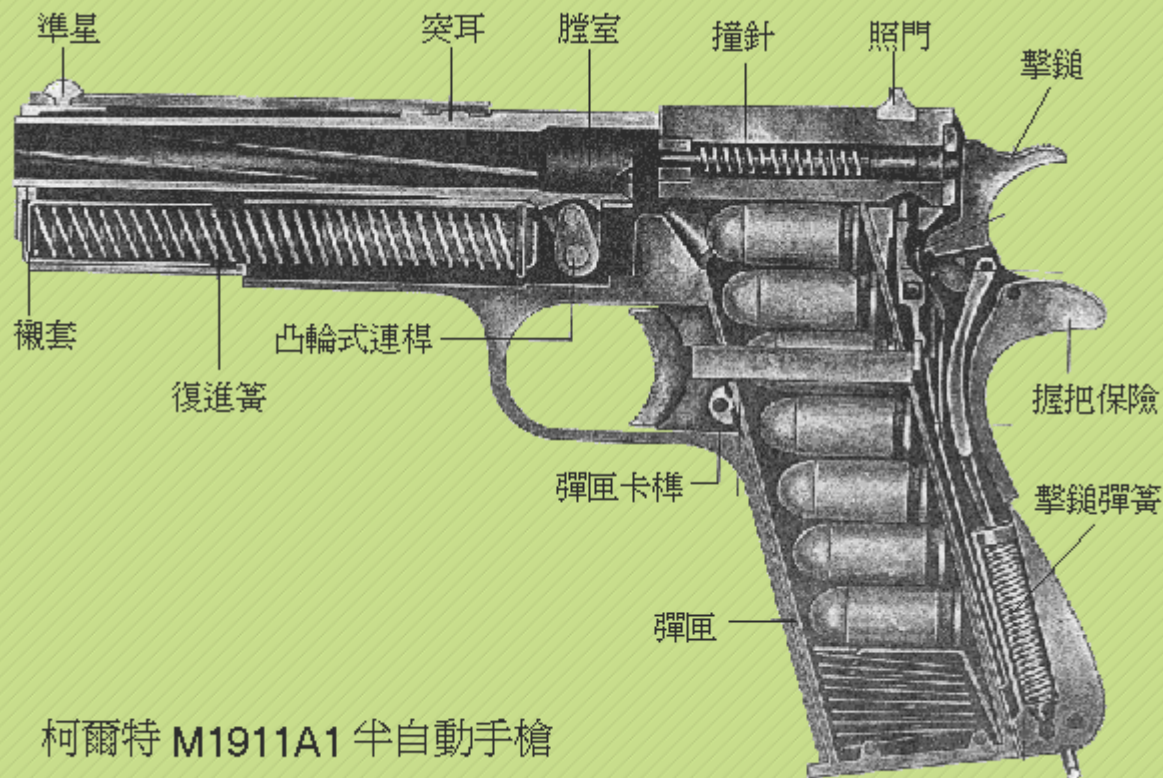
作者: 小甲鱼

让编程改变世界

Change the world by program



棧和队列



柯爾特 M1911A1 半自動手槍



栈的定义

- 栈是一种重要的线性结构，可以这样讲，栈是前面讲过的线性表的一种具体形式。
- 就像我们刚才的例子，栈这种后进先出的数据结构应用是非常广泛的。在生活中，例如我们的浏览器，每点击一次“后退”都是退回到最近的一次浏览网页。
- 例如我们Word, Photoshop等的“撤销”功能也是如此。再例如我们C语言的函数，也是利用栈的基本原理实现的。



栈的定义

- 官方定义：栈（Stack）是一个**后进先出**（Last in first out, LIFO）的线性表，它要求**只在表尾**进行删除和插入操作。
- 小甲鱼定义：所谓的栈，其实也就是一个特殊的线性表（顺序表、链表），但是它在操作上有一些特殊的要求和限制：
 - 栈的元素必须“后进先出”。
 - 栈的操作只能在这个线性表的表尾进行。
 - 注：对于栈来说，这个表尾称为栈的栈顶（top），相应的表头称为栈底（bottom）。



栈的插入和删除操作

- 栈的插入操作称为压栈，入栈。
- 栈的删除操作称为弹栈。如同弹。
- 请看动画。

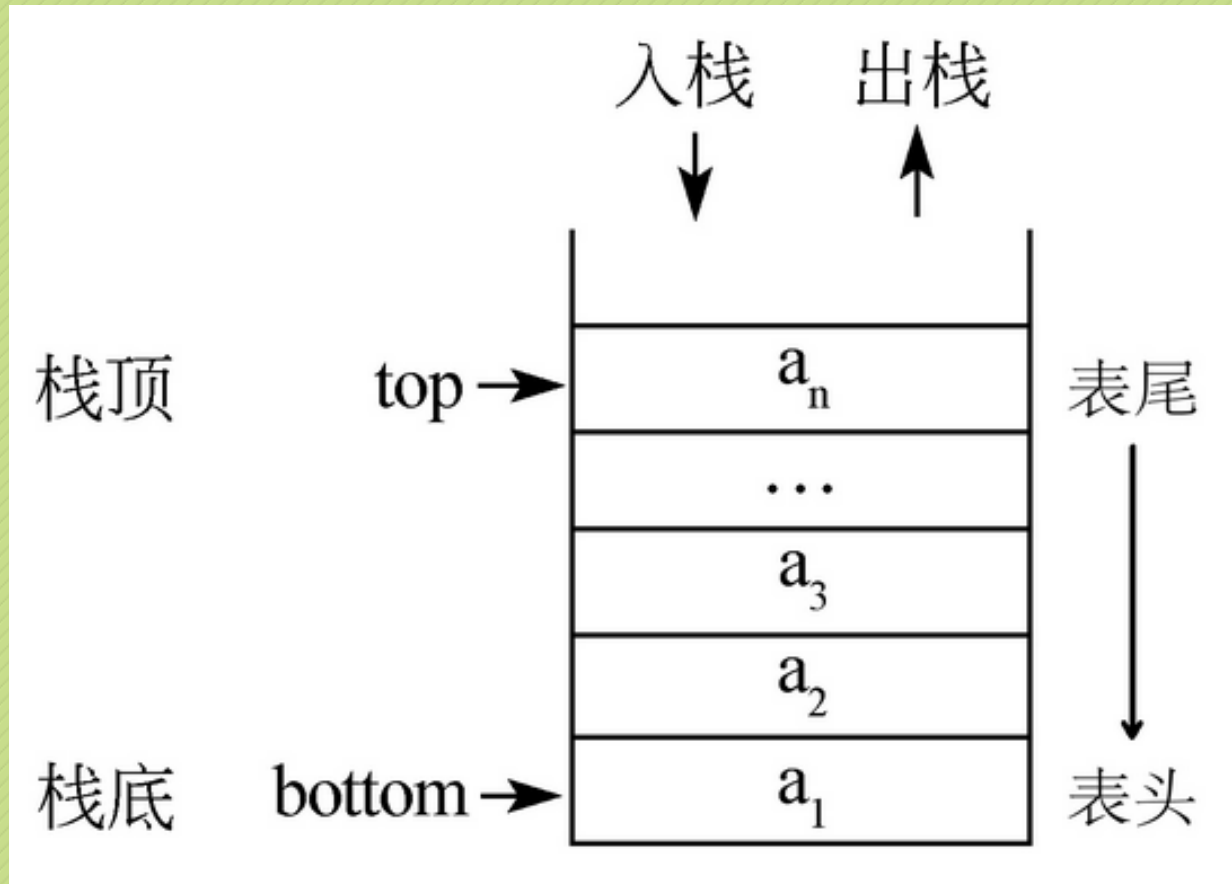


栈的顺序存储结构

- 因为栈的本质是一个线性表，线性表有两种存储形式，那么栈也有分为栈的顺序存储结构和栈的链式存储结构。
- 最开始栈中不含有任何数据，叫做空栈，此时栈顶就是栈底。然后数据从栈顶进入，栈顶栈底分离，整个栈的当前容量变大。数据出栈时从栈顶弹出，栈顶下移，整个栈的当前容量变小。
- 妈呀，说啥呢？
- 请结合下图发挥想象力理解并记住：



栈的顺序存储结构



栈的顺序存储结构

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

- 这里定义了一个顺序存储的栈，它包含了三个元素：base, top, stackSize。其中base是指向栈底的指针变量，top是指向栈顶的指针变量，stackSize指示栈的当前可使用的最大容量。



创建一个栈

```
#define STACK_INIT_SIZE 100
initStack(sqStack *s)
{
    s->base = (ElemType *)malloc( STACK_INIT_SIZE *
    sizeof(ElemType) );
    if( !s->base )
        exit(0);
    s->top = s->base; // 最开始，栈顶就是栈底
    s->stackSize = STACK_INIT_SIZE;
}
```



入栈操作

- 入栈操作又叫压栈操作，就是向栈中存放数据。
- 入栈操作要在栈顶进行，每次向栈中压入一个数据，top指针就要+1，知道栈满为止。
- 请看代码挺小甲鱼详细解释：Push.c



出栈操作

- 出栈操作就是在栈顶取出数据，栈顶指针随之下移的操作。
- 每当从栈内弹出一个数据，栈的当前容量就-1。
- 代码清单：

```
Pop(sqStack *s, ElemType *e)
{
    if( s->top == s->base ) // 栈已空是也
        return;
    *e = *(s->top);
    s->top--;
}
```

