
线性表
栈
队列

什么是栈？

【Question】计算机是如何实现进制转换的?

例如：$(1348)_{10}=(2504)_8$ 其运算过程如下：

N	N div 8	N mod 8
1348	168	4
168	21	0
21	2	5
2	0	2

计算顺序 ↓

↑ 输出顺序

启示：需要有种存储方法，能顺序存储运算数，并在需要时“倒序”输出！

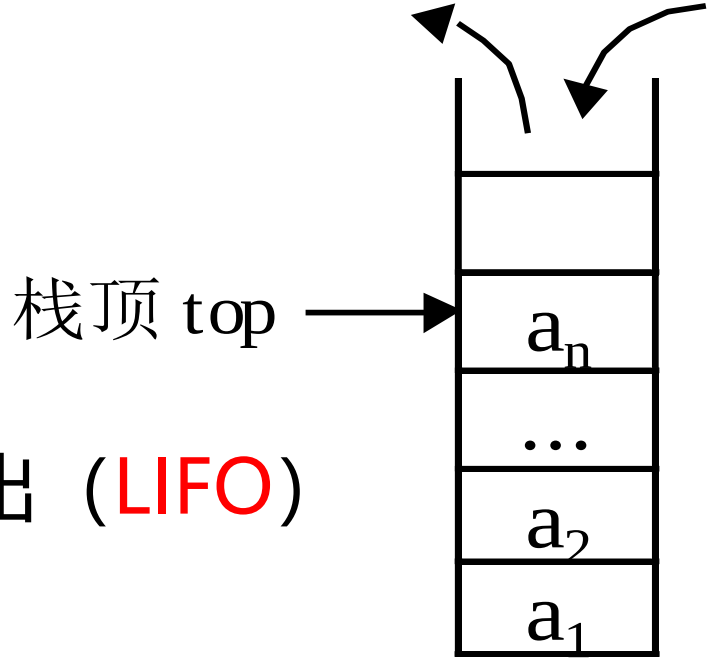
教材P71：

栈是一种特殊的线性表。其特殊性在于限定插入和删除数据元素的操作只能在线性表的一端(即表尾）进行。如下所示：

a_1, a_2, a_3, …, a_n ——插入和删除端

栈顶(top)：进行插入和删除的一端，是浮动的

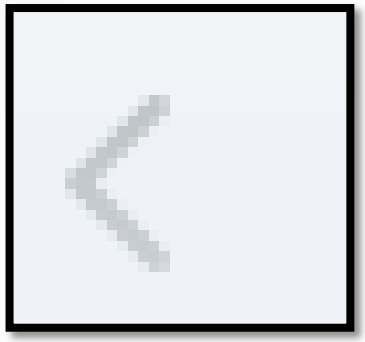
栈底(bottom):固定端



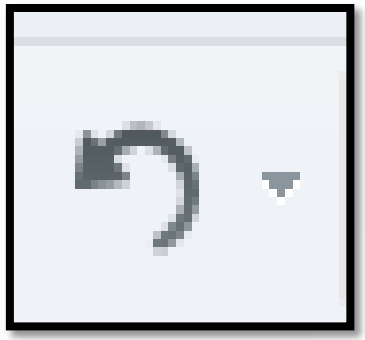
特点：先进后出（FILO）或后进先出（LIFO）

我们身边的栈

浏览器的“后退”按钮



Word的“撤销”按钮



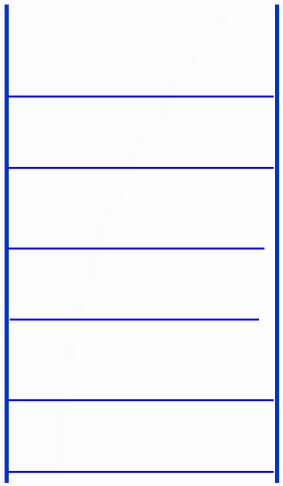
【Project 5】利用 顺序栈 实现进制转换

例如：$(1348)_{10}=(2504)_8$ 其运算过程如下：

N	N div 8	N mod 8
1348	168	**4**
168	21	**0**
21	2	**5**
2	0	**2**

计算顺序 ↓

↑ 输出顺序



算法：余数一个个入栈，然后再一个一个出栈即可。

（1）顺序存储结构栈的定义

```
#define MAXSIZE 100
typedef struct
{
   ElemType elem[MAXSIZE];
    int top;
} SqStack;
```

顺序表的定义

```
#define MAXSIZE 100
typedef struct
{ElemType elem[MAXSIZE];
 int length;
}SqList;
```

（2）初始化操作：建立一个空栈

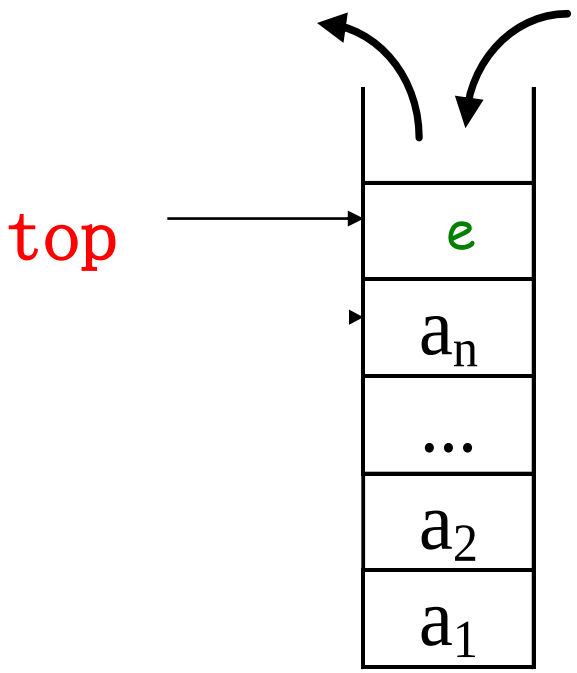
```
void Initstack(SqStack *S)
{    S->top=-1;    }
```

★判断一个栈S为空的条件是：S->top==-1

--压栈（入栈）Push(*S, e)

```
void Push(SqStack *S,ElemType e)
{
        if ( S->top==MAXSIZE-1)
       { printf("栈已满\n");return; }
            S->top++;
            S->elem[ S->top] = e;
}
```

算法时间复杂度：O（1）

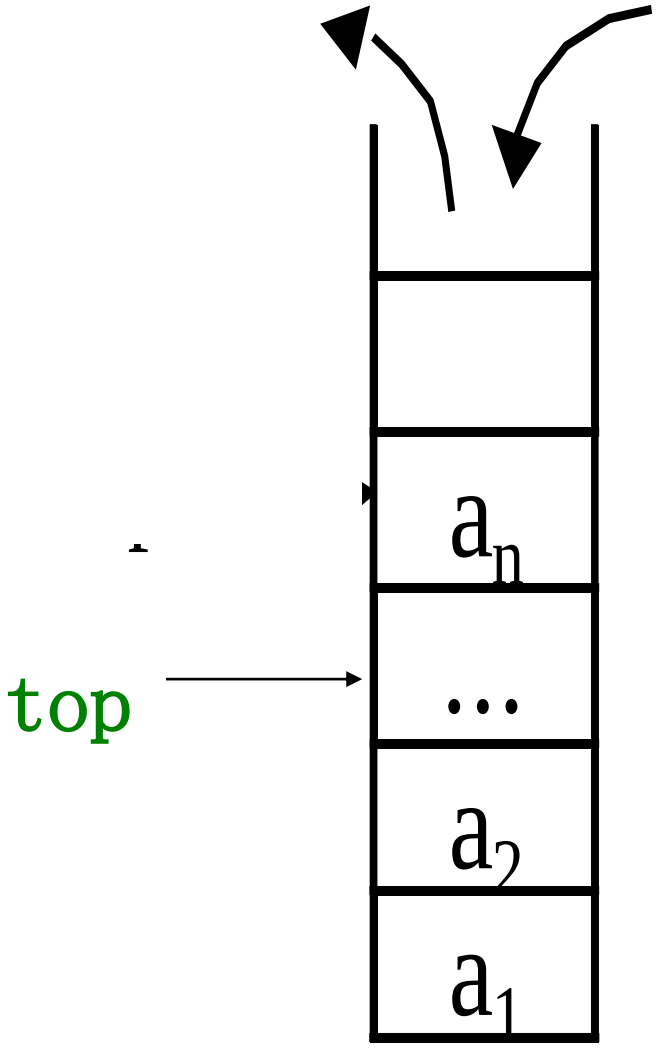


--出栈Pop(*S, *e)

```
void Pop(SqStack *S,ElemType *e)
{
        if (S->top==-1)
        {printf("栈空\n");return;}
         *e=S->elem[S->top];
        S->top--;

}
```

算法时间复杂度：O（1）



```
void conversion(  )
{
        Initstack(S);
        scanf (“%d”,N);
        while(N){
            push(S，N%8);
            N=N/8;
         }
        while(! Stackempty(S)){
            pop(S,e);
            printf(“%d”,e);
        }
    }
```

算法：取栈顶数据元素GetTop(S, *e)

```
void GetTop(SqStack S,ElemType *e)
{
        if (S.top==-1)
        {printf("栈空\n");return;}
       *e= S.elem[S.top];
}
```

算法时间复杂度：O（1）