四川師範大學
SICHUAN NORMAL UNIVERSITY

第3章 栈和队列

DATA STRUCTURE

计算机科学学院 刘 芳

四川師範大學
SICHUAN NORMAL UNIVERSITY

第3章 栈和队列

3.1 栈
3.2 栈的应用举例
3.3 栈与递归
3.4 队列

计算机科学学院　刘 芳

四川師範大學
SICHUAN NORMAL UNIVERSITY

3.1 栈

- 3.1.1 栈的定义
- 3.1.2 栈的顺序表示和实现
- 3.1.3 栈的链式表示和实现

刘　芳 LiuFang

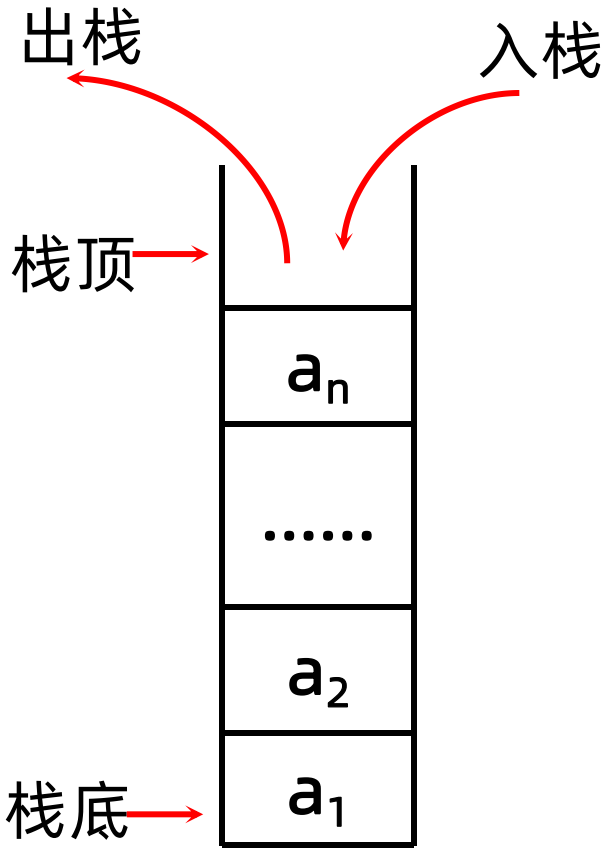
四川師範大學
SICHUAN NORMAL UNIVERSITY

3.1.1 栈的定义

刘　芳 LiuFang

栈的定义

■栈(Stack)是限定在表尾进行插入和删除运算的线性表。



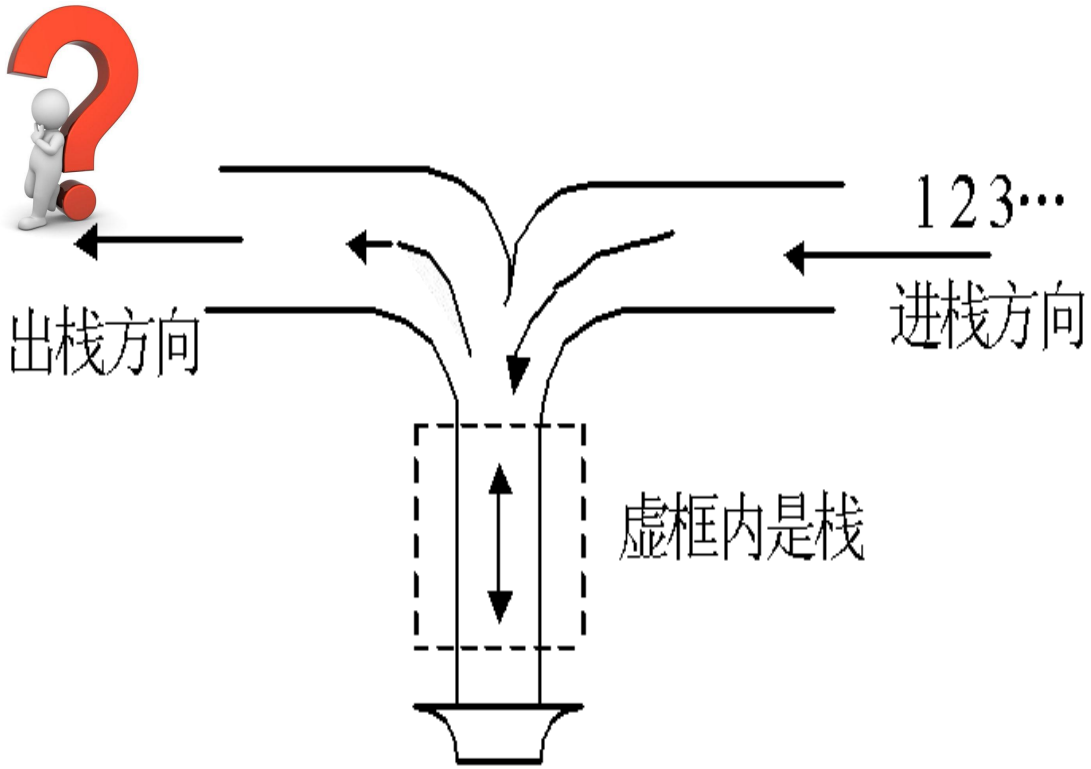
●栈的特性：

后进先出（LIFO，Last In First Out)

练习1

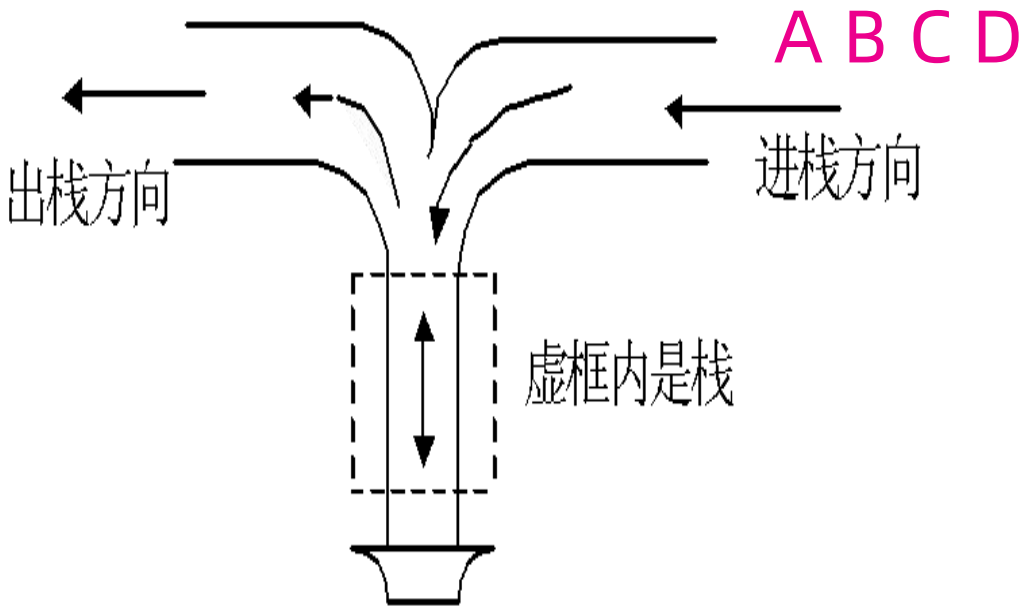
- 若入栈序列为1 2 3，可以得到哪些出栈序列？

1 2 3	✔
1 3 2	✔
2 1 3	✔
2 3 1	✔
3 1 2	✘
3 2 1	✔



课后练习

- 若设入栈序列为A B C D，可以得到哪些出栈序列？



栈的抽象数据类型定义

ADT Stack{

数据对象：$D=\{a_i|a_i \in ElemSet; 1 \le i \le n, n \ge 0;\}$

数据关系：$R=\{<a_i, a_{i+1}>| a_i, a_{i+1} \in D, i=1,2,...,n-1\}$

基本操作：

InitStack(&S)

DestroyStack(&S)

StackEmpty(S)

StackLength(S)

GetTop(S,&e)

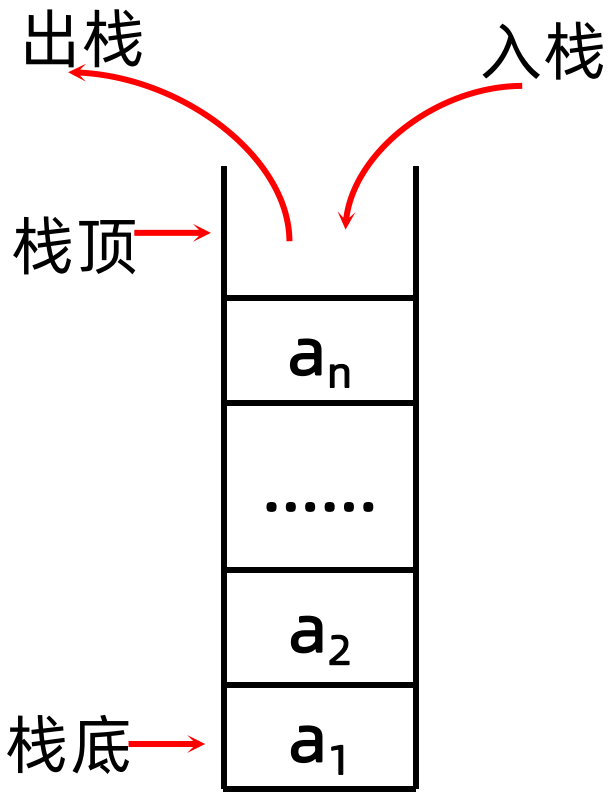
Push(&S,e)

Pop(&S,&e)

} ADT Stack

本节要点

- 栈是一种特殊的线性表，它只能表尾进行插入和删除运算。



	线性表	栈
逻辑结构	线性结构	线性结构
存储结构	顺序表、链表	顺序栈、链栈
运算规则	随机插入删除	后进先出(LIFO)

四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听

业精于勤,荒于嬉;行成于思,毁于随.

四川師範大學
SICHUAN NORMAL UNIVERSITY

第3章 栈和队列

3.1 栈
3.2 栈的应用举例
3.3 栈与函数
3.4 队列

计算机科学学院　刘 芳

四川師範大學
SICHUAN NORMAL UNIVERSITY

3.1 栈

- 3.1.1 栈的定义
- 3.1.2 栈的顺序表示和实现
- 3.1.3 栈的链式表示和实现

刘　芳 LiuFang

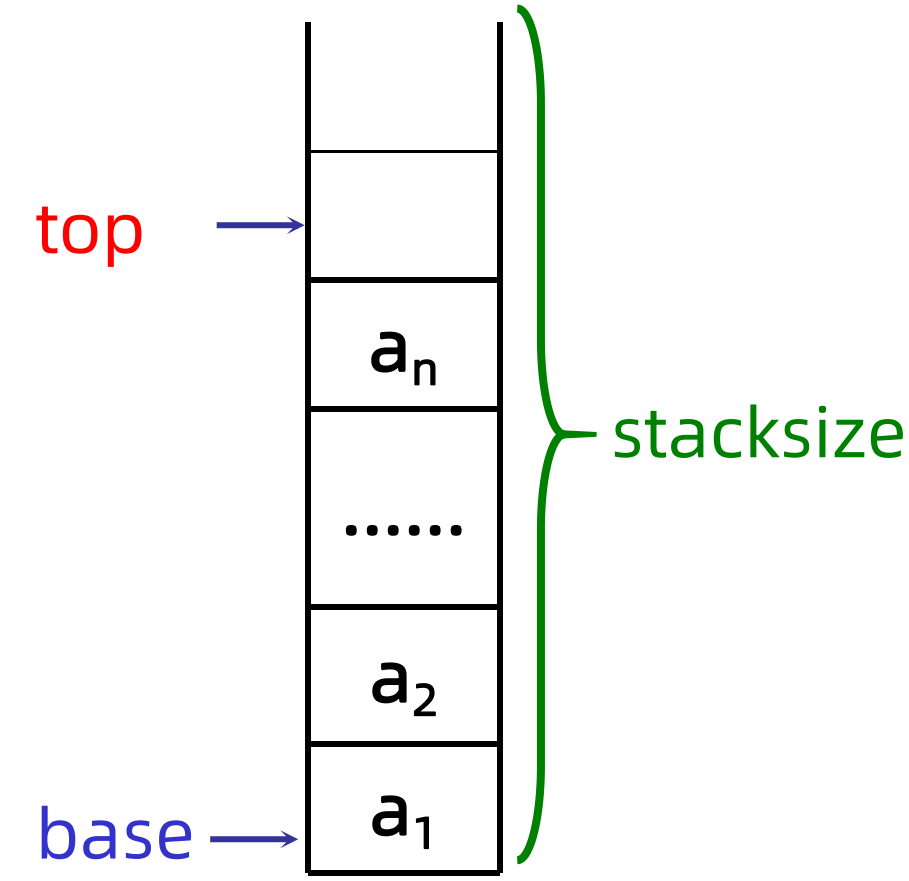
四川師範大學
SICHUAN NORMAL UNIVERSITY

3.1.2 栈的顺序表示和实现

刘　芳 LiuFang

顺序栈的C语言描述

```
#define STACK_INIT_SIZE 100
#define STACKINCREMENT 10
typedef struct {
        SElemType *base;
        SElemType *top;
        int stacksize;
      } SqStack;
SqStack S;
```

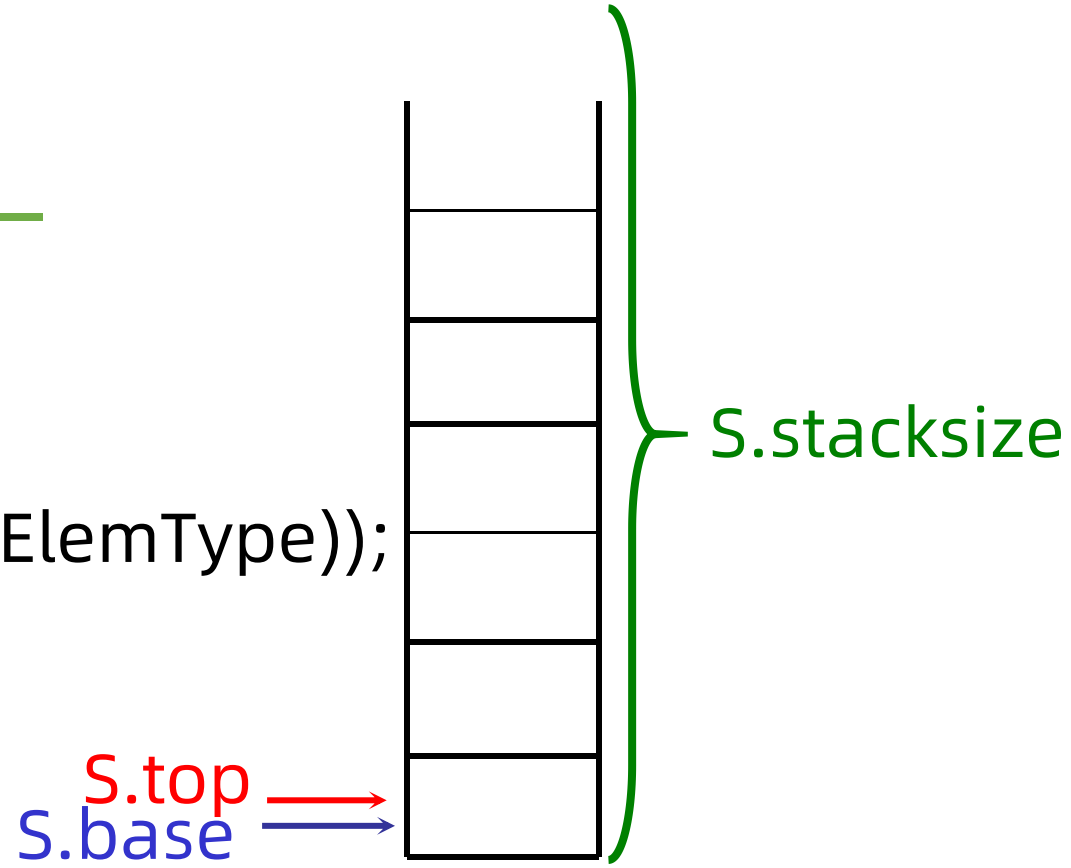


顺序栈的运算

1. 初始化空栈　　InitStack(&S)
2. 判断栈空　　StackEmpty(S)
3. 求栈的长度　　StackLength(S)
4. 取栈顶　　GetTop (S,&e)
5. 入栈　　Push (&S,e)
6. 出栈　　Pop (&S,&e)
7. 栈的销毁　　DestroyStack(&S)

1.初始化空栈

```
Status InitStack(SqStack &S){
    S.base=(SElemType*)
        malloc(STACK_INIT_SIZE*sizeof（SElemType));
    if (!S.base) exit （OVERFLOW）；
    S.top=S.base;
    S.StackSize=STACK_INIT_SIZE;
    return OK;
}
```

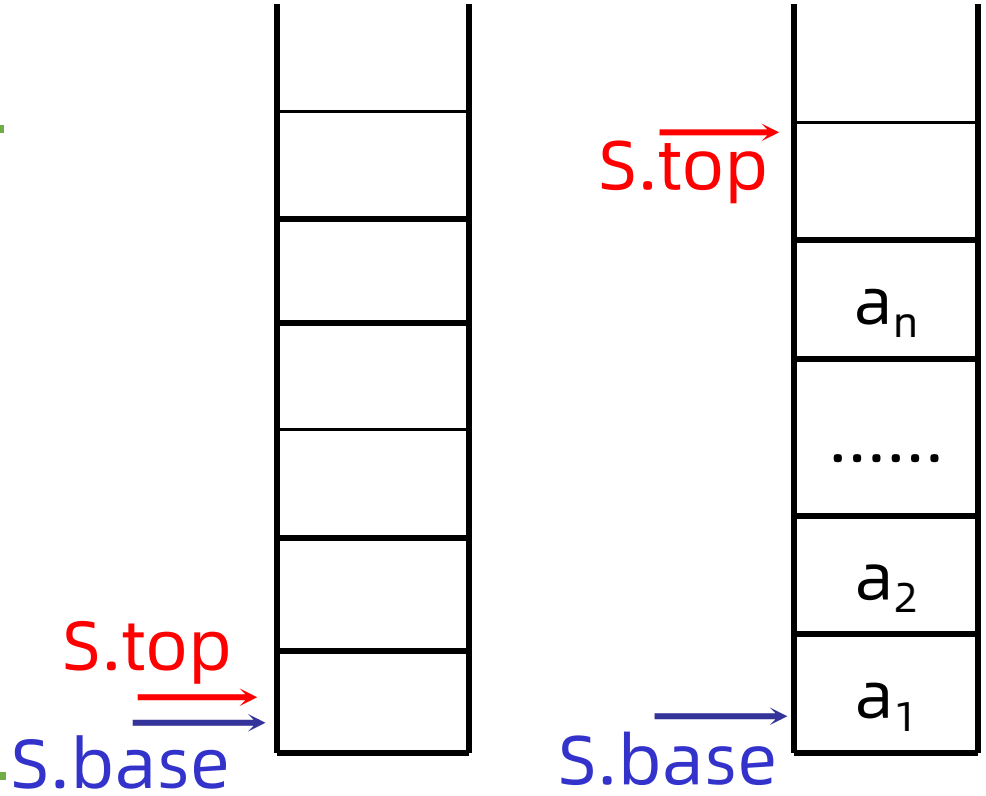


2. 判断栈空

```
Status StackEmpty(SqStack S){
    return  (S.base==S.top) ;
}
```

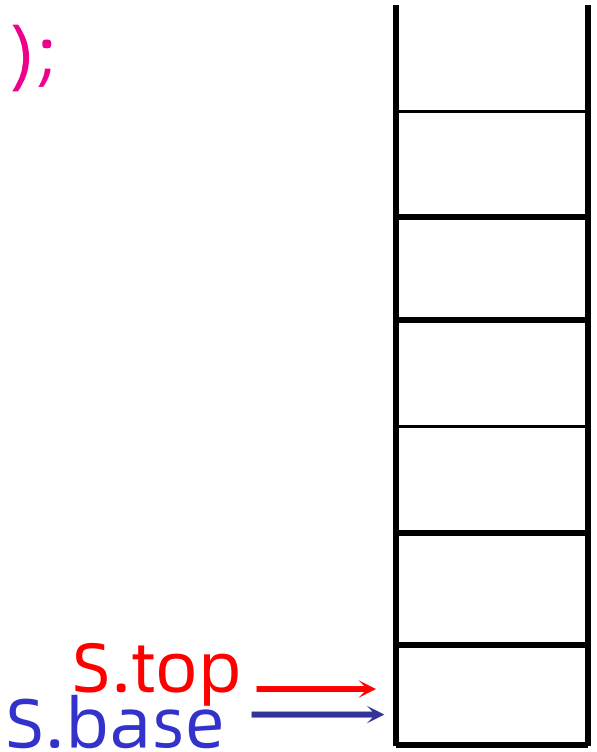
3. 求栈的长度

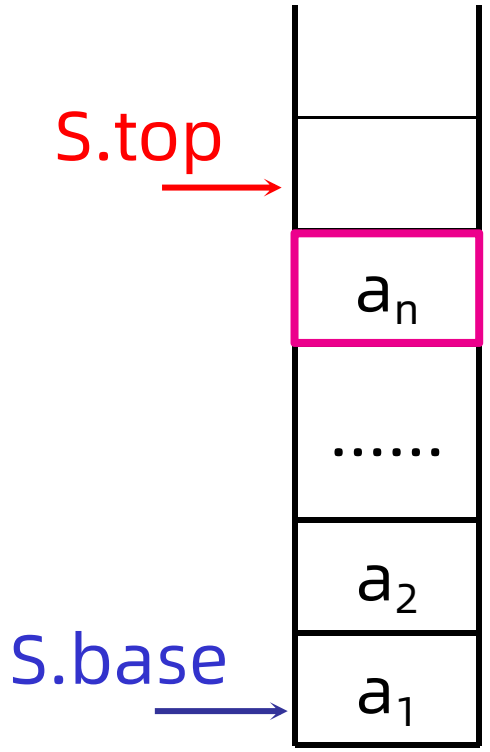
```
int StackLength(SqStack S){
    return S.top - S.base;
}
```



4. 取栈顶

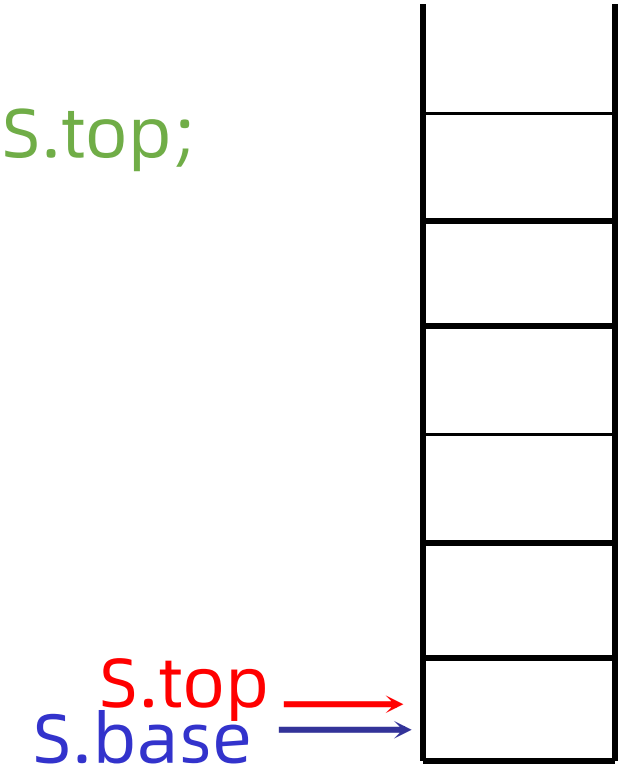
```
Status GetTop(SqStack S,SElemType &e){
    if (S.base==S.top) return ERROR;
    e=* ( S.top - 1);
    return OK;
}
```

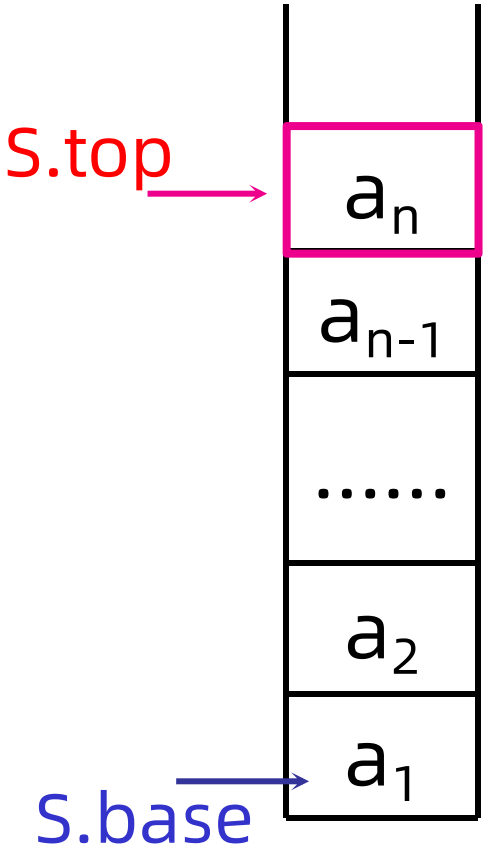




5. 出栈

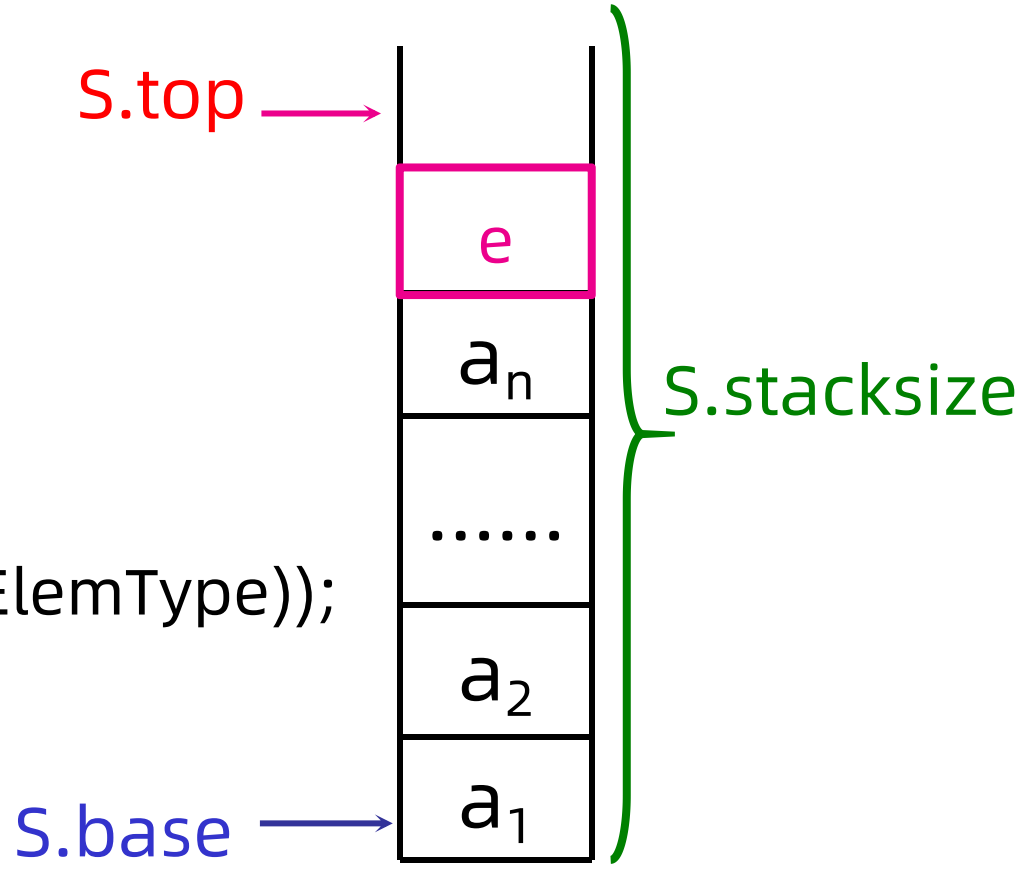
```
Status Pop(SqStack &S,SElemType &e){
    if (S.base==S.top) return ERROR;
    e=*-- S.top;
     //- -S.top;e=*S.top;
    return OK;
}
```





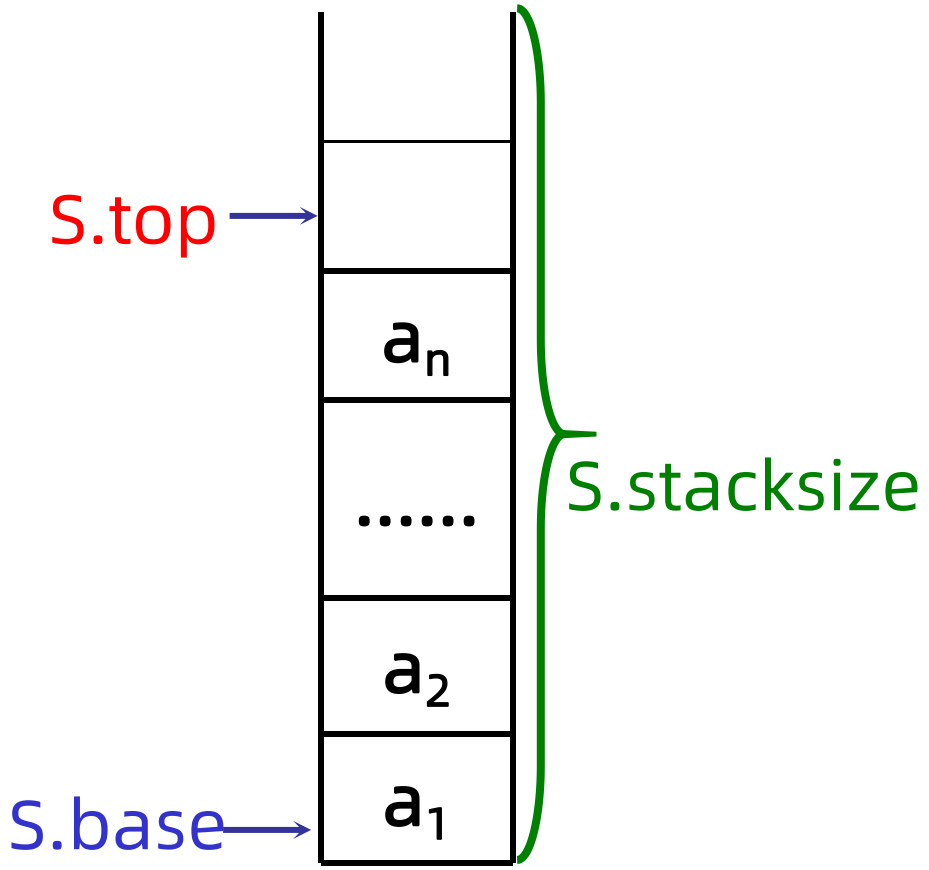
6. 入栈

```
Status Push(SqStack &S,SElemType e){
    if (S.top- S.base>=S.stacksize){
      newbase=(SElemType *)realloc(S. base
       (S.stacksize+STACKINCREMENT)*sizeof(SElemType));
      if (!newbase) exit (OVERFLOW);
      S.base=newbase;
      S.stacksize+=STACKINCREMENT;
    }
   *S.top++=e;        //*S.top=e;S.top++;
   return OK;
}
```



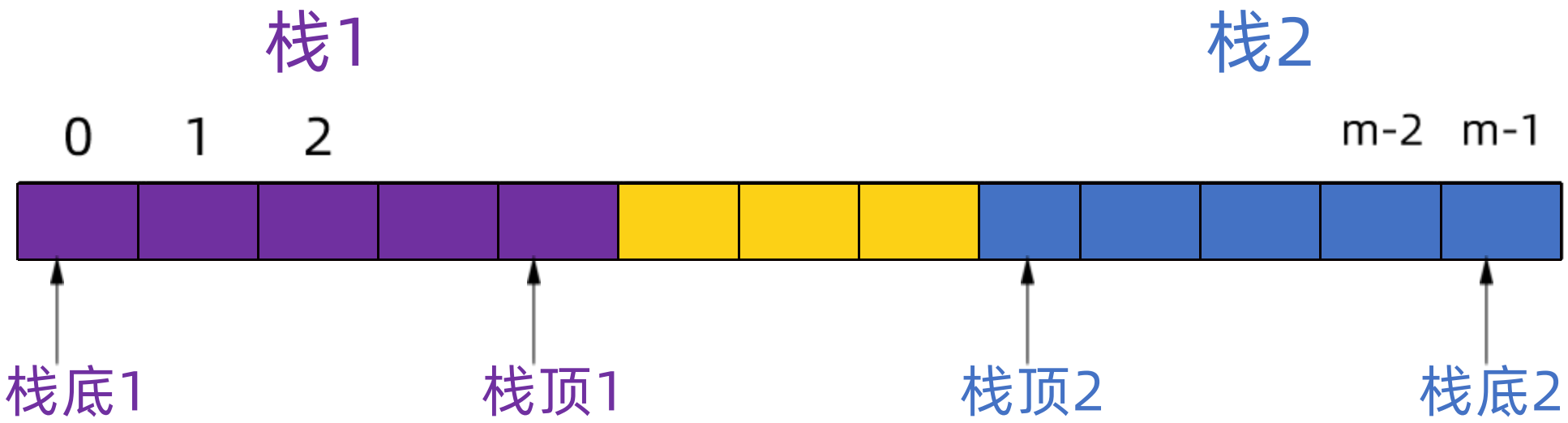
本节要点：栈的顺序表示与实现

1. 初始化空栈　　InitStack(&S)
2. 判断栈空　　StackEmpty(S)
3. 求栈的长度　　StackLength(S)
4. 取栈顶　　GetTop (S,&e)
5. 出栈　　Pop (&S,&e)
6. 入栈　　Push (&S,e)
7. 栈的销毁　　DestroyStack(&S)



拓展思考：

- 两个栈如何共享一片连续的存储空间？



“底设两端、相向而动、迎面增长”

四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听

业精于勤,荒于嬉;行成于思,毁于随.

四川師範大學
SICHUAN NORMAL UNIVERSITY

第3章 栈和队列

3.1 栈
3.2 栈的应用举例
3.3 栈与函数
3.4 队列

计算机科学学院　刘 芳

四川師範大學
SICHUAN NORMAL UNIVERSITY

3.1 栈

- 3.1.1 栈的定义
- 3.1.2 栈的顺序表示和实现
- 3.1.3 栈的链式表示和实现

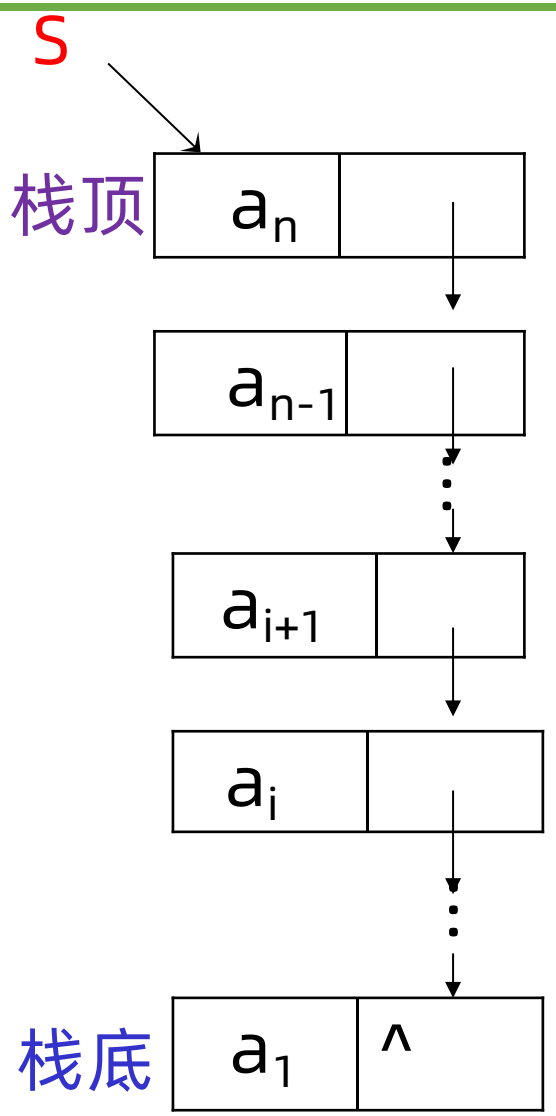
刘　芳 LiuFang

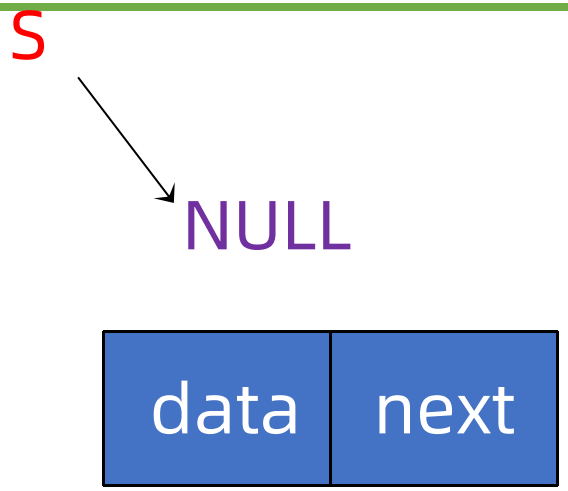
四川師範大學
SICHUAN NORMAL UNIVERSITY

3.1.3 栈的链式表示和实现

刘　芳 LiuFang

链栈的结构和C语言定义






```
typedef  struct  Node{
    SElemType data;
    struct  Node *next;
 } SNode,  *  LinkStack;
LinkStack  S;
```

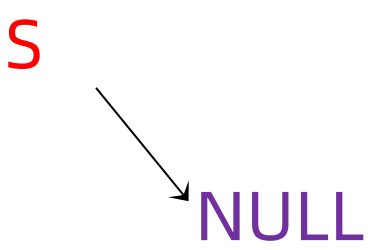
注意：链栈中一般都不设置头结点。

链栈的运算

1. 初始化空栈 InitStack(&S)
2. 判断栈空 StackEmpty(S)
3. 求栈的长度 StackLength(S)
4. 取栈顶 GetTop (S,&e)
5. 入栈 Push (&S,e)
6. 出栈 Pop (&S,&e)
7. 栈的销毁 DestroyStack(&S)

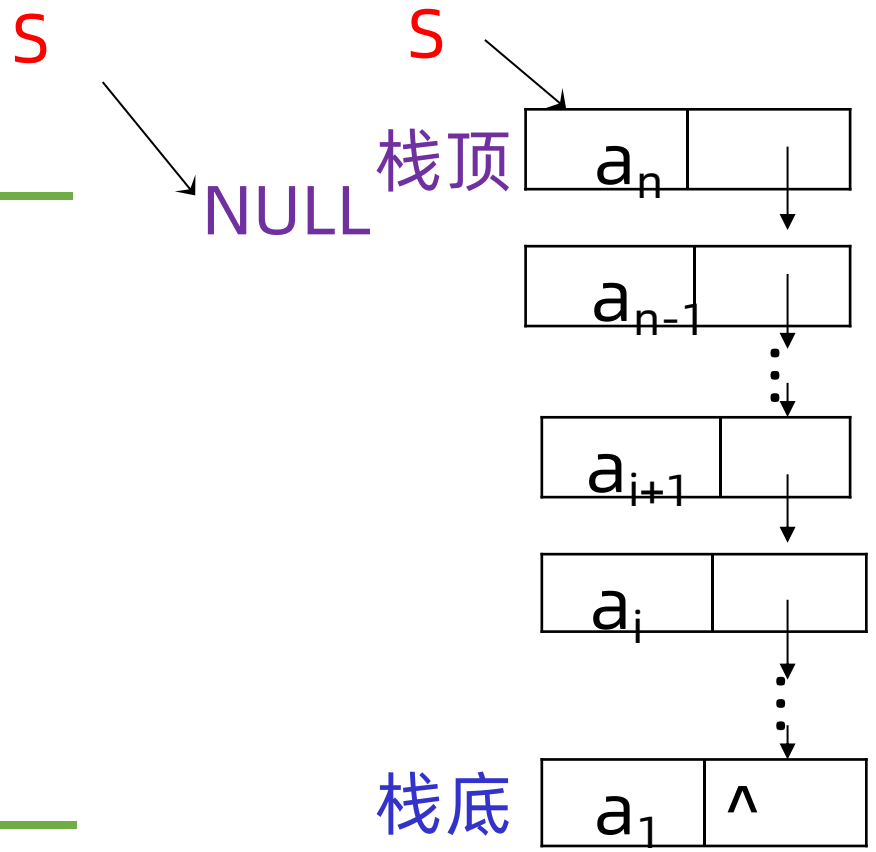
1.初始化空栈

```
Status InitStack(LinkStack &S){
    S=NULL;
    return OK;
}
```



2. 判断栈空

```
Status StackEmpty(LinkStack S){
    if  (S==NULL) return TRUE;
    else    return FALSE;
}
```

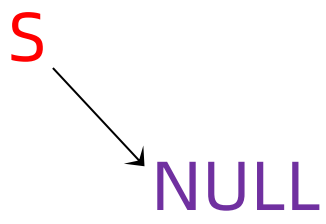


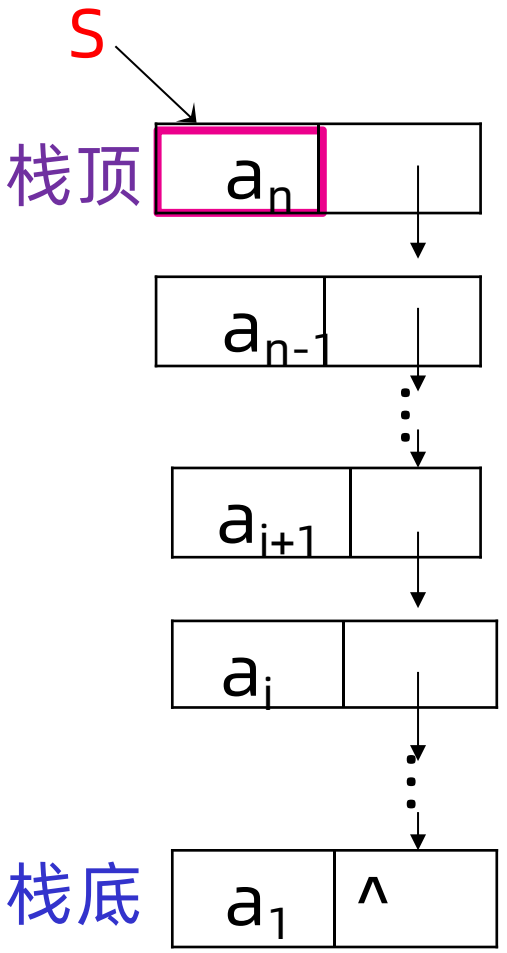
3. 求栈的长度

```
int  StackLength(LinkStack S){
    for (i=0,p=S ;  p  ; i++,p=p->next)  ;
    return i;
}
```

4. 取栈顶

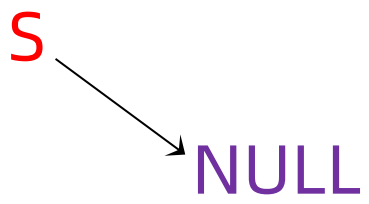
```
Status GetTop(LinkStack S,SElemType &e){
    if ( S==NULL) return ERROR;
    e=S->data;
    return OK;
}
```

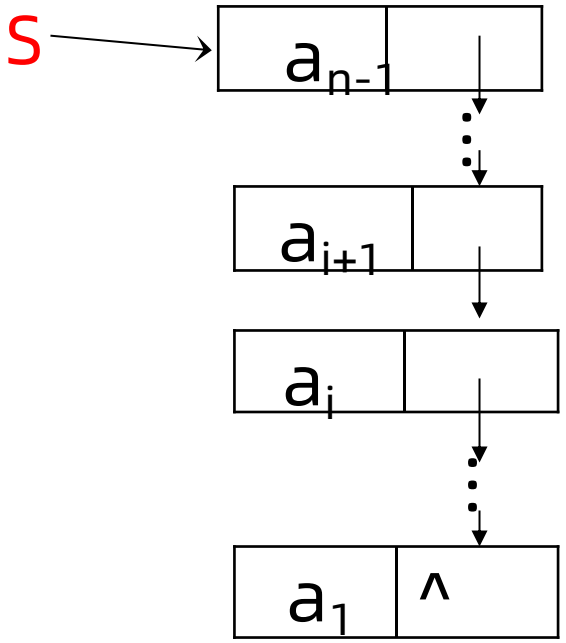




5. 出栈

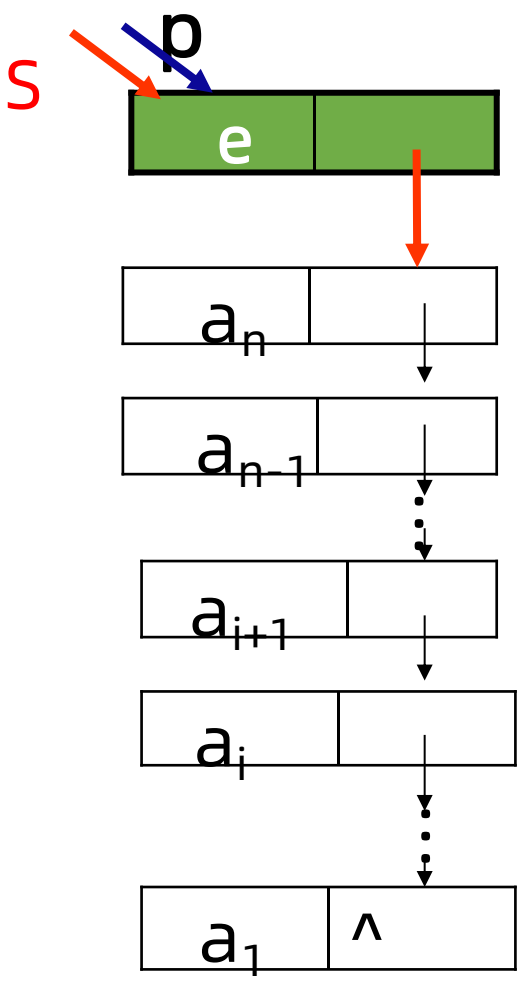
```
Status Pop(LinkStack &S,SElemType &e){
    if (S==NULL) return ERROR;
    p=S;
    S=S->next;
    e=p->data;
    free(p);
    return OK;
}
```





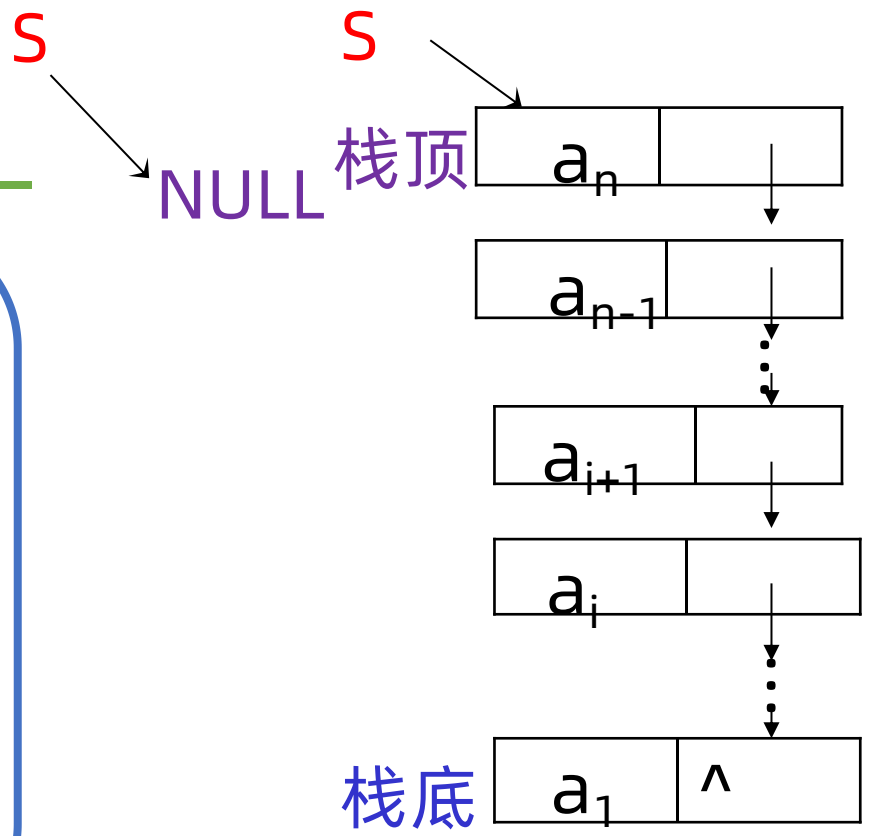
6. 入栈

```
Status Push(LinkStack &S,SElemType e){
    p=(LinkStack)malloc(sizeof(SNode));
    if (!p) return OVERFLOW;
    p->data=e;
    p->next=S;
    S=p;
    return OK;
}
```



本节要点：栈的链式表示与实现

1. 初始化空栈 InitStack(&S)
2. 判断栈空 StackEmpty(S)
3. 求栈的长度 StackLength(S)
4. 取栈顶 GetTop (S,&e)
5. 出栈 Pop (&S,&e)
6. 入栈 Push (&S,e)
7. 栈的销毁 DestroyStack(&S)



四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听

业精于勤,荒于嬉;行成于思,毁于随.