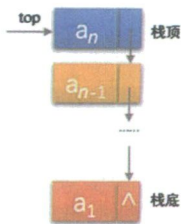


好的办法是把栈顶放在单链表的头部（如右图所示）。另外，都已经有了栈顶在头部了，单链表中比较常用的头结点也就失去了意义，通常对于链栈来说，是不需要头结点的。

对于链栈来说，基本不存在栈满的情况，除非内存已经没有可以使用的空间，如果真的发生，那此时的计算机操作系统已经面临死机崩溃的情况，而不是这个链栈是否溢出的问题。

但对于空栈来说，链表原定义是头指针指向空，那么链栈的空其实就是 $top=NULL$ 的时候。链栈的结构代码如下：



```

/* 链栈结构 */
typedef struct StackNode
{
    SElemType data;
    struct StackNode *next;
}StackNode,*LinkStackPtr;

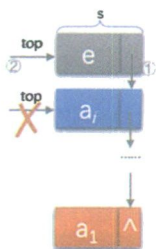
typedef struct
{
    LinkStackPtr top;
    int count;
}LinkStack;
    
```

注：栈的链栈相关代码请参看代码目录下“/第4章栈与队列/03链栈_LinkStack.c”。

链栈的操作绝大部分都和单链表类似，只是在插入和删除上，特殊一些。

4.6.2 栈的链式存储结构——进栈操作

对于链栈的进栈push操作，假设元素值为e的新结点是s，top为栈顶指针，示意图如右图所示，代码如下。



```

/* 插入元素e为新的栈顶元素 */
Status Push(LinkStack *S,SElemType e)
{
    LinkStackPtr s=(LinkStackPtr)malloc(sizeof(StackNode));
    s->data=e;
    s->next=S->top; /* 把当前的栈顶元素赋值给新结点的直接后继，见图中① */
    S->top=s; /* 将新的结点s赋值给栈顶指针，见图中② */
    S->count++;
    return OK;
}
    
```