# 栈和队列

顺序栈：是利用一组地址连续的存储单元依次存放自栈底到栈顶的数据元素

构成：（1）一维数组：data

1. 栈顶指针top

栈空：top=-1

栈满：top=maxSize=-1

概念：栈是一种特殊的线性表：插入、删除受限

特殊：只能在一端进行操作，称为栈顶，top

入栈：插入数据

出栈：删除数据

入栈序列为

栈的顺序存储

1. 栈的初始化 构造函数 \_\_init\_\_(self)
2. 构造函数：一种特殊的方法，创建对象，初始化对象，对成员变量赋初值

栈的链式存储

节点构成：（1）数据：data

1. Next

链栈：

进栈函数

思路：

1. 创建一个新的结点，并将要进栈的元素存入该结点的数据域

tStackNode = StackNode()

tStackNode.data = da

1. 将新的结点的指针域向栈顶结点指针域指向的结点

tstackNode.next = self.top.next

1. 将栈顶结点的指针域指向新结点

Self.top.next = tStackNode

出栈函数

1. 判断栈是否为空
2. 如果为空，则无法执行出栈操作，给出栈为空提示，否则执行(3)
3. 记下此时栈顶结点指针域指向的结点
4. 秀发栈顶结点的指针域，在其中存入(3)记下结点指针域的值
5. 将数值域为x的结点出栈。

链式栈

构成：（1）数据：data

1. 指针：next

约定：不带头结点

优点：多个栈可共享空间

Top 栈顶结点

|  |  |
| --- | --- |
| 28 | ... |
| 12 | ... |
| top |  |

出栈:

使用tstackNode记住栈顶结点指向的结点

tstackNode = self.top.next

Self.top.next = tstackNode.next

等同于self.top.next.next

Return tstackNode.data

队列

队列包括顺序存储结构和链式储存结构

概念:一种特殊的线性表，插入在限定在表的某一端进行，删除在表的另一端进行

队尾：允许插入的一端

队头：允许删除的一端

空队列：不含任何元素的队列

出队 a1 a2 a3 ... an 入队

队头 队尾

特点：先进先出(FIFO)，后进后出 (LILO)

基本运算：

初始化

销毁

进队

出队

取队头元素

判断队空

顺序队列：通常约定队尾指针指示队尾元素的当前位置，队头指针指示队头元素的前一个位置

顺序存储结构：（1）一维数组 data

1. 队头和队尾指针：front,rear

约定：队尾指针指示队尾元素的当前位置

队头指针指示队头元素的前一个位置

队空条件：front==rear

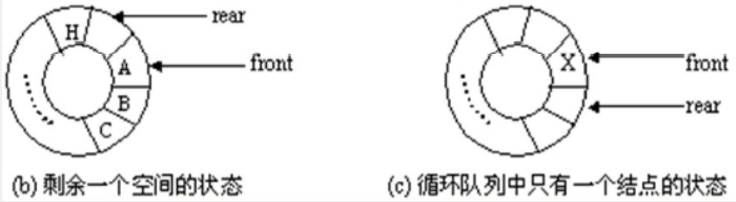
队满条件:rear=Maxszie-1

假溢出：用循环队列或环形队列解决

已知顺序队列sq:

1. 队空条件：sq.front==sq.rear
2. 队满条件:（sq.rear+1)%MaxSize==sq.front

front跟rear相邻，而且在rear前面



1. 元素x入队列
2. sq.rear=(sq.rear+1)%MaxSize
3. 将元素x存入sq.data[sq.rear]中：sq.data[sq.rear]=x
4. 队列顶元素x出队列
5. sq.front = (sq.front+1)%MaxSize
6. 取出队列元素:x=sq.data[sq.front]

注意：显然在这样设置的队满条件下，队满条件成立时队中还有以-一个空闲队列，也就是说这样的队中最多只能进MaxSize-1个元素

循环队列中队头，队尾指针变化

(rear+MaxSize-front)%MaxSize

# 链式队列

队列的链式存储结构简称为链队，它实际上是一个同时带有队头指针front和队尾指针rear的单链表

队头指针指向队头结点，队尾指针指向队尾结点即单链表的最后一个结点，并将队头和队尾指针结合起来构成链队结点



已知链队列ls

1. 队空条件：front==rear
2. 队满条件：不考虑（因为每个结点都是动态分布的）
3. 元素x入队：

创建结点P,将其插入到队尾，并由rear指

P.data = x

1. 出队操作:

删除队头的结点

进队运算操作

主要操作:创建一个新结点，将其链接到链队的末尾，并由rear指向它

出队

1. 队列例里有没有数据 front==rear

没有数据（空），队列为空，无法出队

1. 队列里有一个数据

If self.front.next.next == None:

Self.front.next=None

Self.rear=self.front

1. 队列里有2个或以上数据

tNode = self.front.next

Self.front = tNode.next

Return tNode.data