# 数据结构和算法概述

## 算法概述

### 算法概述

**算法**:是解决特定问题求解步骤的描述，在计算机中表现为指令的有序序列，并且指令表示一个或多个操作。

### 算法的特性

1、输入输出

算法具有零个和多个输入。至少有一个或多个输出。

2、有穷性

算法在执行有限的步骤之后，自动结束而不会出现无限训话你，并且每个步骤可以再接受的时间内完成。

3、确定性

算法的每一个步骤都具有确定的含义，不会具有二义性别。

4、可行性

算法的每一步骤都是可行的，也就是说，每一步都能够通过执兴有限的次数完成。

# 算法的时间复杂度

## 时间复杂度概述

时间复杂度的定义:在进行算法分析时。语句总的执行次数T(n)是关于问题规模n的函数，进而分析T(n)随n的变化情况并确定T(n)的数量级。算法的时间复杂度，也就是算法的时间度量，

记做:T(n)=O(f(n))

他表示随问题规模n的增大，算法执行时间的增长率和f(n)的增长率相同，称作算法的渐进时间复杂度，简称为时间复杂度。其中f(n)是问题规模n的某个函数。

这种使用大写O{}来变现时间复杂度的记法，我们称为大O记法。

一般情况峡，随着n的增大，T[n]增长最慢的算法为最优算法。

有算法的时间复杂度定义可知，求和算法 O(n)、O(1),O(n2)，我们分别给他们取了非官方的名称，O(1)叫做常数阶，O(n)叫做线性阶，O(n)叫做平方阶。

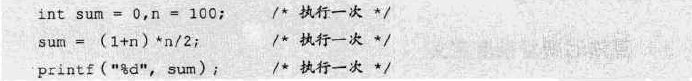
### 推导大O阶的方法

分析一个算法的时间复杂度：即推导大O阶

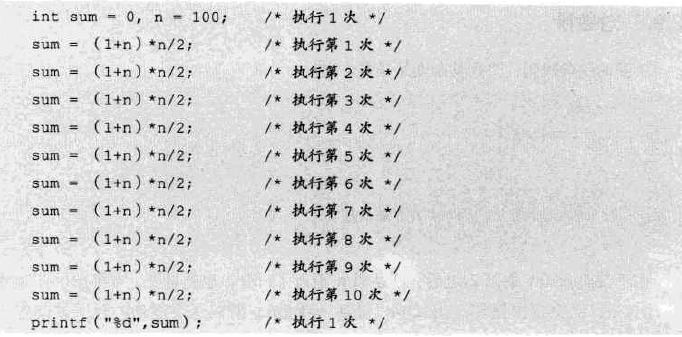
推导规则：

1. 用常数1取代运行时间中所有加法常数。
2. 在修改后的运行次数函数中，只保留最高阶项。
3. 如果最高阶项存在且不是1，则去除与这个项相乘的常数。

## 常数阶



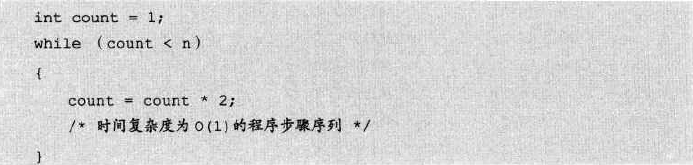
这个算法的运行函数是f(n)=3。根据我们推到大O阶的方法，第一步就是把常数项3改为1，。在保留最高阶时发现，他根本没有最高项，所以这个算法的时间复杂度为O(1);



Sum = (1+n)\*n/2 有10句，无论n为多少，这中与n的大小无关，执行恒定的算法，我们称之为具有O(1)的时间复杂度，又叫常数阶。不管常数为多少都记为O(1).

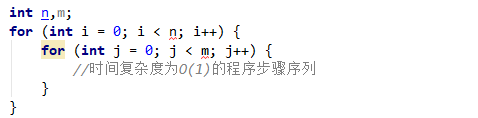
## 线性阶

### 对数阶



由于每次count乘以2之后，就距离n更近了一分。也就是说，有多少个2相乘，后大于n，则会退出循环。由2x=n，x = log2n，所以这个循环的时间复杂度为 O(log2n)

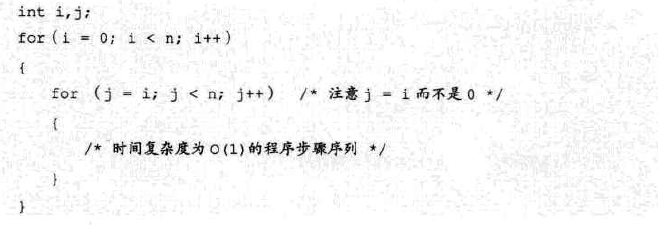
### 平方阶



对于外城循环，内层循环的时间复杂度为O(n)语句，再循环n次，所以这段代码的时间复杂度为O(n2)

如果外城循环的循环次数改为m,时间复杂度变为(m\*n),所以循环的时间复杂度为

循环的时间复杂度 = 循环体的复杂度\*循环运行的次数



# 线性表

## 线性表



### 线性表的定义

线性表:零个或多个数据元素的有限序列。

1. 顺序性，元素质之间是有顺序的。
2. 优先性，元素的个数是有限的。

线性表的长度：线性表元素的个数n(n>=0)定义线性表的长度，当n=0时，称为空表。

位序：非空表中每个数据元素都有一个确定的位置，a1a**2**a3….ai，i称之为元素ai在线性表中的位序。

### 线性表的抽象数据类型

ADT 线性表(List)

Data:线性表的数据对象集合为{a1,a2,…,an},每个元素的类型均为DataType.其中，除第一个元素a1外，每一个元素有且只有一个直接前驱元素，除了最后一个元素an外，每一个元素有且只有一个直接后继元素。数据元素之间的关系是一对一的关系

Operation

InitList(\*L): 初始化操作，建立一个空的线性表L

ListEmpty(L): 判断线性表是否为空表，若线性表为空，返回true，否则返回false

ClearList(\*L): 将线性表清空

GetElem(L,i,\*e):将线性表L中的第i个位置的元素值返回给e

LocateElem(L,e): 在线性表L中查找与给定值e相等的元素，如果查找成功，返回该元素在比表中序号表示成功；否则，返回0表示失败

ListInsert(\*L,i,e): 在线性表L中第i个位置插入新元素e

ListDelete(\*L,i,\*e): 删除线性表L中第i个位置元素，并用e返回其值

ListLength(L): 返回线性表L的元素个数

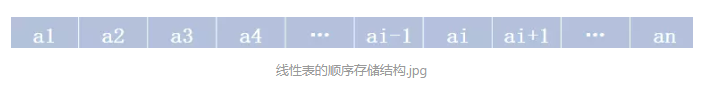
endADT

## 线性表的顺序存储结构

### 顺序存储结构

**1、定义：**用一段地址连续的存储单元依次存储线性表的数据元素。

2、线性表（a1,a2,…,an）的顺序存储如下：



物理上的存储方式事实上就是在内存中找个初始地址，然后通过占位的形式，把一定的内存空间给占了，然后把相同数据类型的数据元素依次放在这块空地中。

3、描述顺序存储结构的是三个属性

1、存储空间的起始位置：数组data，它的存储位置就是存储空间的存储位置。

2、线性表的最大存储容量：数组长度MaxSize。

3、线性表的当前长度：length。

4.数组的长度是存放线性表的存储空间的长度，存储分配后这个量一般是不变的。而线性表的长度是线性表中数据元素的个数，随着线性表插入和删除操作的进行，这个量是变化的。

5.地址计算方法：由于数据是从0开始第一个下标的，因此线性表的第i个元素是要存储在数据下标为i - 1 的位置。存储器中每个存储单元都有自己的编号，这个编号称为地址。

存储器的每个存储单元都有自己的编号，这个编号称为地址。每个数据元素都需要占用一定的存储单元空间的，假设占用的是c个存储单元，对于第i个数据元素ai存储位置为（LOC表示获得存储位置的函数）：

LOC(ai) = LOC(a1) + (i-1)\*c

6.线性表顺序存储结构的优缺点：

优点：无须为表示表中元素之间的逻辑关系而增加额外的存储空间；可以快速地存取表中任意位置的元素。

缺点：插入和删除操作需要移动大量元素；当线性表长度变化较大时，难以确定存储空间的容量；造成存储空间的“碎片”。（实际存储空间小于分配的空间）

7、插入操作

1、如果线性表长度大于等于数组长度，抛出异常

2、如果插入位置不合理，抛出异常

3、从最后一个元素开始向前遍历到第i个位置，分别将它们都向后移动一个4、位置将要插入元素填入位置i

5、表长加1

8、删除操作

1、如果为空表，抛出异常

2、如果删除位置不合理，抛出异常

3、从删除元素位置开始遍历到最后一个元素位置，分别将它们向前移动一个位置

4、表长减1

9、优缺点

线性表的顺序存储结构，在存、读数据时，不管是哪个位置，时间复杂度都是O（1）;而插入或删除时，时间复杂度都是O（n）。

优点：

无需为表示线性表中的逻辑关系而增加额外的存储空间

可以快速的存取线性表中任一位置的元素

缺点：

插入和删除操作需要移动大量的元素

难以确定线性表存储空间的容量

造成存储空间的“碎片”，浪费存储空间

## 线性表的链式存储结构

### 单项链表的创建

创建单链表的过程就是一个动态生成链表的过程。即从“空表”的初始状态起,依次建立各元素结点,并逐个插入链表。