算法复杂度

1.判断算法复杂度的方法

1.1空间复杂度

算法所占据的存储空间

**实例说明**

以打印1-N为例

**（1）递归算法**

Print(N)首先打印N，但并不直接输出，而是调用Print(N-1)，Print(N-1)又调用Print(N-2)……程序等待在原地，这样一来，PrintN到N-1各自占据一个单元位置，加起来占据的空间非常大。

**（2）循环算法**

每次打印一个数据，打印完后空间就释放，占据空间很小

1.2时间复杂度

执行算法所耗费的时间

**1.2.1实例说明**

以计算多项式值为例，计算。乘法所消耗的时间比加法多得多，以乘法次数计量时间复杂度

（1）逐项计算sum=sum+a[i]\*math.pow(x,i)

乘法次数为1+2+…+n=

耗费时间为C\*

时间复杂度是耗费时间的极限，n很大时一次项忽略不计因此时间复杂度为

（2）秦九绍算法，算法表达式p=p\*x+a[i-1]

乘法次数n

耗费时间为*C\*n*

时间复杂度为*O(n)*

1.2.2应关注的时间复杂度

最坏情况复杂度

**2.复杂度的表示**

2.1.复杂度*T(n)*的界限

-上界

*T(n)=*-下界

-上下界相同

计算复杂度就是寻找复杂度计算式中对该界限起决定作用的元素/项

2.2常见的复杂度形式

按复杂程度排序：

看到复杂度为时就应想到尽可能将复杂度降为*nlogn*

2.2复杂度分析方法

（1）两段算法复杂度分别为O(m)和O(n)，如果把两段算法拼在一起，则复杂度为O(m)+O(n)，如果把两段算法嵌套，则复杂度为O(m)\*O(n)。

（2）计算for循环复杂度。用循环次数乘以循环体类程序的复杂度。

（3）计算if else复杂度。复杂度为各分支复杂度中最大者。

（4），起决定作用，复杂度为。