绪论

▼ 基本概念

数据结构三要素：逻辑结构、存储结构、数据的运算

数据：符号的总称

数据元素：数据的基本单位

数据项："原子项"数据项是不可分割的最小单位

关键字：识别一个/多个数据项

数据对象：性质相同的数据元素的集合

数据结构：存在特定关系的数据元素的集合

四种基本结构：

集合结构：属于/不属于同一种类型

线性结构：一对一

树形结构：一对多

图状结构或网状结构：多对多

数据结构

逻辑结构：是个二元组：Data_Structures=(D,S)

D是数据元素的有限集，S是D上关系的有限集

物理结构（存储结构）：

存储结构是逻辑结构在存储器中的映象

包括数据元素的表示和关系的表示

关系的映象方法：

顺序映象→顺序存储结构：存储位置相邻

链式映象→链式存储结构：指针

抽象数据类型（ADT）

数据抽象、数据封装

ADT抽象数据类型名{

数据对象：数据对象的定义

数据关系：数据关系的定义

基本操作：基本操作的定义

}ADT抽象数据类型名

▼ 算法

特性

有穷性、确定性、可行性、输入、输出

评价标准

正确性、可读性、健壮性、效率与存储量需求

时间复杂度、空间复杂度

$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3)$

$O(2^n) < O(n!) < O(n^n)$

计算时间复杂度

▼ 单层循环

- **注意死循环的情况** $O(\infty)$

1. 列出循环趟数 t 与每轮循环**条件变量** i 的变化值
2. 找出 t 与 i 的关系
3. 利用结束条件解出 t 与 i 的数量级关系。

1．下列函数的时间复杂度为（ B ）。

```
int A( int n ) {
    int i=0, sum = 0;
    while (sum < n)    sum += ++i;
    return i; }
```

++i;
sum+=i;

A．$O(\log_2 n)$　　B．$O(n^{1/2})$　　C．$O(n)$　　D．$O(n\log_2 n)$

找关系 趟数t　0　1　2　3　4　…
sum　0　1　1+2　1+2+3　1+2+3+4　…
$sum = 1+2+\dots+t = \frac{(1+t)t}{2}$

结束条件　$sum = \frac{(1+t)t}{2} = n$

解数量级　$t^2 \sim n$
$\Rightarrow t \sim \sqrt{n}$
$\Rightarrow O(n^{\frac{1}{2}})$

① i=n*n;
while(i!=1)
　i=i/2;

找关系 趟数t　0　1　2　…
i　n^2　$\frac{n^2}{2}$　$\frac{n^2}{2^2}$　…
$i = \frac{n^2}{2^t}$

结束条件：　$i = \frac{n^2}{2^t} = 1$

$\Rightarrow 2^t = n^2$
$\Rightarrow t = 2\log_2 n$
$\Rightarrow O(\log_2 n)$

② x=0;
while(n>=(x+1)*(x+1))
　x=x+1;

趟数t　0　1　2　…
x　0　1　2　…
$x = t$

$(t+1)^2 > n$
$\Rightarrow t \sim \sqrt{n}$
$\Rightarrow O(n^{\frac{1}{2}})$

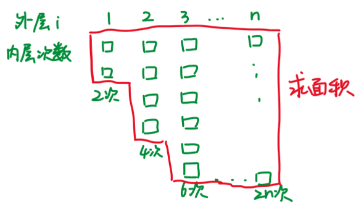
③ for(i=0; i<n; i*=3)
i 恒等于0
$O(\infty)$

▼ 两层循环

二维，类似求面积。外层为宽度，内层为长度。

- 若外层比较复杂，可使用计算单层循环的方法算出宽度（如例2）

1. 列出外层循环中 i 的变化值（宽度）
2. 列出内层语句的执行次数
3. 求和（求面积）

①
```
m=0;
for(i=1;i<=n;i++)
  for(j=1;j<=2*i;j++)
    m++;
```


求和：$2+4+6+\dots+2n=\frac{(2+2n)n}{2}=n(n+1)$

$\Rightarrow O(n^2)$

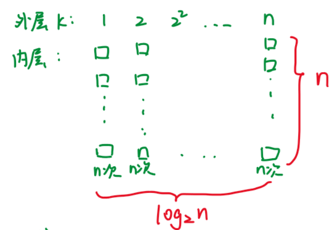
②
```
count=0;
for(k=1;k<=n;k*=2)
  for(j=1;j<=n;j++)
    count++;
```
使用解单层的方法解出宽度

外层：循环数 t 0 1 2 ...

k 1 2 2^2 ...

$k=2^t=n$

$\Rightarrow t=\log_2 n$（宽度）



求和：$n\times\log_2 n$

$\Rightarrow O(n\log_2 n)$

③
```
for(i=n-1;i>1;i--)
  for(j=1;j<i;j++)
    if(A[j]>A[j+1])
      ...
```


求和：$1+2+\dots+n-2=\frac{(n-1)(n-2)}{2}$

$\Rightarrow O(n^2)$