



第2章 绪论

作业选讲

数据之法
结构之美
算法之道



zhuyungang@jlu.edu.cn



习题1-1

- 以学生的成绩表为例，按照逻辑结构、存储结构和相关操作三个方面，讨论它的数据结构。
- 逻辑结构：线性结构
- 存储结构：顺序存储、链接存储、索引存储
- 相关操作：插入、删除、修改、排序、查找

习题1-6

分析下面程序段的时间复杂性。

```
int s = 0, i, j, k;  
for( i = 0; i<=n; i++)  
    for( j = 0; j<=i; j++)  
        for( k = 0; k<j; k++)  
            s++;
```

$$\sum_{i=0}^n \sum_{j=0}^i \sum_{k=0}^{j-1} 1$$

习题1-6

$$\begin{aligned}\sum_{i=0}^n \sum_{j=0}^i \sum_{k=0}^{j-1} 1 &= \sum_{i=0}^n \sum_{j=0}^i j = \sum_{i=0}^n \frac{i(i+1)}{2} \\&= \frac{1}{2} \sum_{i=0}^n (i^2 + i) = \frac{1}{2} \left(\sum_{i=0}^n i^2 + \sum_{i=0}^n i \right) \\&= \frac{1}{2} \left[\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \right]\end{aligned}$$

$$= \frac{n(n+1)(2n+1)}{12} + \frac{n(n+1)}{4}$$

$$= \frac{n(n+1)(2n+1)}{12} + \frac{3n(n+1)}{12}$$

$$= \frac{n(n+1)(2n+4)}{12}$$

$$= \frac{n(n+1)(n+2)}{6}$$

$O(n^3)$

```
int count=0;  
for(int i=0; i<n; i++)  
    for(int j=i; j<n; j++)  
        count++;
```

$$\begin{aligned} & \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} 1 \\ &= \sum_{i=0}^{n-1} (n - i) \\ &= \sum_{k=1}^n k \\ &= \frac{n(n+1)}{2} = O(n^2) \end{aligned}$$



习题1-9

- 将下列算法的时间复杂性 $O(n)$, $O(2^n)$, $O(\log_2 n)$, $O(n \log_2 n)$, $O(n^5)$, $O(n^2+1)$, $O(n^3-n^2)$, 按由低到高的顺序排列, 其中, n 是数据的输入规模。
- $O(\log_2 n)$, $O(n)$, $O(n \log_2 n)$, $O(n^2+1)$, $O(n^3-n^2)$, $O(n^5)$, $O(2^n)$

习题1-5

- 编写一个算法，判断任一整数 n 是否为素数。
- 素数：指在一个大于1的自然数中，除了1和此整数自身外，不能被其他自然数整除的数。

```
bool IsPrime(int n){  
    if (n<=1) return false ;  
    for( int i=2; i<n; i++)  
        if (n % i==0) return false;  
    return true;  
}
```


改进一

- 一个数不可能被“比自己的一半还大的数”整除。
- 一个数除以“比它的一半还大的数”，一定除不尽。

例如 $n=10$,

2 3 4 5 6 7 8 9

- 如果 $2\dots n-1$ 中的数 i 能整除 n ,即 n/i 等于某个正整数 k ,由于 $i < n$,所以 k 不可能等于1,故 $k \geq 2$.
- $n/i \geq 2$, 所以 $i \leq \lfloor n/2 \rfloor$
- 对于 $2\dots n-1$, 我们只需测试前 $n/2$ 个数, $n/2$ 以后的数肯定不能整除 n , 无需测试。



改进一

```
bool IsPrime(int n){  
    if (n<=1) return false ;  
    for( int i=2; i<=n/2; i++)  
        if (n % i==0) return false;  
    return true;  
}
```

改进二

- 如果 n 不是素数，它有约数， $n = a \times b$ ，其中 a 和 b 为正整数
- 则 a 、 b 中必有一个小于或等于 \sqrt{n} ，一个大于或等于 \sqrt{n} 。

2 a \sqrt{n} b $n-1$

- 如果 n 不是素数，它必然有一个约数 $\leq \sqrt{n}$

```
bool IsPrime(int n) {  
    if(n<=1) return false ;  
    int t=int(sqrt(n));  
    for(int i=2; i<=t; i++)  
        if(n % i==0) return false;  
    return true;  
}
```



改进三

除2之外的偶数都不是素数。

```
bool IsPrime(int n){  
    if(n==2) return true ;  
    if(n<=1 || n % 2==0) return false ;  
    int t=int(sqrt(n));  
    for(int i=3; i<=t; i+=2)  
        if(n % i==0) return false;  
    return true;  
}
```