

数据结构



- 任课老师：黄煜廉
- 邮箱：e_win2003@163.com

课程要求



- 课堂讲授：提问、讨论、小测
- 实验课程：20个作业习题+3个大实验
- 课程成绩：实验(30%)+期末考试(70%)

参考书



- 计算机程序设计艺术(Art of Computer Programming)
Donald E.Knuth著 译者:苏运霖 出版社:国防工业出版社
- 数据结构C++语言描述 William Ford 著清华大学出版社
- 数据结构 严尉敏,吴伟民 著 清华大学出版社

问题思考

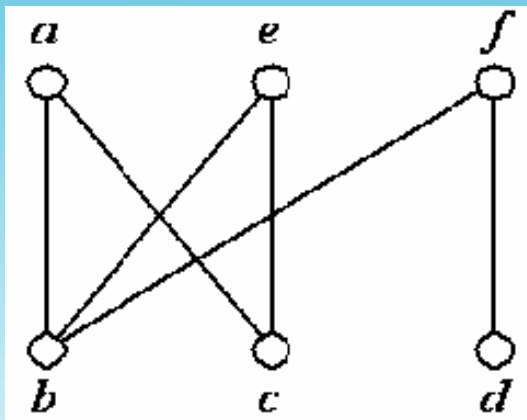


问题1: 六名间谍被擒，已知懂汉语、法语和日语，懂德语、俄语和日语，懂英语和法语，懂西班牙语，懂英语和德语，懂俄语和西班牙语，问至少用几个房间监禁他们，能使在一个房间里的人不能直接对话。

数学模型的建立



- 解：以六人为顶点，懂共同语言，则在顶点间连边得下图。因为图中没有长度为奇数的回路，所以是二部图，故至少应有两间房间即可。



- 问题2：田径赛的时间安排问题



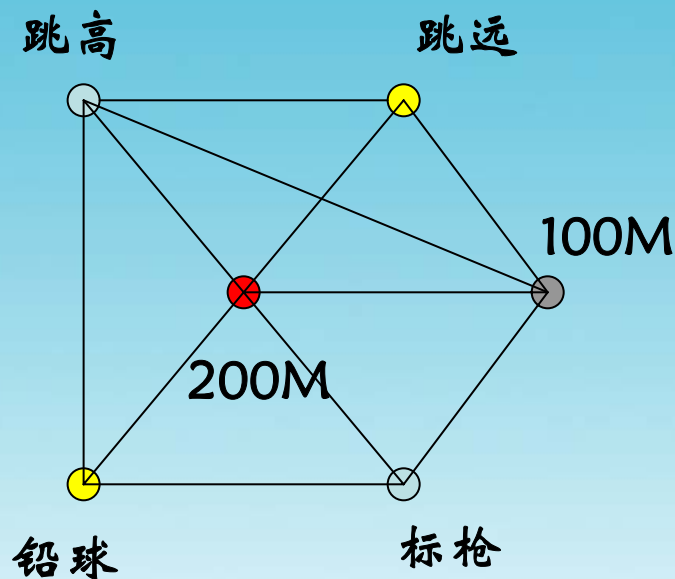
姓名	项目1	项目2	项目3
丁1	跳高	跳远	100M
马2	标枪	铅球	
张3	标枪	100M	200M
李4	铅球	200M	跳高
王5	跳远	200M	

数学模型的建立



解：该问题归结为图顶点的上色问题，因此根据题目给定的已知条件建立起**图模型**。

姓名	项目1	项目2	项目3
丁1	跳高	跳远	100M
马2	标枪	铅球	
张3	标枪	100M	200M
李4	铅球	200M	跳高
王5	跳远	200M	



问题3:

7位客人入席, A只会讲英语, B会讲英, 汉语, C会讲英, 意大利, 俄语, D会讲日, 汉语, E会讲德, 意大利语, F会讲法, 日, 俄语, G会讲法, 德语. 能否安排圆桌席位使每位客人与其左右邻不用翻译便可交谈?



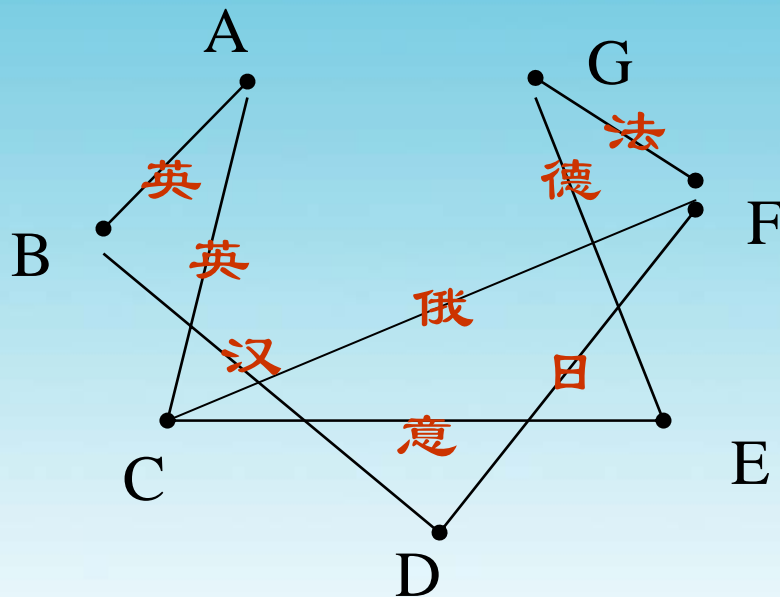
数学模型的建立



解:首先建立无向图模型:

结点为客人;会共同语言的2结点相邻接. 则问题归结为求此图的一条Hamilton回路.

此图存在一条Hamilton回路是: (A, B, D, F, G, E, C, A).
按此回路安排席位可满足要求.



总结



- 首先建立数学模型；
- 接着思考如何设计算法实现该数学模型。
- 疑问：
 - (1) 如何建立数学模型？可以建立什么方面的数学模型？
 - (2) 如何设计算法呢？要考虑什么方面的问题呢？

有问
题啦？

逻辑结构——数学模型



- 数据之间的相互关系称为逻辑结构。通常分为四类基本结构：
- **1.集合** 结构中的数据元素除了同属于一种类型外，别无其它关系。
- **2.线性结构** 结构中的数据元素之间存在一对一的关系。
- **3.树型结构** 结构中的数据元素之间存在一对多的关系。
- **4.图状结构或网状结构** 结构中的数据元素之间存在多对多的关系。
- 逻辑结构的描述方法：数学方法（关系和关系图）

线性结构



结论：线性结构中各数据成员之间的线性关系：有直接前驱和直接后继(除最前、最后一个元素)

例：电话号码查询问题

方法1：顺序存储，顺序查找

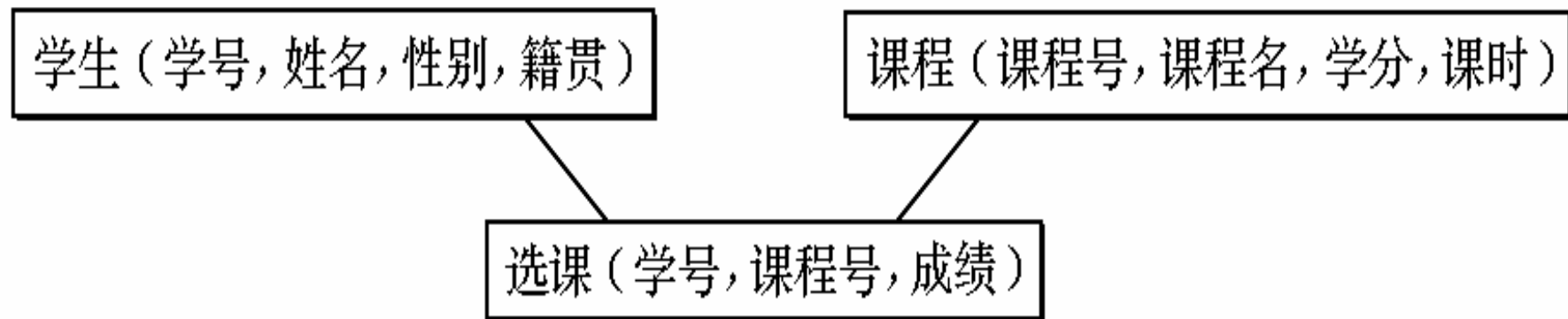
非线性结构



选课单包含如下信息

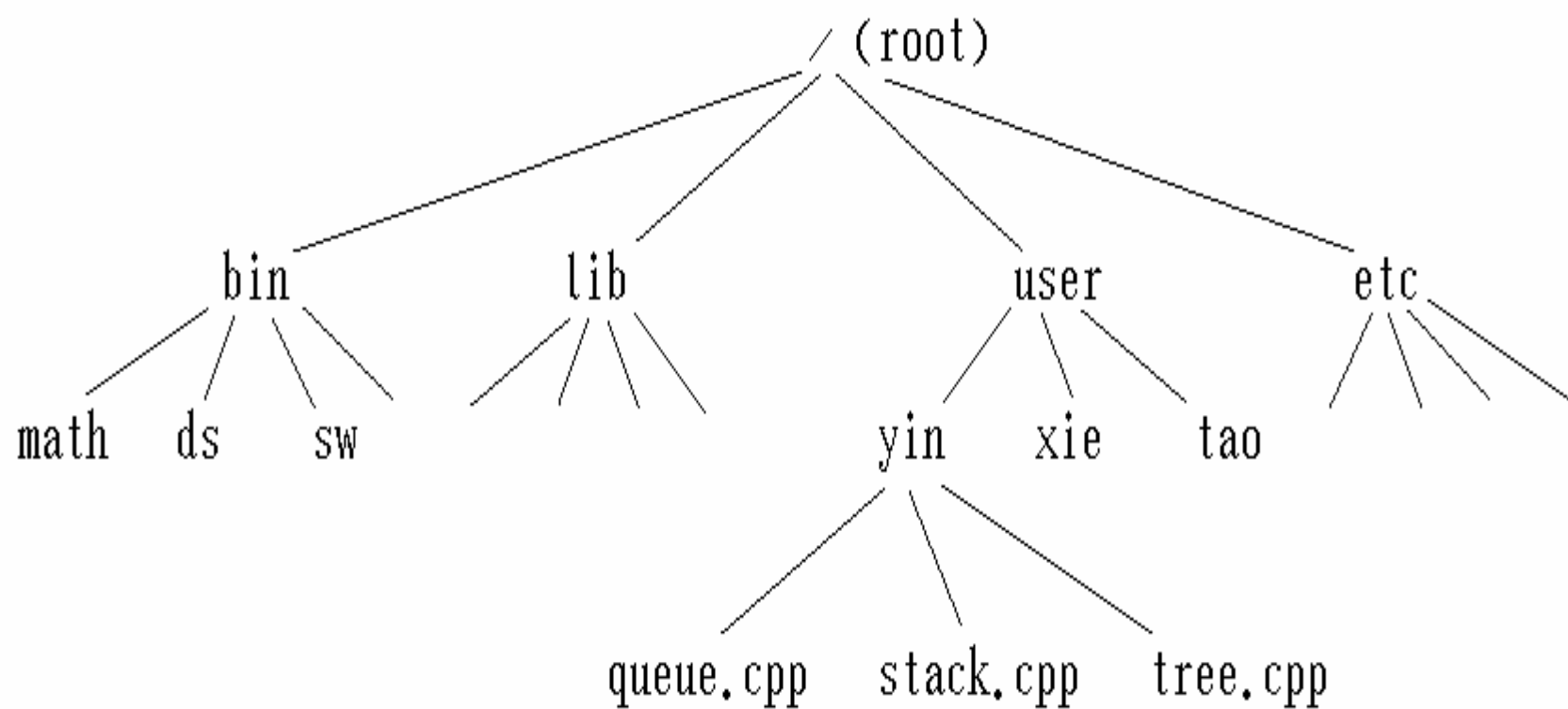
学 号 课程编号 成 绩 时 间

学生选课系统中实体构成的网状关系

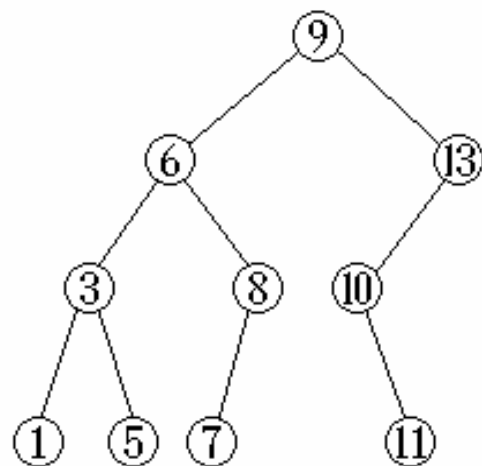
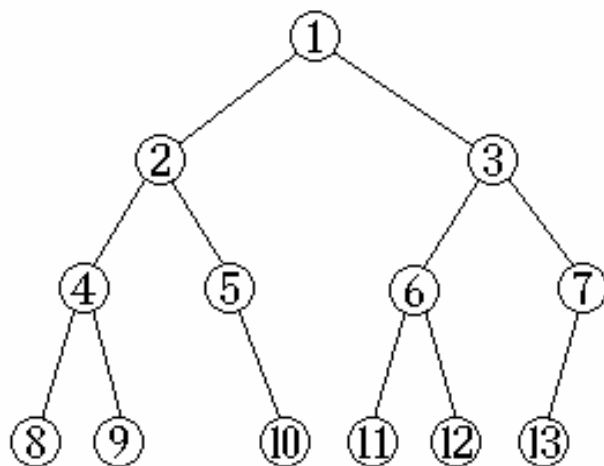
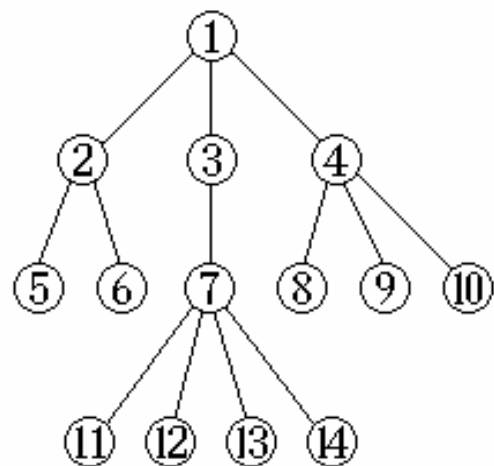


结论：非线性结构中各数据成员之间的没有线性关系：
前驱和后继可能多于一个

UNIX文件系统的系统结构图



树形结构

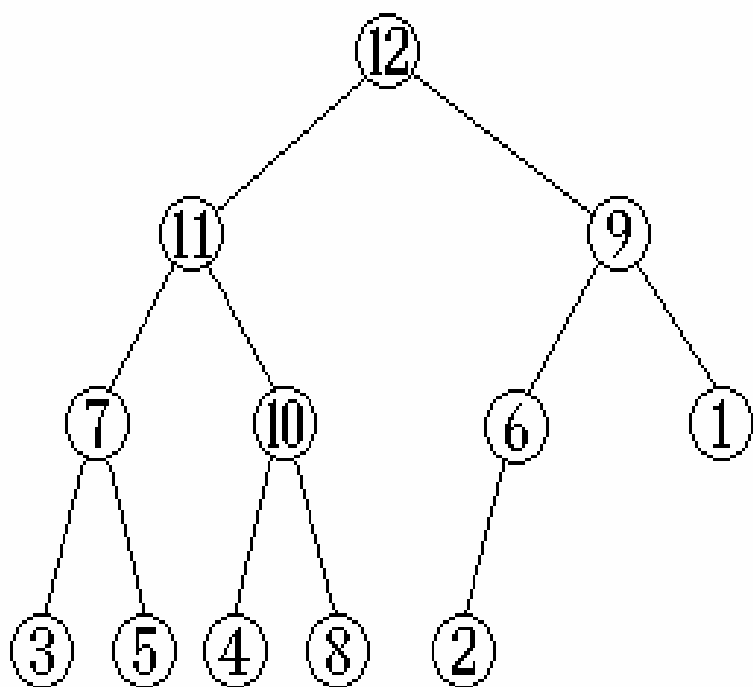


树

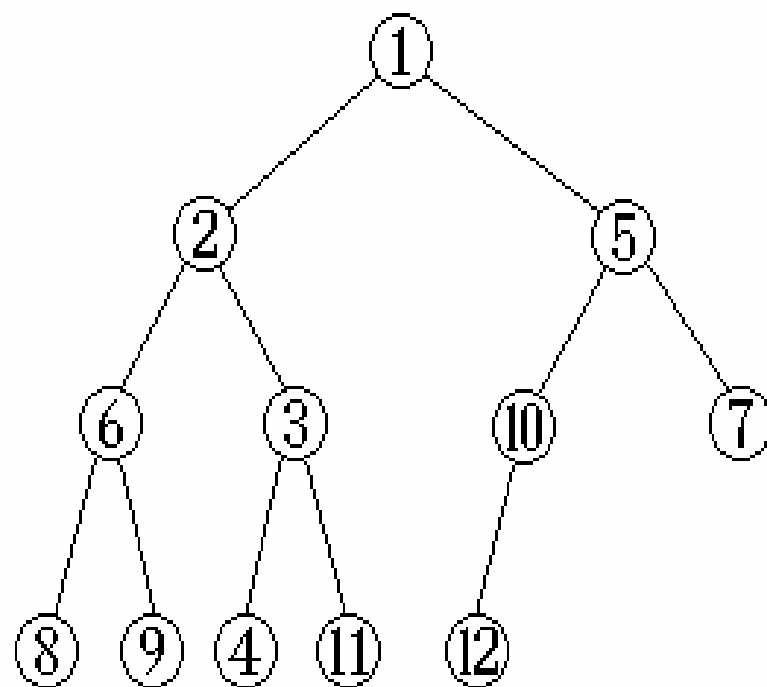
二叉树

二叉搜索树

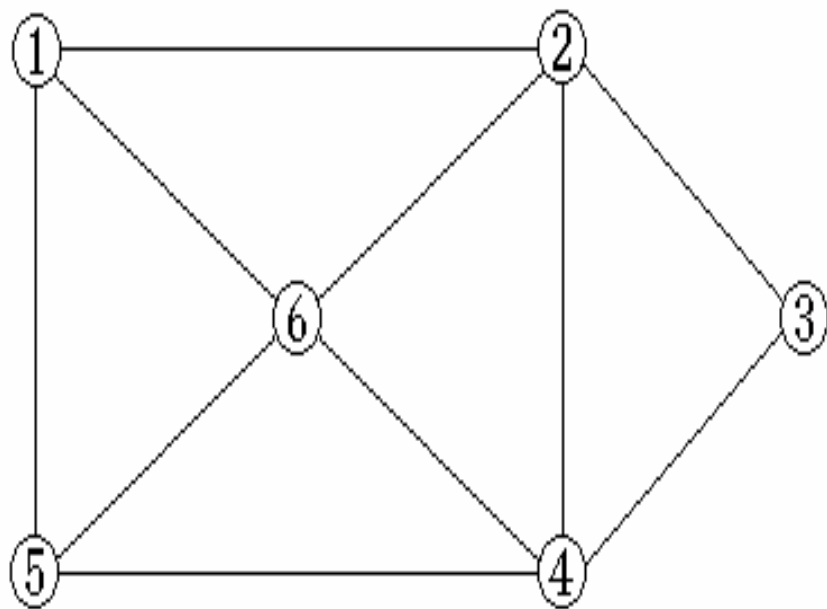
堆结构



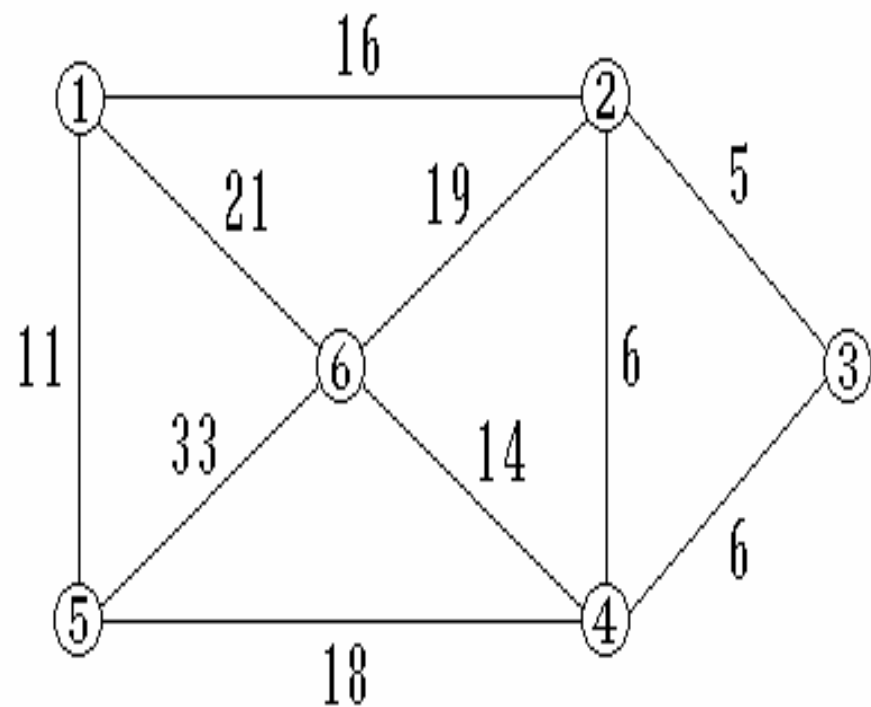
大根堆



小根堆



图结构



网络结构

存储结构



- 数据结构在计算机中的表示称为数据的物理结构，又称为存储结构。
- 存储结构的种类：
 - 顺序存储
 - 链式存储
 - 索引存储
 - 散列存储

1.2 算法和算法分析



- **算法**：是对特定问题求解步骤的一种描述。

算法是指令的有限序列，其中每一条指令表示一个或多个操作。

算法具有以下五个特性：

- (1) 有穷性
- (2) 确定性
- (3) 可行性
- (4) 输入
- (5) 输出



算法的描述方法



- 自然语言
- 程序设计语言
- 类语言

问题

求前n项的和值： $1+2+\dots+n=?$



二分程序清单sum-bin.cpp



```
#include<iostream>
using namespace std;
int f(int low,int high);
```

```
void main()
```

```
{
    int n;
    cin>>n;
    cout<<f(1,n);
}
```

```
int f(int low,int high)
```

```
{
    int mid;
    if (low>high) return 0;
    else
    {   mid=(low+high)/2;
        return mid+f(low,mid-1)+f(mid+1,high);
    }
}
```

算法的评价标准



- 评价一个好的算法有以下几个标准：
- (1) **正确性 (Correctness)** 算法应满足具体问题的需求。
- (2) **可读性 (Readability)** 算法应该好读。以有利于阅读者对程序的理解。
- (3) **健壮性 (Robustness)** 算法应具有容错处理。当输入非法数据时，算法应对其作出反应，而不是产生莫名其妙的输出结果。
- (4) **时间 (时间复杂度)**
- (5) **空间 (空间复杂度)**

时间复杂度



1. $x=a$; $a=b$; $b=x$;

2. $s=0$;

for ($i=0$; $i<n$; $i++$)

$s=s+1$;

3. $s=0$;

for ($i=0$; $i<n$; $i++$)

for ($j=i+1$; $j\leq n$; $j++$)

$s=s+1$;

4. for ($i=0$; $i<n$; $i++$)

if ($A[i]==x$) break;

时间复杂度



例：求两个方阵的乘积 $C = A * B$

#define n 自然数

```
MATRIXMLT(float A[n][n],float B[n][n],float C[n][n])
{
    int i,j,k;
    for(i=0;i<n;i++)                //n+1
        for(j=0;j<n;j++)            //n(n+1)
        {
            C[i][j]=0;                //n*n
            for( k=0;k<n;k++)        //n*n*(n+1)
                C[i][j]=A[i][k]*B[k][j] //n*n*n
        }
}
```

时间复杂度

练习题:

1. for ($i=1; i \leq n; i++$) $x = x+1$;

2. for ($i=1; i \leq n; i++$)
 for ($j=1; j \leq n; j++$) $x = x+1$;

3. for ($i=1; i \leq n; i++$)
 for ($j=1; j \leq n; j++$)
 { $s=0$;
 for ($k=1; k \leq n; k++$)
 $s = s + a[i][k] * b[k][j]$;
 $c[i][j] = s$;
 }



时间复杂度

练习题：

```
4.i=1;  
  while (i<=n )  
    i=i*2;
```



常见的时间复杂度，按数量级递增排序：



常数阶

$$O(1)$$

对数阶

$$O(\log_2 n)$$

线性阶

$$O(n)$$

线性对数阶

$$O(n \log_2 n)$$

平方阶

$$O(n^2)$$

立方阶

$$O(n^3)$$

.....

K次方阶

$$O(n^k)$$

指数阶

$$O(2^n)$$

递归的时间复杂度的计算



```
int f(int n)
{
    if (n==1) return 1;
    else return n*f(n-1);
}
```

空间复杂度



- $x=a; a=b; b=x;$

空间的复杂性



问题：数组逆置

方法一：

```
for (i=1; i<=n; i++) b[n-i+1]= a[i];  
for (i=1; i<=n; i++) a[i]= b[i];
```

方法二：

```
for (i=1; i<=n/2; i++)  
{ w= a[i];  
  a[i]= a[n-i+1];  
  a[n-i+1]= w;  
}
```


算法描述时的考虑



- 首先算法必须是正确的。
- 此外还需考虑：
 - 1、算法易于理解，易于编程（在计算机上实现），易于调试。
 - 2、要求算法高效，节省时间与空间。
- 一个算法运行所需时间的长短、空间的大小具有非常重要的意义。
- 时间和空间相互之间有一种制约关系，何者为重需视具体情况而定。

常见算法设计技术



- 分治法
- 贪心法
- 动态规划法——最优化原理技术
- 遍历/检索技术
- 回溯法

• 基本概念和术语



- **数据(Data):**是对信息的一种符号表示。在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。
- **数据元素(Data Element):**是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。
- **数据项:**一个数据元素可由若干个数据项组成。数据项是数据的不可分割的最小单位。
- **数据元素集合**

数据结构



- 什么是数据结构

1.逻辑结构是一个二元组:

$$\text{Data-Structure} = (D, S)$$

其中: D 是数据元素的有限集, S 是 D 上关系的有限集。

2.存储结构

3.算法

- 利用计算机来解决实际问题的步骤?

建模能力测试方法



- ACM测试
- acm.zju.edu.cn
- acm.pku.edu.cn

算法设计练习



- 试写一个算法，实现将数组 $a[n]$ 中的 n 个整数循环后移一个位置，并给出它的时间复杂度和辅助空间的大小。