

# 数据结构

# 课程要求

- 课堂讲授：提问、讨论、小测
- 作业：算法设计
- 成绩：平时成绩(30%)+期末考试(70%)

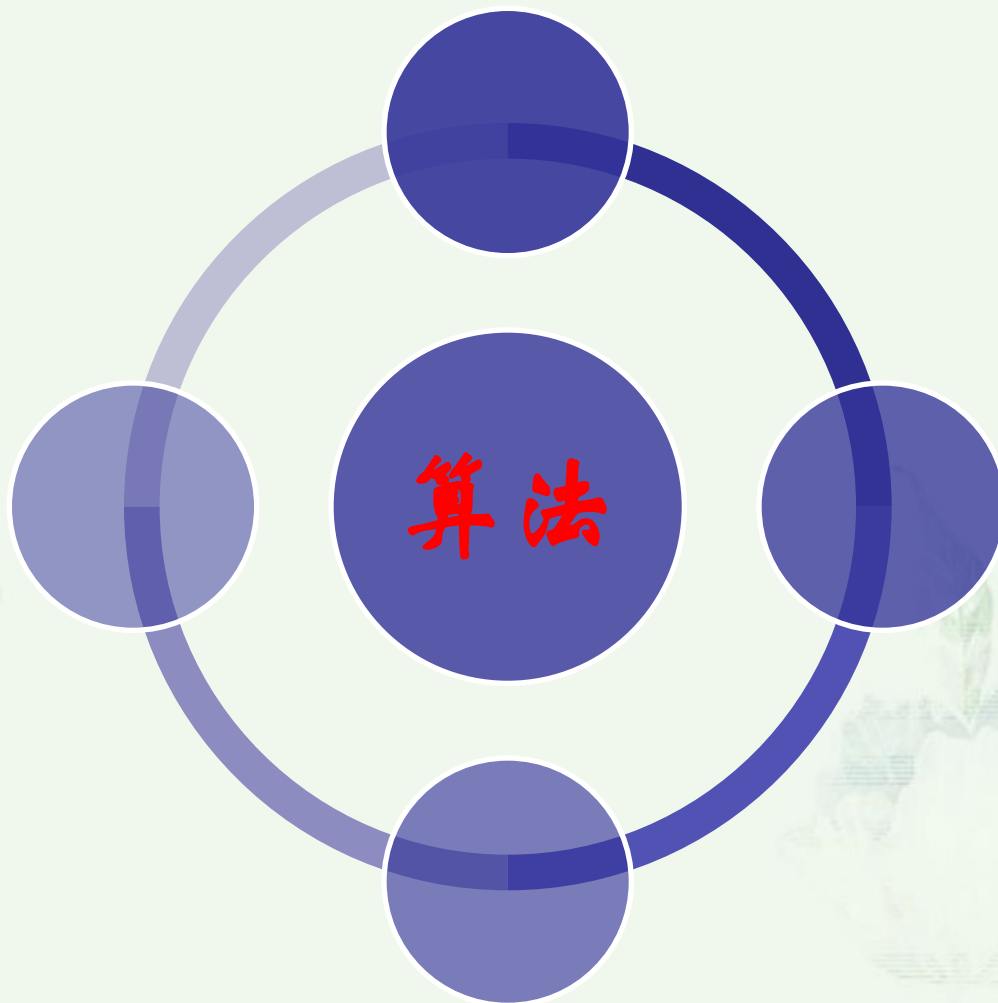
# 需要提前熟练掌握的知识

- 面向对象程序设计思想
  - 封装
  - 继承
  - 多态
- 模板的设计方法
- 文件流的操作方法
- 程序的调试方法与步骤

# 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

# 计算机科学的核心



# 思考

- 现要求将一个学校的学生们的某科成绩进行等级评定。  
输入一个整数分数，如果大于等于90，输出“优”，如果大于等于80，小于90，输出“良”，如果大于等于70，小于80，输出“中”，如果大于等于60，小于70，输出“及格”，如果小于60，输出“不及格”。

```
#include <iostream>
using namespace std;
void main( )
{ char name[20];  int score;
  while(1) {
    cout<< "请输入学生姓名和成绩:";
    cin>>name>>score;
    if(score==-1) break;// 输入分数为-1则结束程序执行
    cout<<name<<" "<<score<<" ";
    if (score >= 90)  cout << "优" << endl;
    else if((score>=80)&&(score<90))cout<< "良" << endl;
    else if((score>=70)&&(score<80))cout<<"中" << endl;
    else if((score>=60)&&(score<70))cout<<"及格" <<endl;
    else if( score < 60) cout << "不及格 " <<endl;
  }
}
```

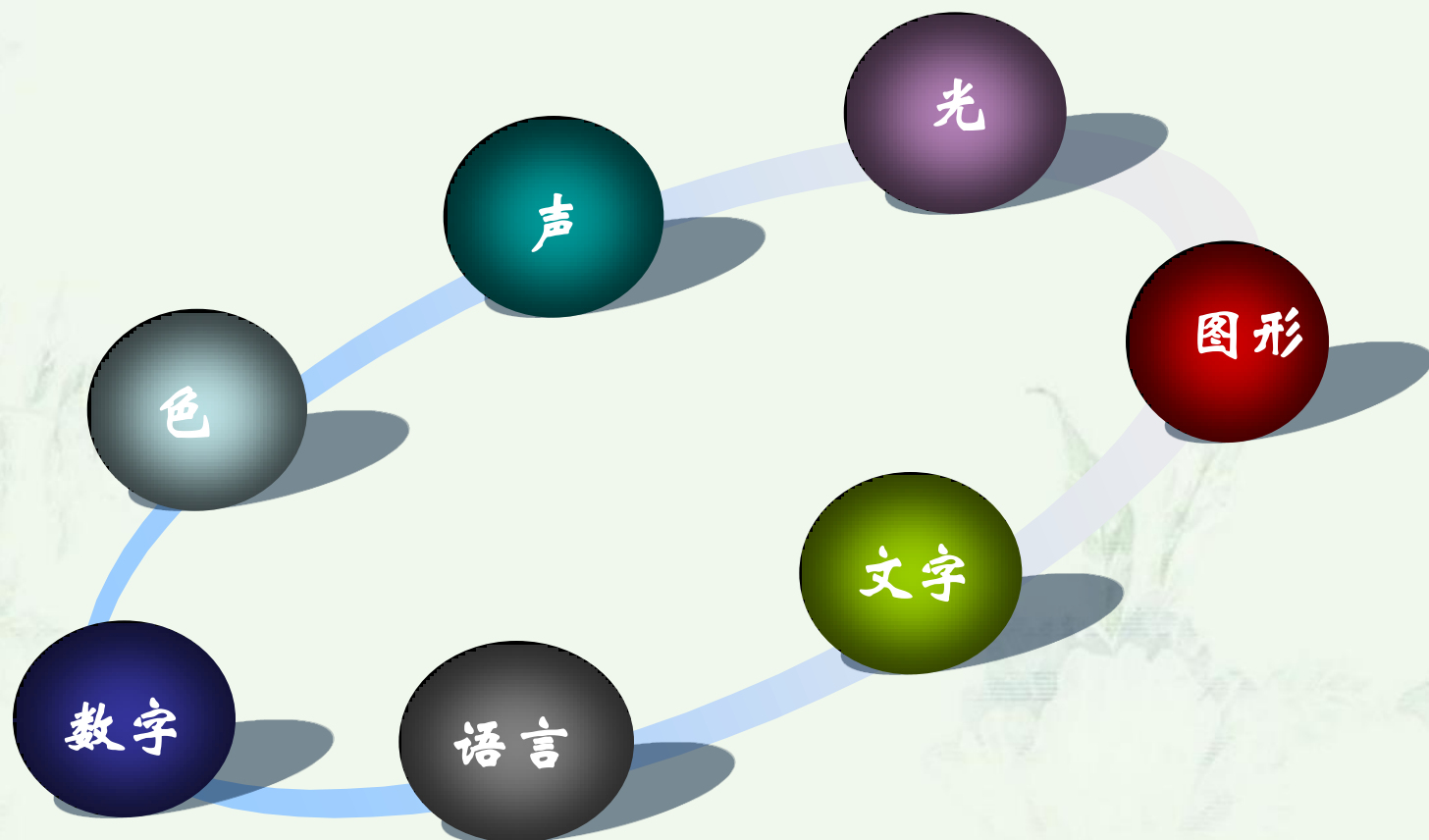
```
#include <iostream>
using namespace std;
void main( )
{ char name[20];  int score;
  char grade[6][20]={"不及格","及格","中等","良好","优秀","优秀"};
  while(1) {
    cout<<"请输入学生姓名和成绩:";
    cin>>name>>score;
    if(score==-1) break;// 输入分数为-1则结束程序执行
    cout<<name<<" "<<score<<" ";
    int res=score/10-5;
    if (res>0) cout << grade[res] << endl;
    else cout << grade[0]<<endl;
  }
}
```





数据？

生活上...





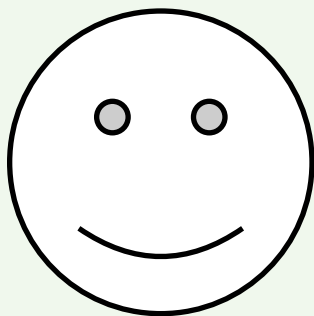


描述事物

生活中：自然语言

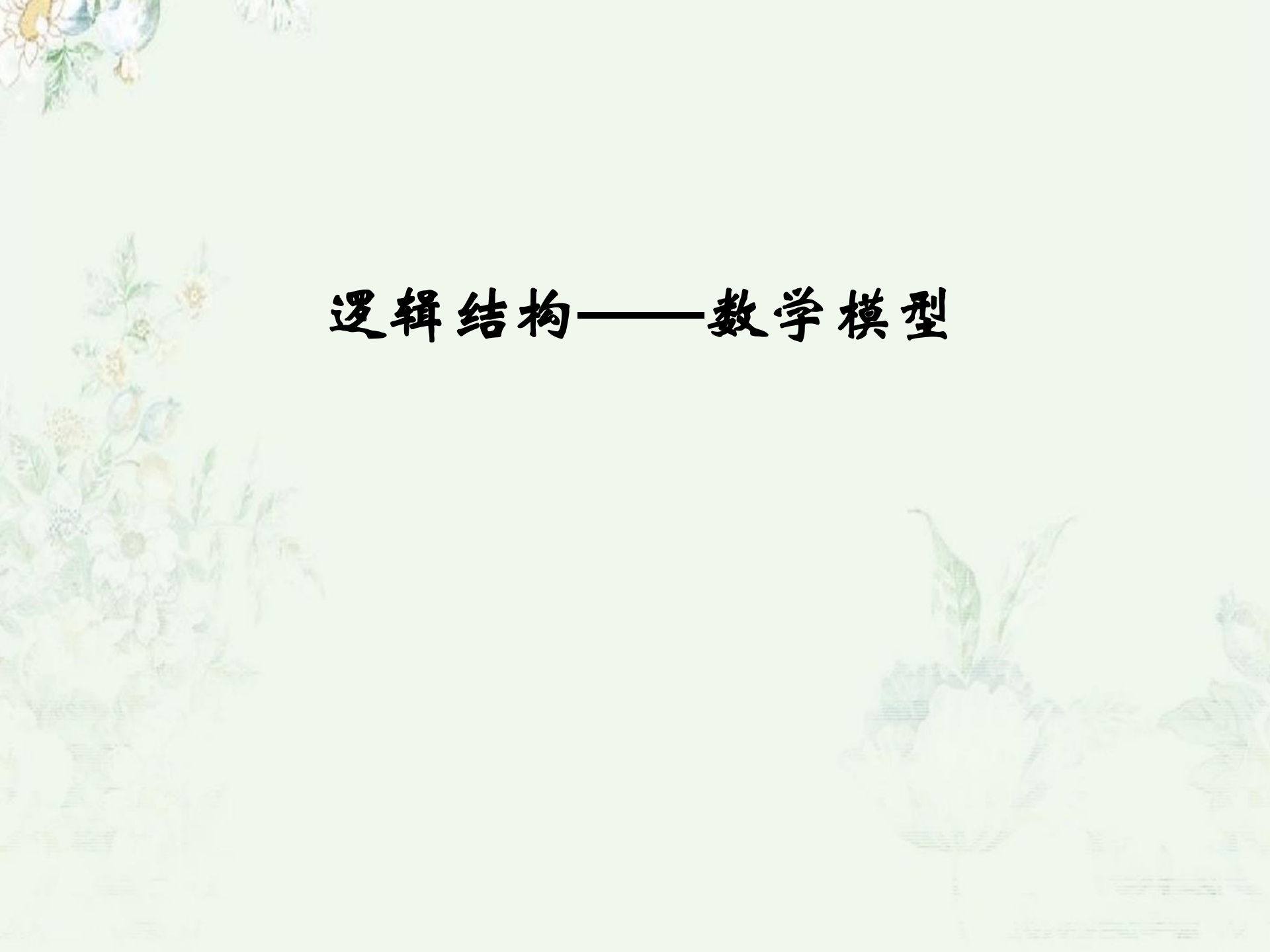
计算机：特征记录

例，一幅黑白图象



文字表达——笑脸

数据——黑白点阵

The slide features a light green background with decorative floral patterns in the corners. The top-left corner has a cluster of colorful flowers in shades of blue, yellow, and white. The bottom-left corner shows a large, detailed white flower with green leaves. The bottom-right corner has a smaller white flower with green leaves. The title is centered in the upper half of the slide.

# 逻辑结构——数学模型

# 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)

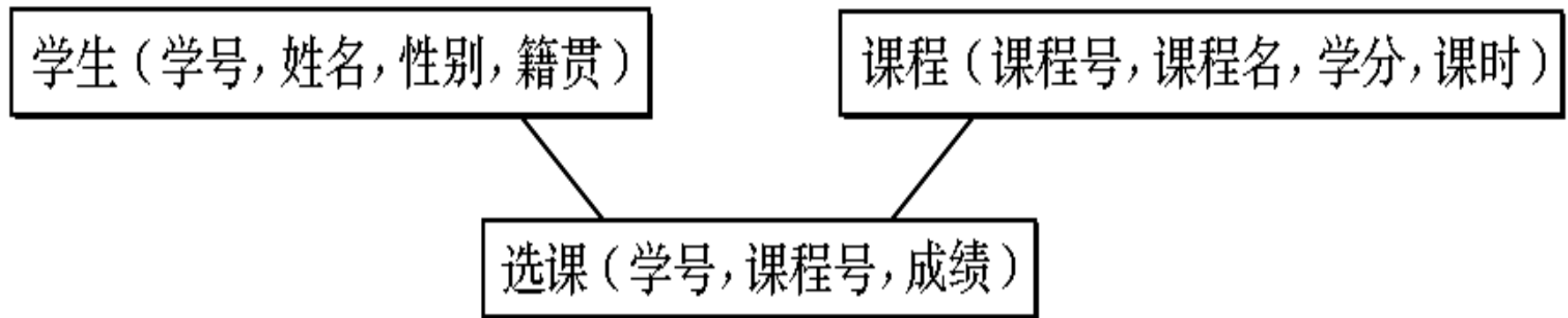


# 非线性结构

选课单包含如下信息

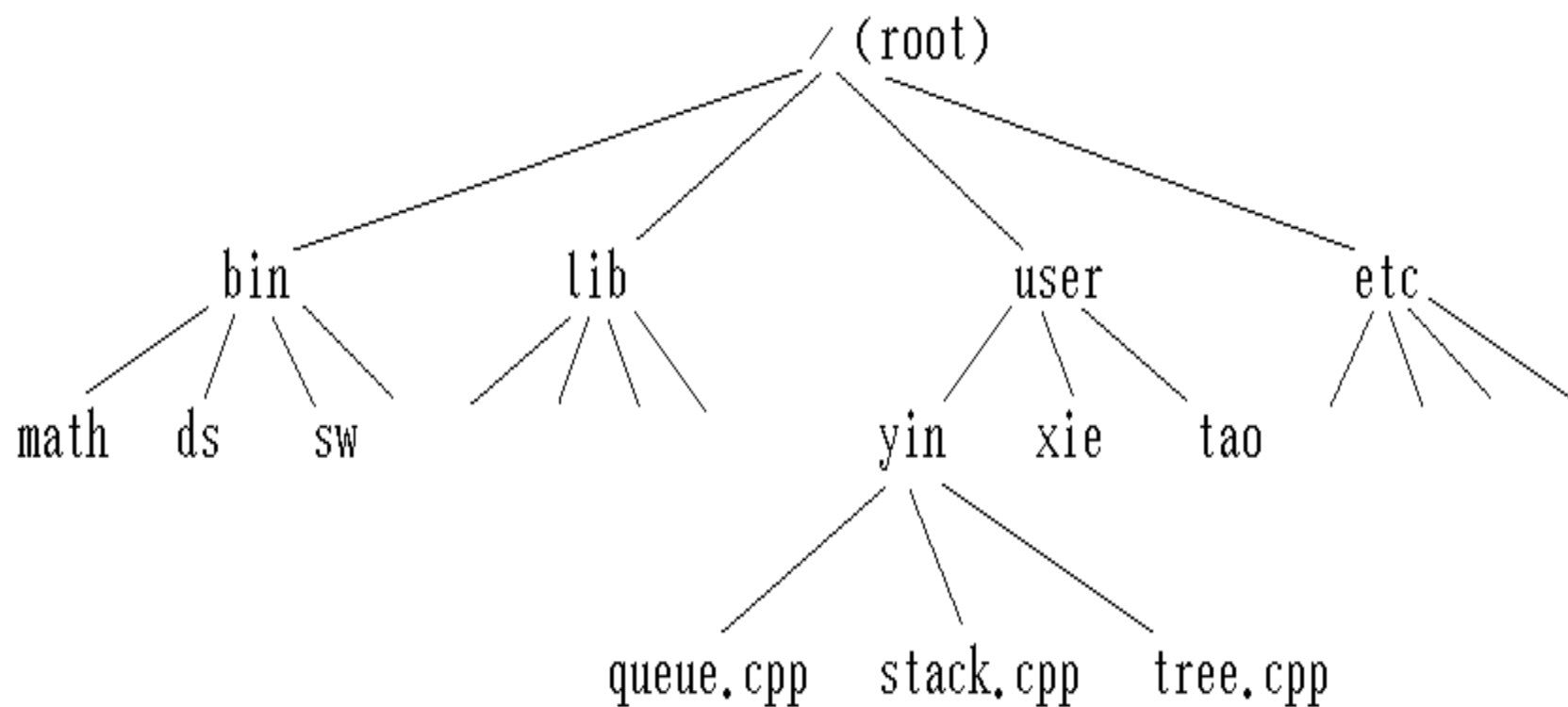
学 号    课程编号    成 绩    时 间

学生选课系统中实体构成的网状关系

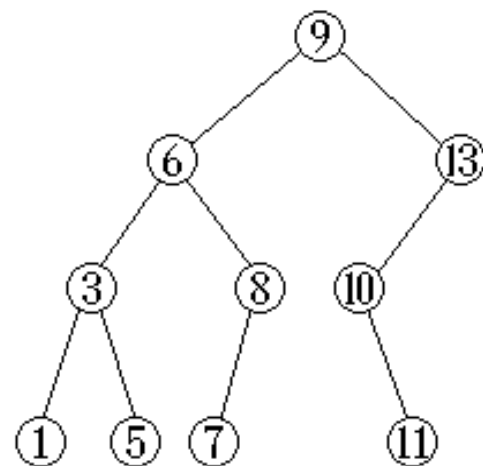
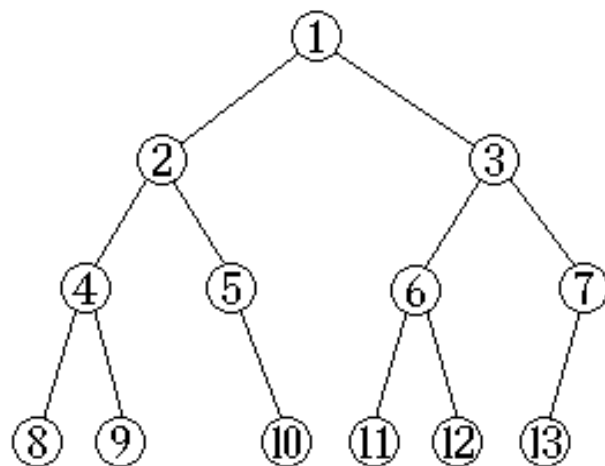
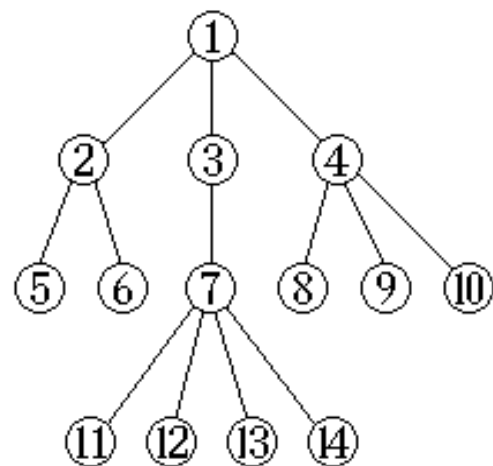


结论：非线性结构中各数据成员之间的没有线性关系：  
前驱和后继可能多于一个

# UNIX文件系统的系统结构图



# 树形结构

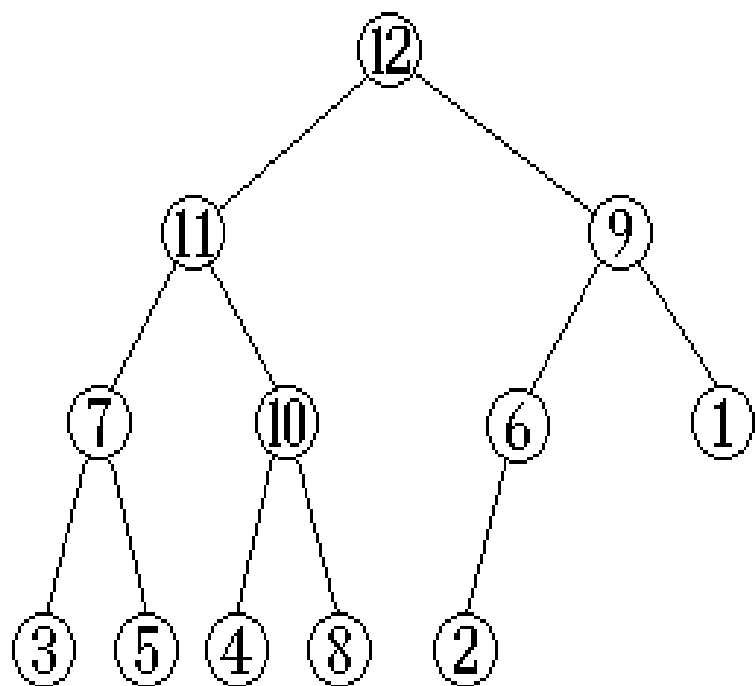


树

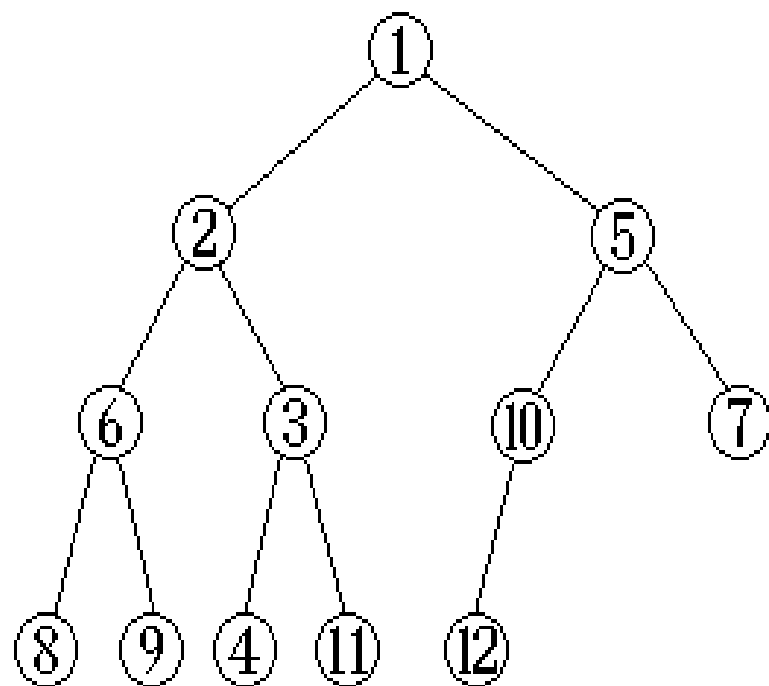
二叉树

二叉搜索树

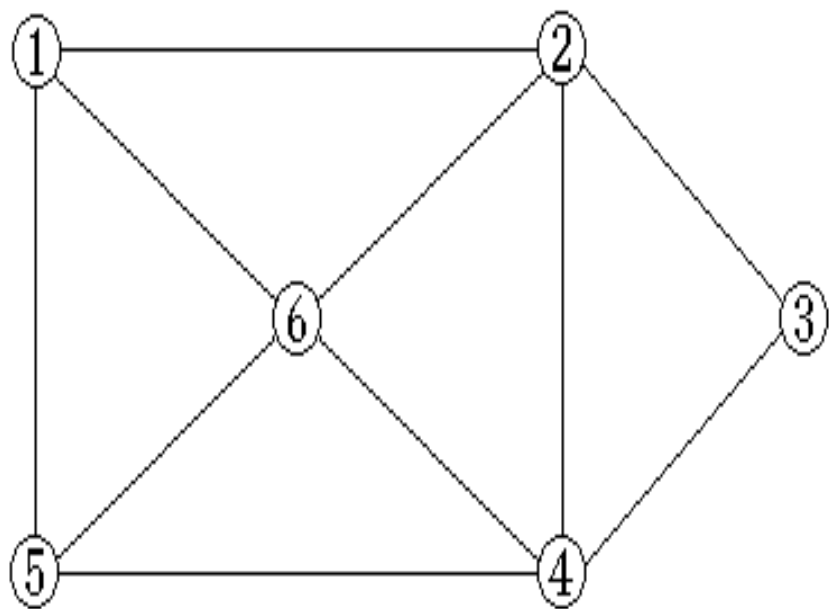
# 堆结构



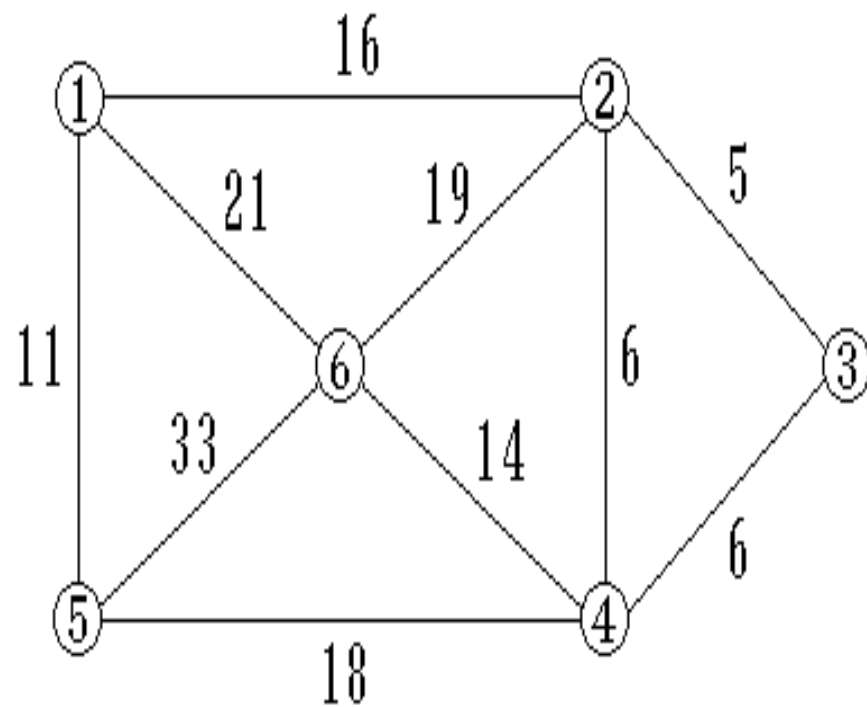
大根堆



小根堆



图结构



网络结构

# 数学模型在计算机中实现

- 理解计算机的运作模式

# 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**

# 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？



## • 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**

## 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

# 算法的评价标准

# 问题

求前n项的和值： $1+2+\cdots+n=?$

# 数学公式法

```
#include<iostream>
using namespace std;
void main()
{
    int n;
    cin>>n;
    cout<<n*(n+1)/2;
}
```

# 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)**



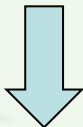
# 时间复杂度

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

2.s=0;

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

s=s+1;



s=0;

i=0;

L1: if(i==n) 结束;

s=s+1;

i=i+1;

goto L1;

# 时间复杂度

3. for (i=0;i<n;i++)  
    if ( A[i]==x) break;

- 最好情况:
- 最坏情况:
- 平均情况:

$$1/n * (1+2+\dots+n)$$

0	10
1	20
2	5
3	30
4	8
	9
	40
	...
	...
n-1	100

# 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<=n;i++)  
  for (j=1;j<=n;j++) x= x+1;
```

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

# 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);
}
```

F(5)  
↓  
F(4)  
↓  
F(3)  
↓  
F(2)  
↓  
F(1)

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

$$O(2^n)$$

# 空间复杂度

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

# 空间的复杂性

问题：数组逆置

A	20	30	5	8	...	...			10
	0	1	2	3	...	...			n-1

B									
	0	1	2	3	...	...			n-1

方法一：

for (i=0; i<n; i++) B[n-i-1]= A[i];

for (i=0; i<n; i++) A[i]= B[i];



# 空间的复杂性

问题：数组逆置

A	20	30	5	8	...	...			10
	0	1	2	3	...	...			n-1

方法二：

```
for (i=0; i<n/2; i++)  
{  
    w = A[i];  
    A[i] = A[n-i-1];  
    A[n-i-1] = w;  
}
```

# 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
- <https://pintia.cn/problem-sets/91827364500/exam/problems/type/7>
- poj.org
- 算法测试
- <https://leetcode-cn.com/>
- <https://pintia.cn/>