

# 数据结构与算法

## 第一章 绪论

王 昭

北京大学信息科学技术学院

wangzhao@pku.edu.cn



# 内容提要

- 课程简介

- 第一章

绪论



# 为什么要学习该课程

- 学分

- 毕业





# Computer is everywhere

计算改变了生活





Computing in the 21<sup>st</sup> Century “二十一世纪的计算”学术研讨会

计算人生  
COMPUTING LIFE

- 计算彻底融入到我们的生活，成为必不可少的一部分时，我们已经进入了“计算生活”时代。
- 而将“**计算生活**”从概念变为现实的是那些将毕生精力投入于计算科学的大师们，他们在创造这个时代的同时，也成就了自己精彩的“**计算人生**”。

----- 张益肇





# 计算成果

- 众所周知的高科技医疗器械CT，即是X 射线技术与计算技术相结合的创新，其理论的首创者和器械的首创者共同获得了1979 年诺贝尔医学和生理学奖。
- 其他与计算有关的诺贝尔奖获得者还有：
- 威尔逊因重正化群方法获1982 年物理学奖，
- 克鲁格因生物分子结构理论获1982 年化学奖，
- 豪普曼因X 光晶体结构分析方法获1985 年化学奖，
- 科恩与波普尔因计算量子化学方法获1998 年化学奖
- 闻名遐迩的中国科学大师华罗庚的“华-王方法”，冯康的有限元方法，以及吴文俊的 吴方法 ，也均是与计算有关的重大科学创新



# 四色问题

- **四色问题**又称四色猜想、四色定理，是世界近代三大数学难题之一。
- **四色问题**的内容是“任何一张地图只用四种颜色就能使具有共同边界的国家着上不同的颜色。”
- **图四色定理**（Four color theorem）是**1852**年由一位叫Francis Guthrie的英国大学生提出来的。
- **1878~1880**年两年间，著名的律师兼数学家肯普(Alfred Kempe)和泰勒(Peter Guthrie Tait)两人分别提交了证明四色猜想的论文，宣布证明了四色定理。
- **1890年**，在牛津大学就读的年仅**29**岁的赫伍德以自己的精确计算指出了肯普在证明上的漏洞。



# 四色问题

- 美国数学家富兰克林于1939年证明了22国以下的地图都可以用四色着色。
- 1950年，温恩从22国推进到35国。
- 1960年，有人又证明了39国以下的地图可以只用四种颜色着色；随后又推进到了50国。
- 1976年6月，在美国伊利诺斯大学的两台不同的电子计算机上，用了1200个小时，作了100亿个判断，结果没有一张地图是需要五色的，最终证明了四色定理。





# Niklaus Wirth



- **Niklaus Wirth, 1934年出生于瑞士, 1963年在加州大学伯克利分校取得博士学位**
- **1968年创建与实现了Pascal语言**
- **凭借一句话获得图灵奖的Pascal之父—Niklaus Wirth, 让他获得图灵奖的这句话就是他提出的著名公式:**

**“算法+数据结构=程序”**

- **1971年, 沃思基于其开发程序设计语言和编程的实践经验, 在4月份的 Communications of ACM上发表了论文“通过逐步求精方式开发程序’ (Program Development by Stepwise Refinement), 首次提出了“结构化程序设计” (structure programming) 的概念。**

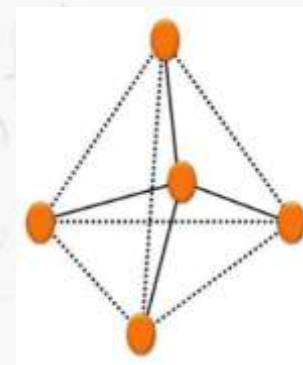
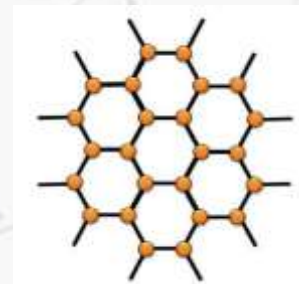
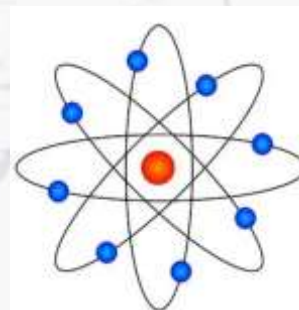
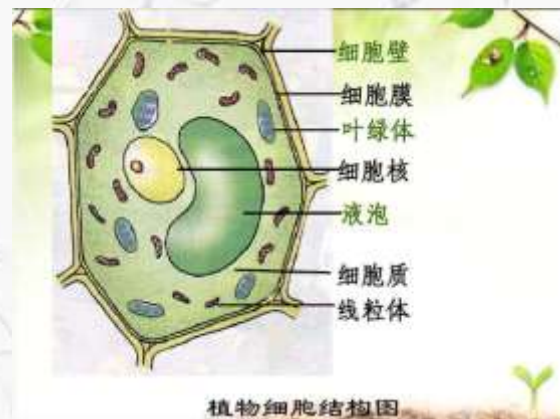
**自顶向下、逐步求精及模块化**



# 数据结构

- 汉字结构
- 文章结构
- 细胞结构
- 物质结构
- 地球的内部结构
- 建筑结构

上下结构	思、华
	霜、花
	基、想
上中下结构	意、
	褒、裹
	村、联
左右结构	伟、搞
	刚、郭



- 结构: **实体+关系**, 把某些**成份**按一定的规律或方式组织在一起的**实体**或某些**成份**组织在一起的**方式**
- 在这里, 我们把实体看作**数据**



# 算法

- 是对特定**问题求解方法和步骤**的一种描述。

- 最大公因数的求解算法
- 一元二次方程的求解
- 圆周长、圆面积
- 立方体的表面积和边长
- 排序
- 分治、贪心、动态规划……





# 计算效率

- 100,000个数据排序
- 冒泡13分钟
- 快排0.3秒
- 计算时间的差异决定了是否值得做



# 算法+数据结构=程序

Niklaus Wirth

- **程序**：为计算机解决问题编制的指令集，是按照事先设计的功能和性能要求执行的指令序列
- 从程序设计的观点来看，
  - **信息的表示**：“数据结构”研究的问题
  - **信息的处理**：“算法”研究的问题
- 了解计算机原理、掌握程序设计的必由之路。



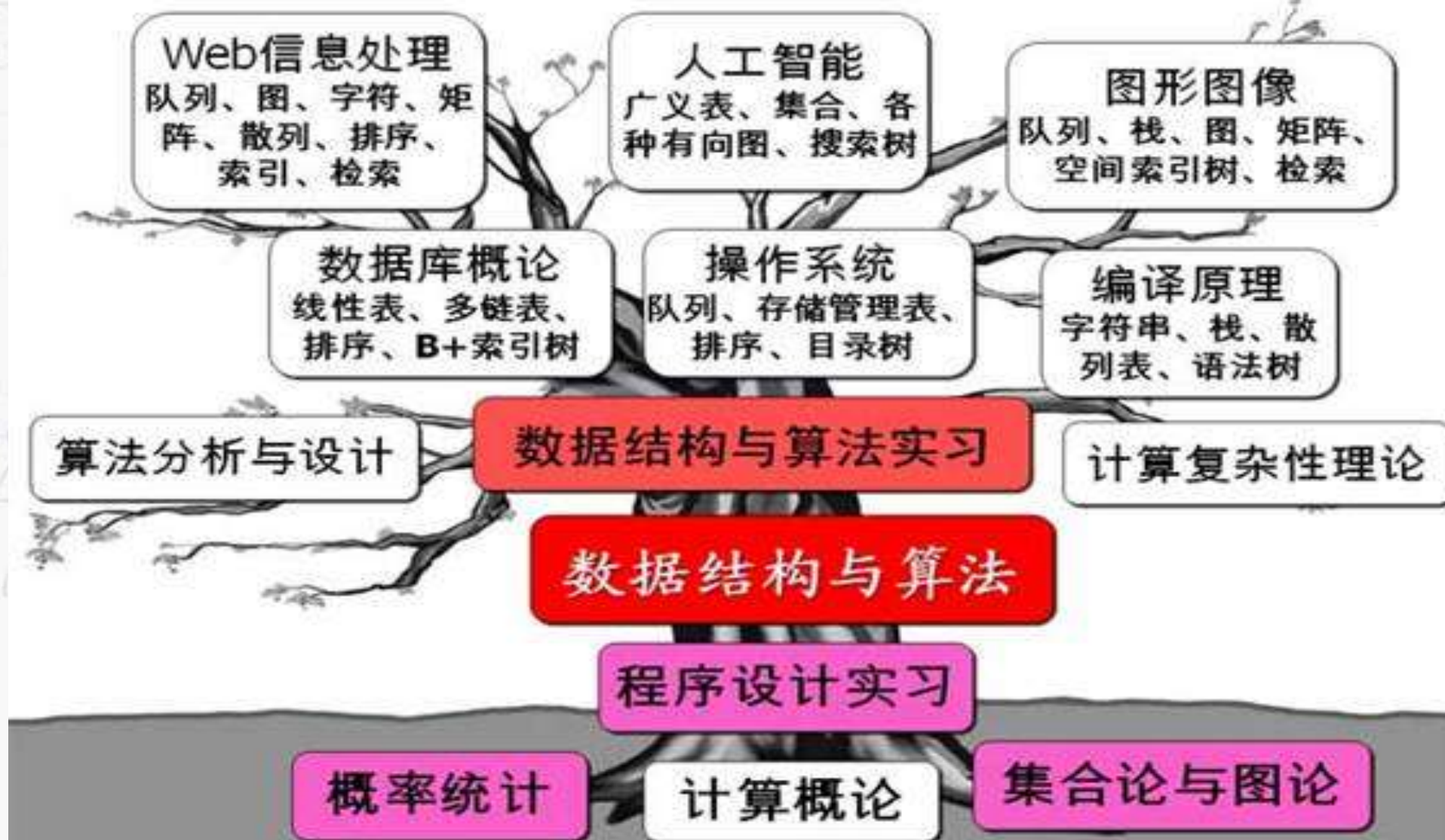
# 课程目标

- 学会怎样组织信息，以便支持高效的数据处理
  - 掌握常用的数据结构及其应用
  - 学会合理组织数据、有效地处理数据
  - 基本掌握算法的设计与分析方法
  - 提高程序设计能力





# “数据结构与算法”课程与计算机专业其他课程的关系



# 一堂让人变得更聪明的课

- 是需要调整思维习惯和方式而非仅仅充实知识库。
- 要变得聪明一些，就要学会选择适当的角度
- 一旦领会，终生受益。



# 建议的学习方案

- 听课，思考，提问，讨论
  - 三人行，必有我师焉
  - 学而不思则罔，思而不学则殆
  - 不耻下问
  - 独学而无友 则孤陋而寡闻
- 上机
  - 纸上得来终觉浅，绝知此事要躬行
- 听懂很容易，学会才是真



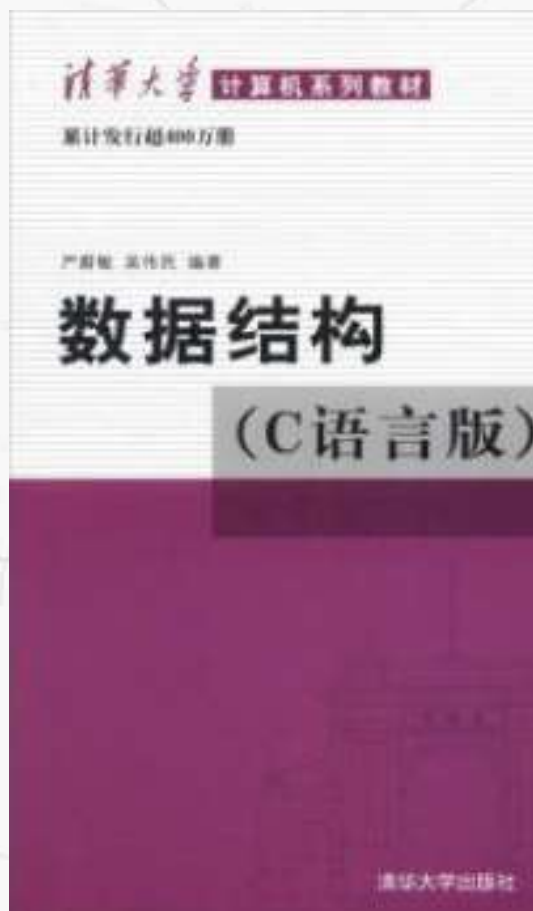


# 教材与参考书

- 教材：
  - 算法与数据结构-C语言描述（第3版），  
张乃孝主编，高等教育出版社，2011，6
- 参考书：
  - 张铭，赵海燕，王腾蛟，数据结构与算法，北京：高等教育出版社，2008年
  - 数据结构-C语言版，（有配套习题集与习题解答）严蔚敏等，清华大学出版社
  - 霍红卫译，算法：C语言实现（第1~4部分）基础知识、数据结构、排序及搜索（原书第3版），(美)Robert Sedgewick著，(Pts. 1-4)，北京：机械工业出版社，2009 年10月



# 教材与参考书



以 教材 和 参考书 为主，  
充分利用各种在线资源和帮助系统。



# 教学方式

- 课堂讲授： 3学时/周

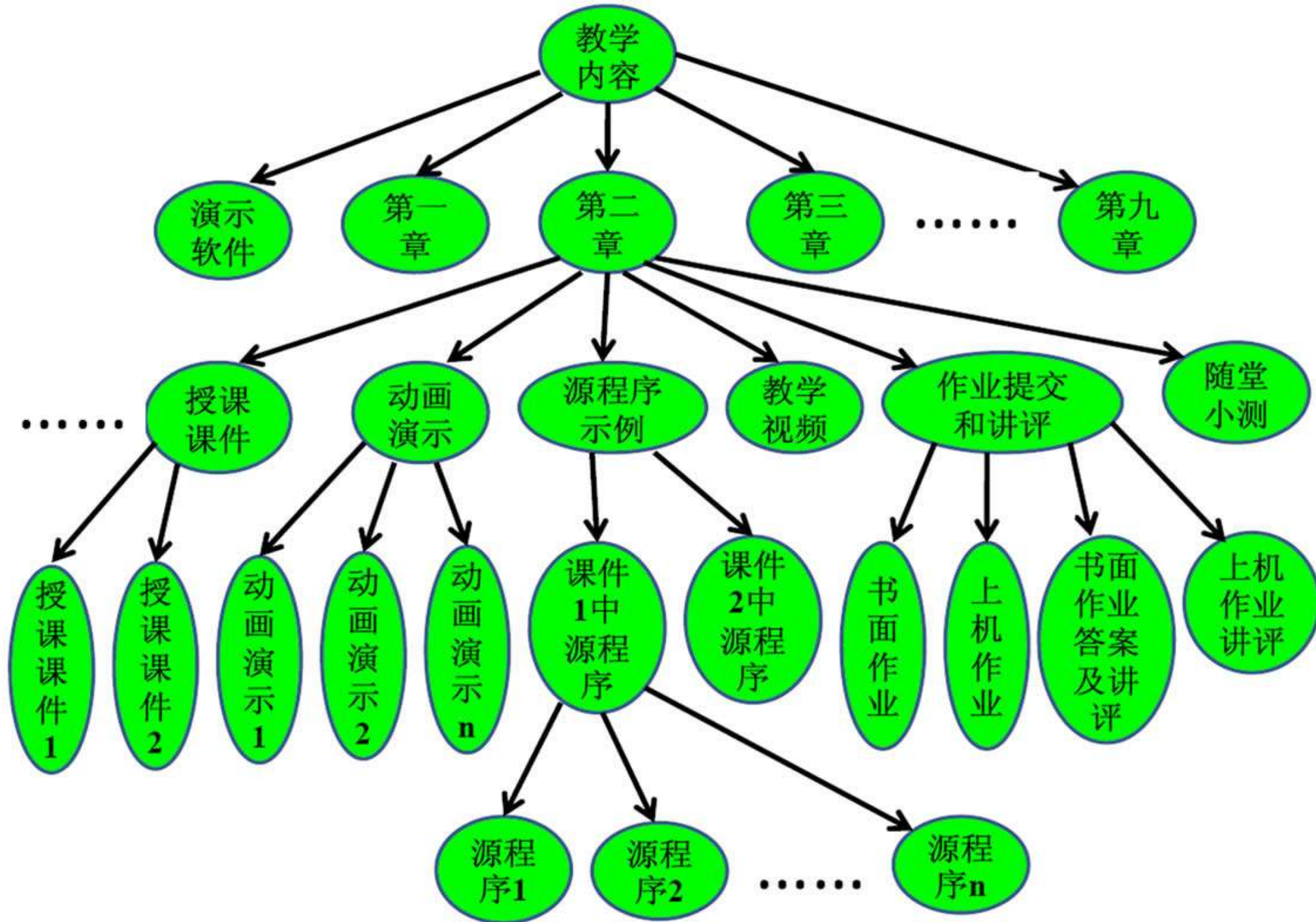
每周： 星期二7~9节（15:10~18:00）

- 上机实践： 2学时/周，

每周： 星期五9~10节（17:40~19:30）







# 教学辅导

- 日常交流

- 大家可以利用课程网站及课程微信群进行交流、讨论，或向助教发送E-mail询问问题。有关课程的讲义、作业、通知等都将在网上发布。
- 在上课或者上机课上讨论

- 搜索引擎

- <http://www.google.com/>
- <http://www.baidu.com/>

- 百科全书

- <http://www.wikipedia.org/>
- <http://baike.baidu.com/>



# 中国专业IT社区

- <https://www.csdn.net/>
  - 中国专业IT社区CSDN (Chinese Software Developer Network) 创立于1999年
- <http://www.chinaunix.net/>
  - 中国最大的Linux/Unix技术社区网站，也交流程序开发

程序设计			本版新帖
 <b>C/C++</b> 最后发表: 01-06 14:59 by cdfarsight RSS订阅 主题: 78757 帖数: 671222	 <b>Linux环境编程</b> 最后发表: 01-26 12:06 by ouyixq RSS订阅 主题: 15459 帖数: 69036	 <b>内核源码</b> 最后发表: 10-05 18:35 by jiufei19 RSS订阅 主题: 25583 帖数: 155636	
 <b>Shell</b> 最后发表: 02-20 21:23 by shang2010 RSS订阅 主题: 63564 帖数: 528038	 <b>Perl</b> 最后发表: 07-19 11:39 by lij RSS订阅 主题: 24296 帖数: 168460	 <b>Java</b> <a href="#">Java文档中心</a> 最后发表: 02-10 11:15 by cdhqyj RSS订阅 主题: 33155 帖数: 87410	
 <b>PHP</b> <a href="#">PHP文档中心</a> 最后发表: 12-17 21:48 by swingcoder RSS订阅 主题: 27816 帖数: 114630	 <b>Python</b> <a href="#">Python文档中心</a> 最后发表: 01-19 13:53 by 唐胜 RSS订阅 主题: 14635 帖数: 71918	 <b>Ruby</b> 最后发表: 05-19 10:34 by Sevk RSS订阅 主题: 1150 帖数: 5827	
 <b>嵌入式开发</b> 最后发表: 02-13 17:40 by cdhqyj RSS订阅 主题: 9264 帖数: 47342	 <b>驱动开发</b> 最后发表: 01-04 14:20 by xiaohengsang RSS订阅 主题: 3401 帖数: 18232	 <b>Web开发</b> 最后发表: 01-13 17:17 by cdhqyj RSS订阅 主题: 8245 帖数: 19402	
 <b>架构设计</b> 最后发表: 04-11 15:18 by 夏末18844 RSS订阅 主题: 389 帖数: 4175	 <b>CPU与编译器</b> 最后发表: 10-11 14:09 by 视壮 T_19928853549 RSS订阅 主题: 2025 帖数: 14895	 <b>软件配置管理</b> 最后发表: 12-16 18:00 by boxpei RSS订阅 主题: 2944 帖数: 13215	
 <b>Golang</b> 最后发表: 12-05 19:54 by jerry_shen RSS订阅 主题: 93 帖数: 740	 <b>Erlang</b> 最后发表: 10-23 17:18 by ighack RSS订阅 主题: 237 帖数: 1425		





# 成绩考核

**“学生成绩=平时成绩+上机考试+期末考试成绩”**

{	平时成绩 30%	书面作业（一定要按时交） 8 %
		随堂测验+考勤 4%
		上机作业 18 %
{		
上机考试 30%		
{		
期末考试成绩 40%		

**“优秀率(85分以上)原则上不超过40%，不及格率(60分以下)不超过10%。”**

- 备注：注重学生综合能力的考评，平时表现突出、上机能力较强的可以考虑通过一定方式奖励加分。



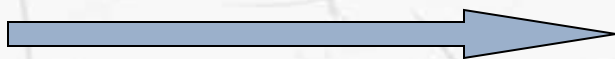


# 作业要求

- 按时交，杜绝抄袭（一旦发现，该次作业成绩0分）

- 书面作业

- 上机作业  
程序编写  
程序调试  
运行结果  
助教检查  
上机报告



- 做什么
- 怎么做
- 结果
- 体会与收获



# 上机安排

- 上机时间  
周五 9-10节 (17:40~19:30)
- 上机地点:  
计算中心机房 5/6 号机房 (理科1号楼2层)
- 辅导助教:
  - 刘星宇明<liuxym@pku.edu.cn>
  - 戴舒羽<daishuyu@pku.edu.cn>
  - 张明韬<mingtaozhang@pku.edu.cn>
  - 南馨语<2301213095@pku.edu.cn>
  - 张子涵<2201213135@stu.pku.edu.cn>



# 上机要求

- 上机环境

**Win10专业版, Bloodshed Dev C++ 5.11, Code Blocks 20.03,  
Eclipse 4.14.0, Visual C++ 2019, Python 3.2, 3.8, 3.9  
Visual Studio Code 1.67.0, 1.75.1**

- 要求

- 认真准备，有备而来；
- 严禁玩游戏；
- 及时向辅导老师反映问题；
- 培养独立解决问题的能力 。



# 答疑安排

- 平时有疑问请在课下及时解决，如有问题发邮件
- 考试前安排**1-2**次答疑时间
- 有何意见及建议请及时反映





# 内容提要

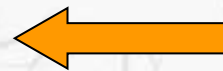
- 课程简介

- 第一章 绪论



# 第一章 绪论

- 本课程在计算机问题求解中的作用
- 数据结构的基本概念和术语
- 算法与算法评价
- 总结



# 数值计算与非数值计算

- 对于数值计算问题，处理的对象为简单的数值，数学模型为**数学方程**；
- 对于非数值计算问题（如资料查询、交通管理等），处理的对象之间具有一定的**逻辑关系**，其数学模型不能简单地用数学方程描述，**此时必须根据对象之间的逻辑关系建立描述问题的数据结构。**



# 非数值计算问题分析

- 信息管理（图书、档案、职工等）
- 交通管理（城市交通管理、航线、铁路、公路等）

关键：

数据的表示（数据结构问题）和数据的处理（算法问题）





## 线性表问题

每人一个记录，多个数据项；

什么样的逻辑关系？

如何存储？

什么样的操作？

（统计、检索问题）

编号	姓名	性别	年龄	月收入
1	李泉	男	51	980
2	王怡	女	47	945
3	张三	男	35	870
4	马丁	男	27	840
...	...	...	...	...



# 集合的并

- 空气污染严重的城市有：太原、北京、乌鲁木齐、兰州、重庆、济南、石家庄、青岛、广州、沈阳。
- 水污染严重的城市有：临汾、阳泉、大同、石嘴山、三门峡、金昌、石家庄、重庆、株洲和洛阳。
- 列出空气污染严重或水污染严重的城市（集合的并）。
- 第一种解法：  $O(mn)$
- 第二种解法：  $O(m+n)\log(m+n)$
- 第三种解法：  $m\log n$  或  $n\log m$



如现国家要统计某年我国的野生植物保护情况。现有各省市各种珍稀野生植物的种植面积，要统计各种野生植物在我国总的种植面积。

重复的树种很多

西藏	巨柏	XX 公顷
	长叶松	XX 公顷
	白皮松	XX 公顷
	云杉	XX 公顷
	冷杉	XX 公顷
	铁杉	XX 公顷
四川	金钱松	XX 公顷
	银杉	XX 公顷
	银杏	XX 公顷
	水杉	XX 公顷
	云杉	XX 公顷
	冷杉	XX 公顷
	红杉	XX 公顷
	红豆杉	XX 公顷
	连香树	XX 公顷
湖南	银杉	XX 公顷
	水杉	XX 公顷
	云杉	XX 公顷
	金钱松	XX 公顷
	伯乐树	XX 公顷
	连香树	XX 公顷
	蕨囊蕨	XX 公顷
.....	.....	.....



# 根据各城市的GDP值，给出全国GDP的总排名。

某年南方**GDP**排名前二十如下： 同年北方**GDP**排名前二十如下：

排名	城市	GDP 及增长
01	上海	10297+12.0%
02	广州	6068+14.7%
03	深圳	5684+15.0%
04	苏州	4820+15.5%
05	重庆	3486+12.2%
06	杭州	3441+14.3%
07	无锡	3360+15.0%
08	佛山	2927+19.3%
09	宁波	2864+13.4%
10	南京	2774+15.1%
11	成都	2750+13.8%
12	东莞	2624+19.1%
13	武汉	2590+14.8%
14	泉州	1901+15.0%
15	温州	1834+13.3%
16	长沙	1791+14.8%
17	南通	1758+15.7%
18	绍兴	1678+13.2%
19	福州	1660+12.2%
20	常州	1560+15.0%

排名	城市	GDP 及增长
01	北京	7720+12.0%
02	天津	4338+14.4%
03	青岛	3207+15.7%
04	大连	2568+16.4%
05	沈阳	2483+16.5%
06	烟台	2402+17.0%
07	唐山	2362+14.8%
08	济南	2185+15.7%
09	哈尔滨	2094+13.5%
10	石家庄	2064+13.2%
11	郑州	2002+15.7%
12	长春	1934+14.5%
13	潍坊	1721+16.5%
14	淄博	1645+15.8%
15	大庆	1618+11.4%
16	济宁	1456+16.5%
17	西安	1450+12.8%
18	东营	1450+17.0%
19	临沂	1405+16.3%
20	威海	1369+15.9%





找出下列DNA片断是否包含

**TTCCTATGGGAGTGGCCCTCAGTCCGTTTCTCCTGGGCTCAGTTTACTA**序列。

- **DNA SEQUENCE:**

- **ATGGGAGGTT CGTCTTCCAA AGCTCGACAA GGCATGGGGA  
CGAATCTTTC TGTTGCCAAT**

- **CCTCTGGGAT TCTTTCCCGA TCACCAGTTG GACCCTGGGT  
TGGGAGCCAA CTCAAACAAT**

- **CCAGATTGGG ACTTGAACCC CAACAAGGAT CACTGGCCAG  
AGGCAAATCA GGTAGGAGCG**

- **GGAGCATTCTG GGCCAGGGTT CACCCCACCA CACGGCGGTC  
TTTTGGGGTG GACCCCTCAG**

- **GCTCAGGGGA TTTTGACAAC AGTGCCAGTA GCACCTCCTC  
CTGCCTCCAG CAATCGCCAG**

- **CAGTGGAATC CACAACATT CCACCAACCT CTGCCAGACC  
CCAGAGTGAG GGGCCTATAC**



某生态系统的食物网如下图所示。找出对山狮的生存有影响的动物。

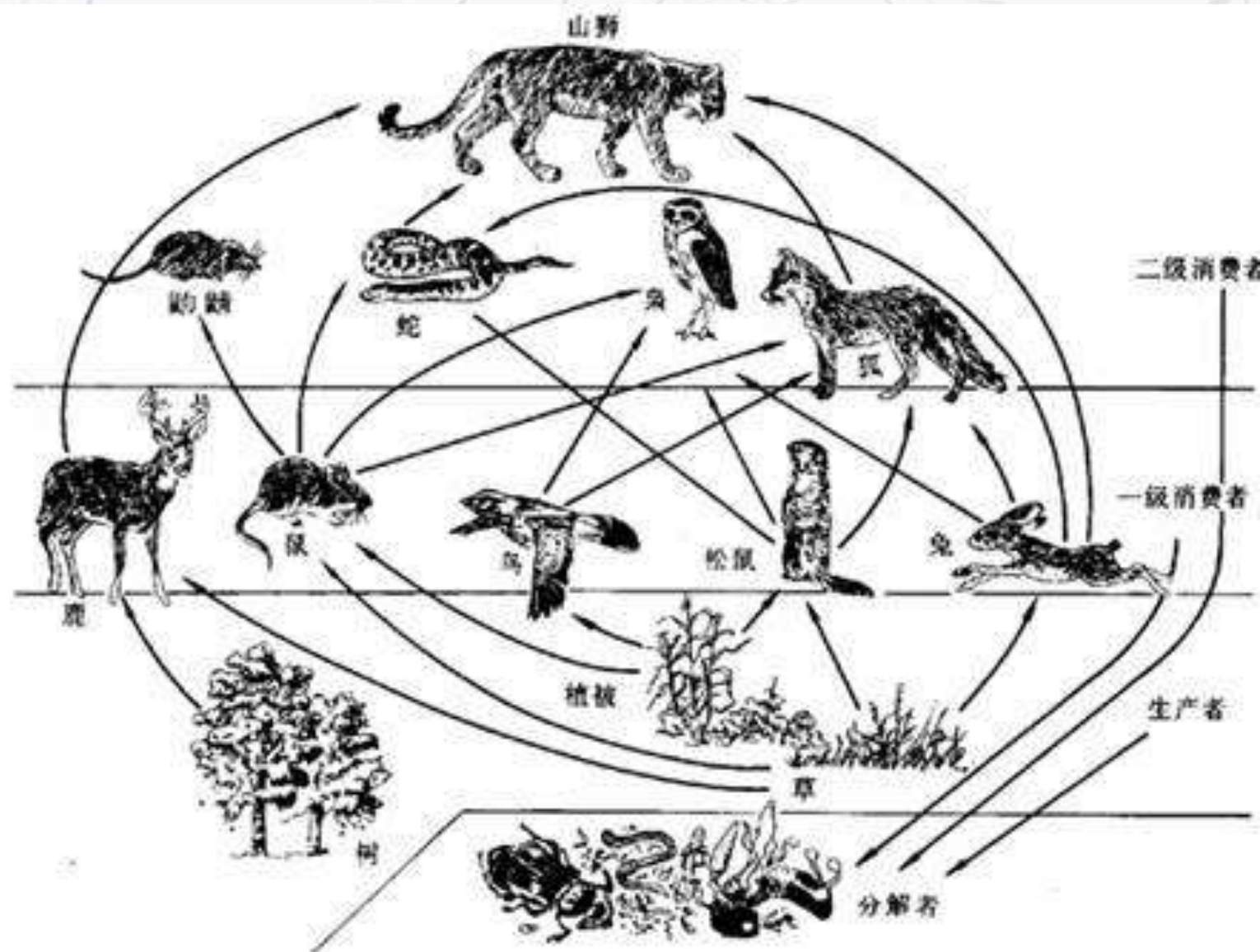


图 9-5 一个陆地生态系统的部分食物网





Google地图，输入出发起始点和到达终止点，网站即推荐乘车方式。

from: 北京崇文区天坛 to: 北京海淀区中关村北大街116号北京大学 - Google 地图 - Windows Internet Explorer

http://ditu.google.com/ Live Search

北京公交-地图 from: 北京崇文区天坛 t... 百度地图搜索\_从北京大学...

已保存地址 | 登录 | 帮助

Google 地图 BETA 网页 图片 资讯 地图 论坛 更多 »

北京崇文区天坛 北京海淀区中关村北大街116号北京 行车路线

搜索地图 搜索周边 行车路线

搜索结果 获取返程路线

自: 北京崇文区天坛 修改

驾驶: 21.9 公里

1. 进入天坛路向东南 232 米
2. 左转掉头进入天坛路向西 0.6 公里
3. 右转进入祈年大街向北 1.4 公里
4. 左转进入前门东大街向西北 9.2 公里
5. 靠左进入西直门北大街向北 2.9 公里
6. 右转进入蓟门桥向东北 47 米
7. 右转进入北三环西路辅路向西 0.6 公里
8. 靠左进入北三环西路向西 1.9 公里
9. 靠右进入北三环西路辅路向西 0.6 公里
10. 右转进入中关村大街向北 3.6 公里
11. 左转掉头进入中关村北大街向南 0.8 公里

至: 北京海淀区中关村北大街116号北京大学 修改

本行车路线仅供计划之用，实际路况可能不同于地图显示结果。

地图数据 ©2007 Mapabc.com

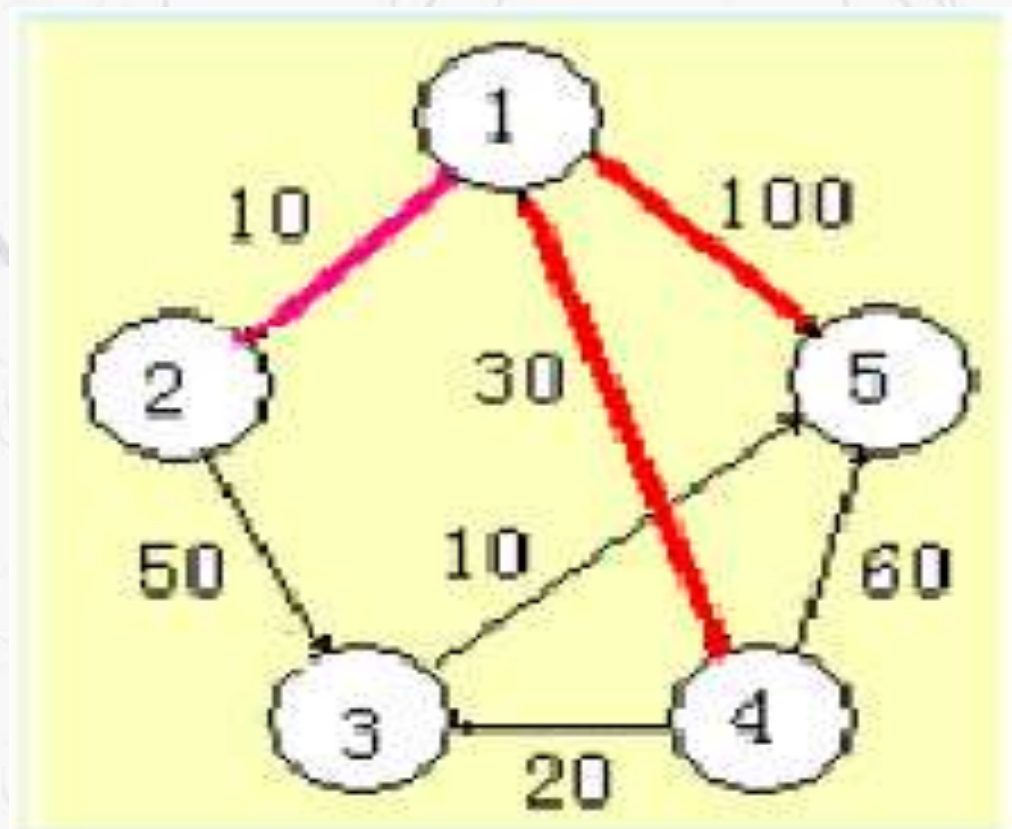
打印 电子邮件 显示本页链接

5 公里 2 英里

©2007 Google - 地图数据 ©2007 Mapabc.com - 使用条款

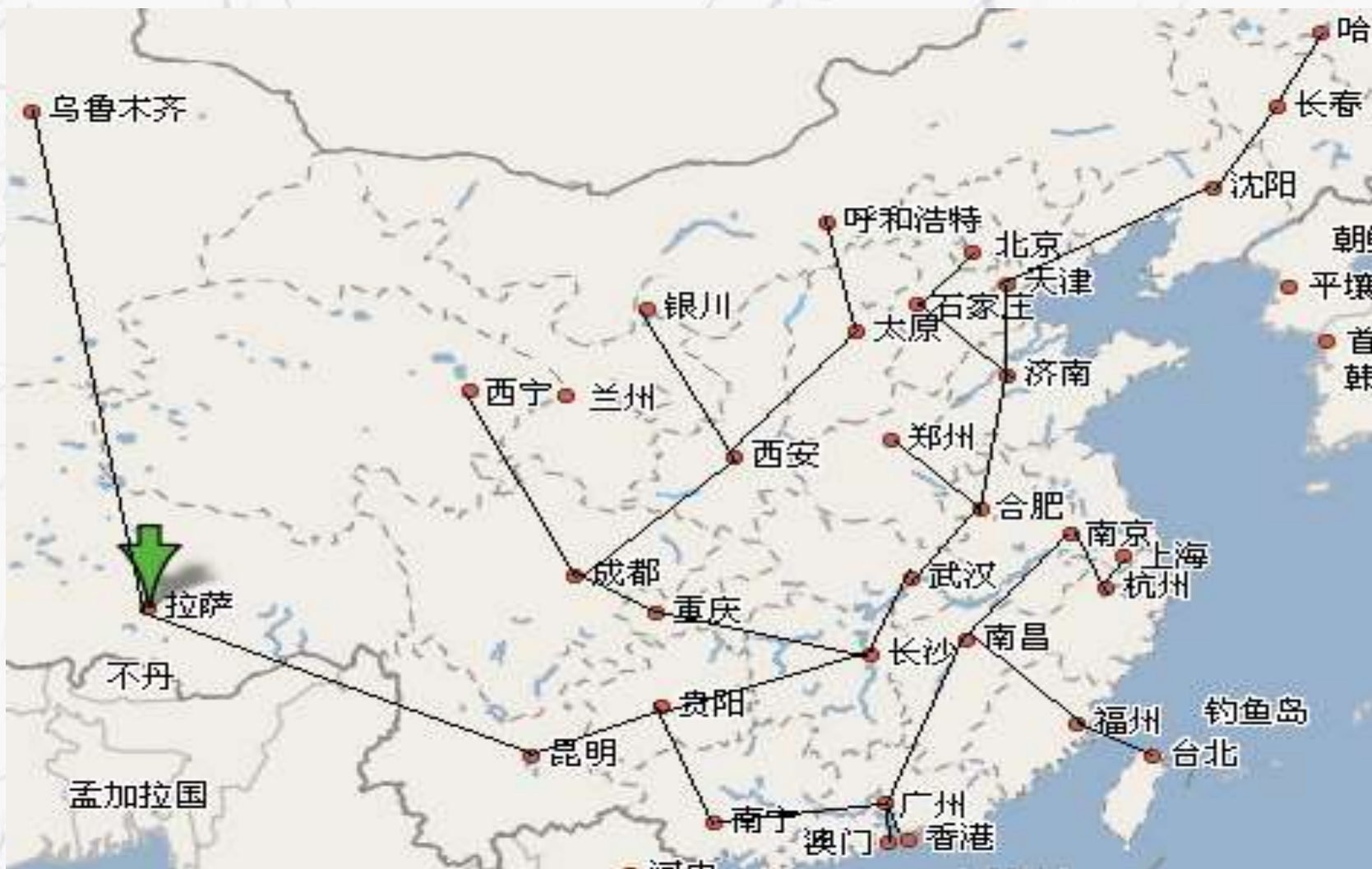
Internet e 100%

Google地图，输入出发起始点和到达终止点，网站即推荐乘车方式，本质即为**最短路径问题**。图中的距离还可根据交通拥塞程度加权。



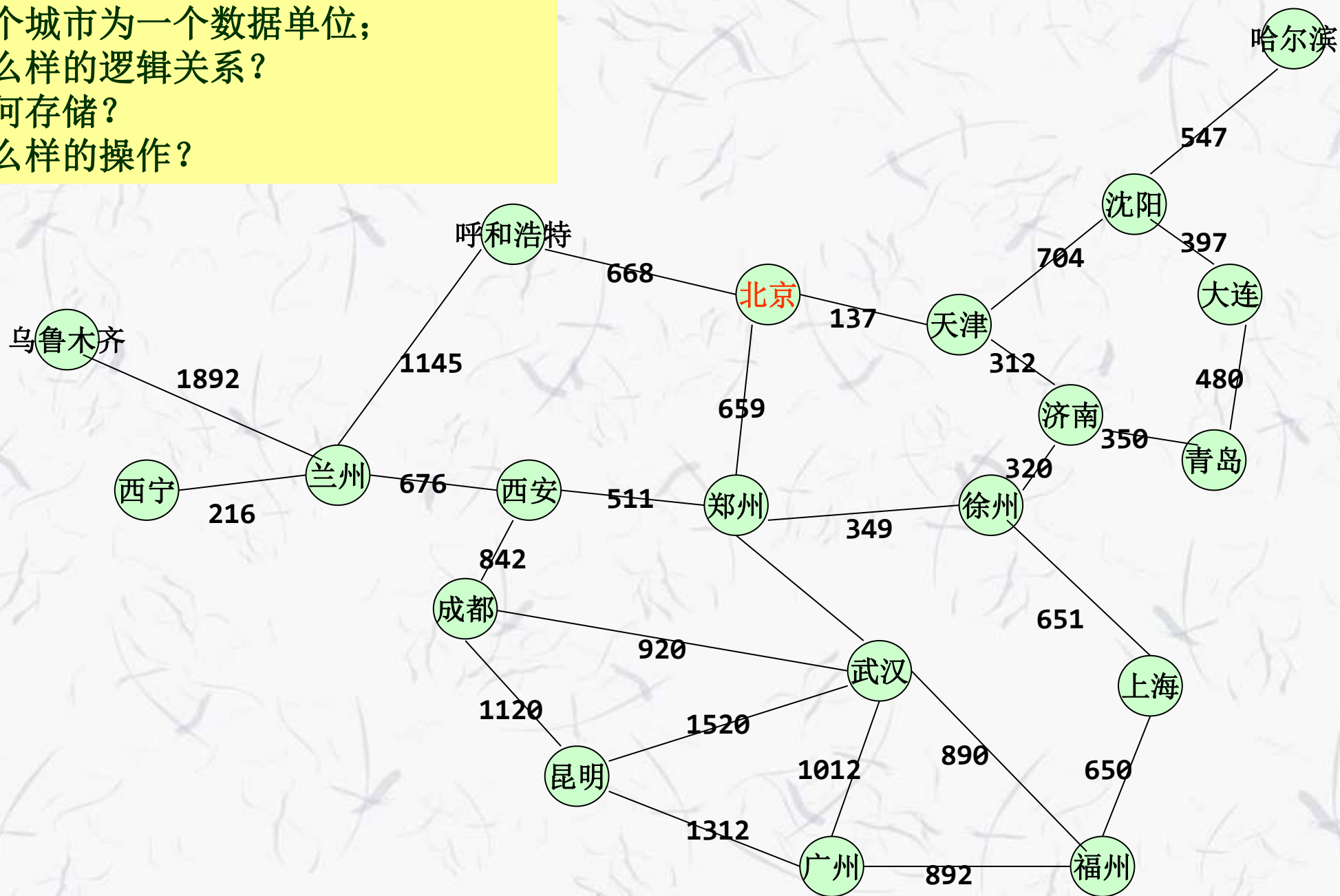


假设在中国有**30个**城市，城市之间想建公路，假设公路的建设费用与公路的长度成正比，而这**30个**城市之间的位置都已经假定，分别用坐标表示，**如何链接公路使得所有的城市都能链接，并且成本最小？**

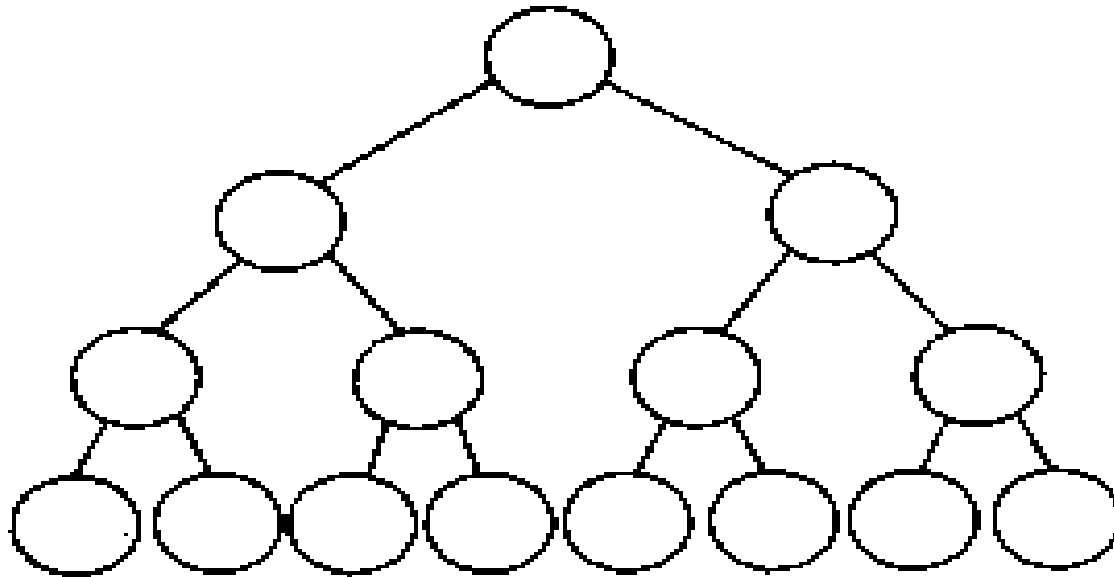


## 非线性问题

每个城市为一个数据单位；  
什么样的逻辑关系？  
如何存储？  
什么样的操作？



工厂污水检测问题：假设有**10000**个工厂，现在从每个工厂里面的排水系统获得样品，每个工厂的排水样品被访到一个标号的试管里面，然后我们的工作就是对样品就行污水测试。从而线性测试的成本为**0 (10000)**，显然任务比较繁重，而且实际中有很多工厂还是比较尊守法律，不会排出污水，请问用什么样的方法才能简化污水的测试？这里假设不同工厂污水之间不发生化学反应？





# 计算机解决问题的过程

- **分析阶段**：弄清所要解决的问题是什么，并用一种语言清楚地描述出来。
- **设计阶段**：建立程序系统的结构，重点是算法的设计和数据结构的设计。
- **编码阶段**：采用适当的程序设计语言，编写出可执行的程序。
- **测试和维护**：发现和排除在前几个阶段中产生的错误，经测试通过的程序便可投入运行，在运行过程中还可能发现隐含的错误和问题。





## 多叉路口交通信号灯的管理：

(一) 问题分析：首先需要分析一下所有车辆的行驶路线的冲突问题。这个问题可以归结为对车辆的可能行驶方向作某种分组，对分组的要求是使任一个组中各个方向行驶的车辆可以同时安全行驶而不发生碰撞。

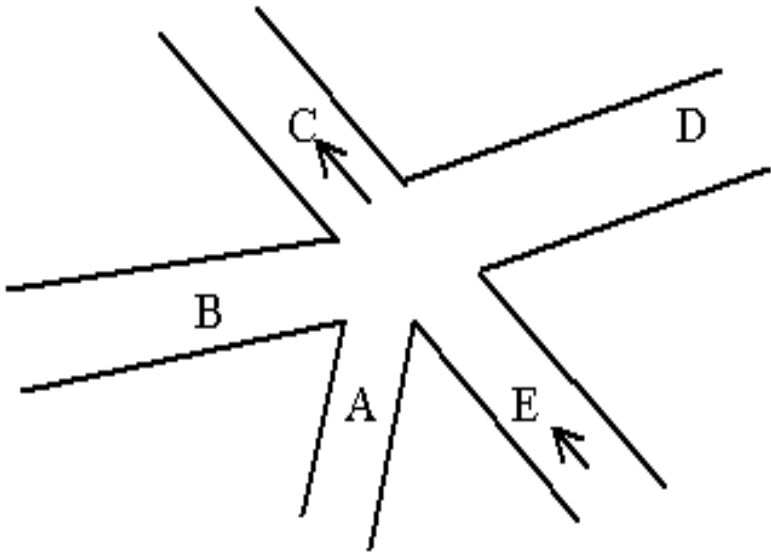


图 1.1 一个交叉路口的模型

根据这个路口的实际情况可以确定13个可能通行方向：

A→B, A→C, A→D, B→A,  
B→C, B→D, D→A, D→B,  
D→C, E→A, E→B, E→C,  
E→D。



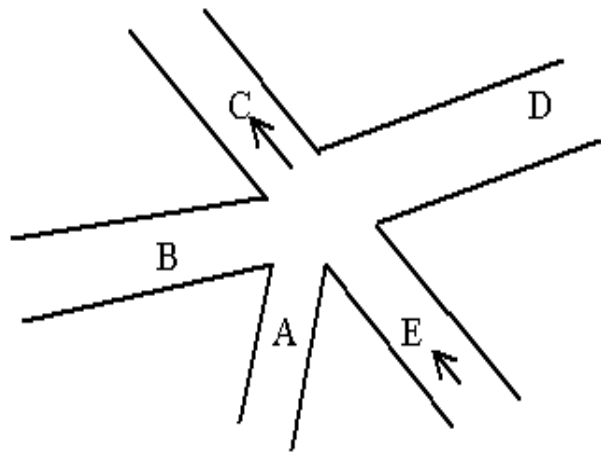


图 1.1 一个交叉路口的模型

为了叙述方便，我们下面把  $A \rightarrow B$  简写成  $AB$ ，并且用一个小椭圆把它框起来，在不能同时行驶的路线间画一条连线(表示它们互相冲突)，便可以得到图1.2所示的图式。

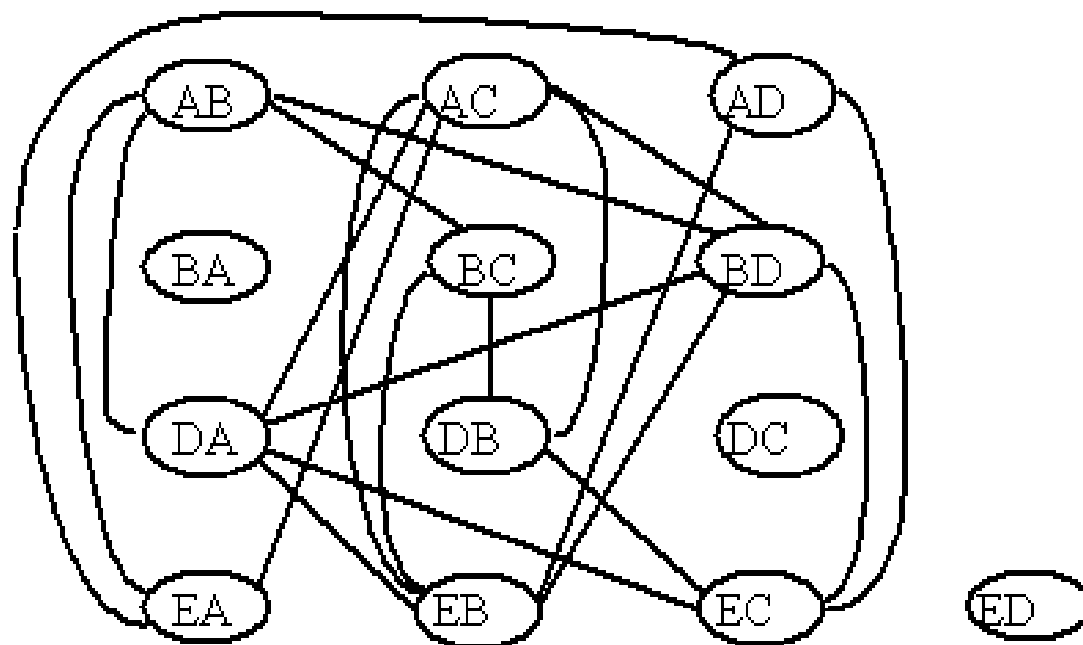
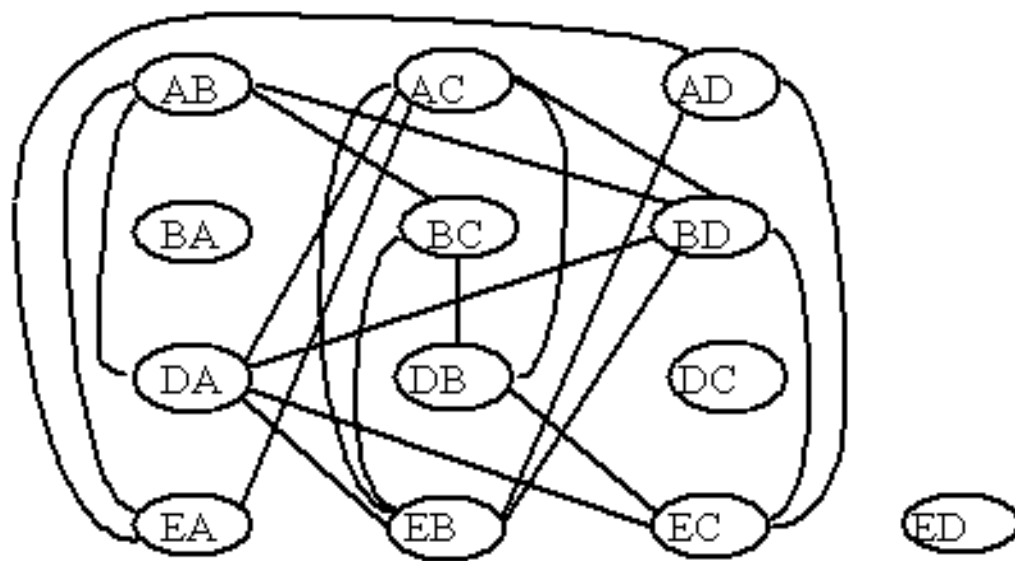


图 1.2 交叉路口的图式模型



- 这样做就把要解决的问题借助图的模型变成了另一个抽象问题：要求将图1.2中的结点分组，使有线相连(互相冲突)的结点不在同一个组里。
- 如果把上图中的一个结点理解为一个国家，结点之间的连线看作两国有共同边界，上述问题就变成著名的“着色问题”：即求出要几种颜色可将图中所有国家着色，使得任意两个相邻的国家颜色都不相同。
- 通过上面的分析，我们就获得了该交通管理系统的**数学模型**：图



## (二) 算法和数据结构的设计

- 可行解：满足要求的普通解。
- 最优解：分组数最少的解。
- 次优解：分组数接近最优解的可行解。
- 数据结构中涉及的算法大致有如下一些：
  - 穷举法
  - 贪心法：如着色问题
  - 分治法：如二分法检索
  - 回溯法：如迷宫问题
  - 动态规划法
  - 。 。 。 。 。





# 算法设计

## 算法设计

1. 对 $n$ 个结点，逐个测试其所有组合；
2. “贪心法”

```
while 有结点未着色 {  
    选择一种新颜色;  
    在未着色的结点中, 给尽可能多的彼此之间没有边的连接结点着色;  
}
```





# 分组结果

●把前述方法应用于交叉路口的模型图，得到下面的分组：

○绿色：AB, AC, AD, BA, DC, ED

○蓝色：BC, BD, EA

○红色：DA, DB

○白色：EB, EC



### (三) 编码

- 假设需要着色的图是**G**，集合**V1**包括图中所有未被着色的结点，着色开始时**V1**是**G**所有结点集合(用**G.V**表示)。**NEW**表示已用新颜色着色的结点集合。

从**V1**中找出可用新颜色着色的结点集的工作可以用下面的程序框架描述：

```
置NEW为空集合；  
for 每个  $v \in V1$  do  
    if  $v$ 与NEW中所有结点间都没有边：  
        从V1中去掉  $v$  ；  
        将  $v$  加入NEW ；
```





```
int colorUp(Graph G)
{  int color = 0;
   set V1 = G.V;      /*V1初始化为图G的结点集合V*/
   set NEW;
   while(!isEmpty(V1))
   {  NEW={ };
      while (v ∈ V1.notAdjacentWithSet(NEW, v, G))
      {  add (NEW, v);
         remove(V1, v);
      }
      ++color;
   }
   return (color);    /*返回使用的颜色数*/
}
```



# 学习数据结构要解决的问题

- 要描述**非数值计算问题**，单靠数学方程是无法解决的，它涉及表、图、树等类的数据结构。因此，数据结构课是一门研究**非数值计算问题**的程序设计中计算机的操作对象以及它们之间的关系和运算操作等的一门学科。
- 常见的计算机语言并不支持图、树、集合等数据结构，通常只提供基本的数据类型（整数、实数等）和一些数据构造手段（如数组、结构、指针等）。因此，复杂数据结构的设计以及相应的操作必须有用户自己实现。
- 用计算机求解问题，首先分析问题的需求，抽出抽象模型，然后设计适当的数据结构和有关的算法，最后采用计算机编程语言精确地描述所需要的数据和算法，实现程序。



# 第一章 绪论

- 本课程在计算机问题求解中的作用
- 数据结构的基本概念和术语
- 算法与算法评价
- 总结



# 基本概念和术语

- **数据**：指能够被计算机识别、存储和加工处理的信息载体。(数据是客观事物的符号表示。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**：就是数据的基本单位，在某些情况下，数据元素也称为元素、**结点**、顶点、记录。数据元素有时可以由若干**数据项**组成。数据项是具有独立含义的最小标识单位。
- **数据对象**：性质相同的数据元素集合，是数据的一个子集。



编号	姓名	性别	年龄	月收入
1	李泉	男	51	980
2	王怡	女	47	945
3	张三	男	35	870
4	马小丁	男	27	840
...	...	...	...	...





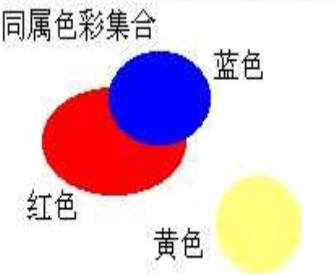
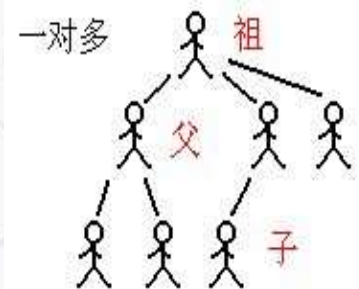

# 数据结构

- 没有公认的定义
- **数据结构(Data Structure)**: 按照逻辑关系组织起来的一批数据, 按一定的存储方法把它存储在计算机中, 并在这些数据上定义了相关运算的集合。
  - 一般包括三个方面的内容: 数据的逻辑结构、存储结构和数据的运算
    - **数据结构**: 是相互之间存在一种或多种特定关系的数据元素的集合。
    - **数据结构的表示**: 是指一种逻辑结构可以有不同的存储结构。
    - **数据结构的实现**: 在具体数据结构的表示的基础上各种操作的具体过程的描述。



# 数据的逻辑结构

- 数据元素之间的逻辑关系，也称数据的逻辑结构。与数据的存储无关，是独立于计算机的。数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。

	集合结构	线性结构	树形结构	图状或网状结构																														
特征	元素间为松散的关系	元素间为严格的一对一关系	元素间为严格的一对多关系	元素间为多对多关系																														
示例	<p>同属色彩集合</p> 	<p>一对一</p> <table><tr><th>编号</th><th>姓名</th><th>性别</th><th>年龄</th><th>月收入</th></tr><tr><td>1</td><td>李泉</td><td>男</td><td>51</td><td>980</td></tr><tr><td>2</td><td>王怡</td><td>女</td><td>47</td><td>945</td></tr><tr><td>3</td><td>张三</td><td>男</td><td>35</td><td>870</td></tr><tr><td>4</td><td>马丁</td><td>男</td><td>27</td><td>840</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr></table>	编号	姓名	性别	年龄	月收入	1	李泉	男	51	980	2	王怡	女	47	945	3	张三	男	35	870	4	马丁	男	27	840	...	...	...	...	...	<p>一对多</p> 	<p>多对多</p> 
编号	姓名	性别	年龄	月收入																														
1	李泉	男	51	980																														
2	王怡	女	47	945																														
3	张三	男	35	870																														
4	马丁	男	27	840																														
...	...	...	...	...																														

[集合、线性表、字符串、栈与队列、树与二叉树、字典、图]



# 数据的存储结构

- 数据元素及其关系在计算机存储器内的表示，称为**数据的存储结构**，数据的存储结构是逻辑结构用计算机语言的实现，它依赖于计算机语言。
- 顺序存储结构、链接（链状）存储结构、索引存储结构、散列存储结构。
- **四种存储结构既可单独使用，又可组合使用**



# 顺序存储结构

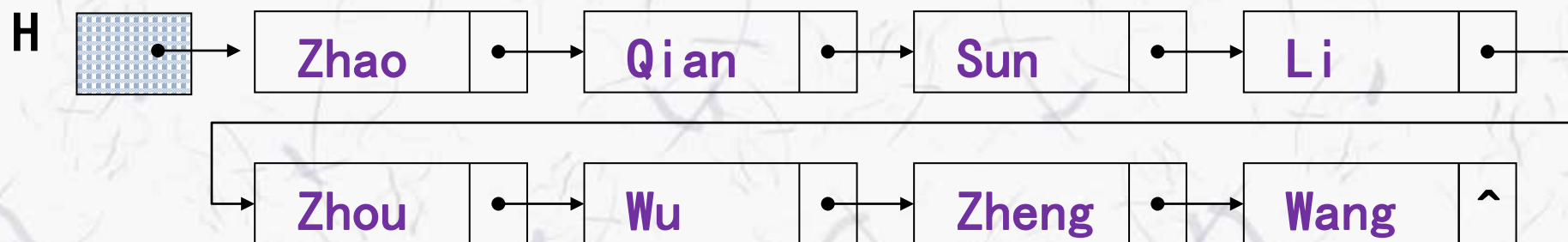
- **顺序存储结构**：它是把逻辑上相邻的结点存储在物理位置相邻的存储单元里，结点间的逻辑关系由存储单元的邻接关系来体现。





# 链接（链状）存储结构

- **链接（链状）存储结构**：它不要求逻辑上相邻的结点在物理位置上亦相邻，结点间的逻辑关系是由附加的指针字段表示的。

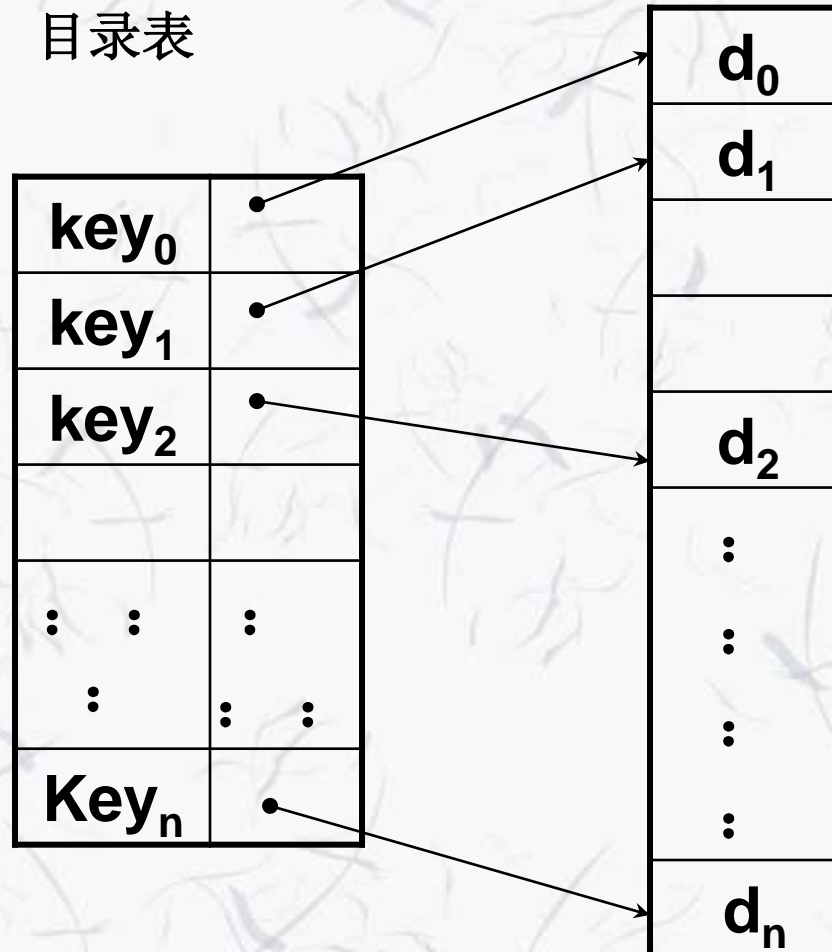


存储地址	数据域	指针域
1	Li	43
7	Qian	13
13	Sun	1
19	Wang	NULL
25	Wu	37
31	Zhao	7
37	Zheng	19
43	Zhou	25



# 索引存储结构

- 索引存储结构：除建立存储结点信息外，还建立附加的索引表来标识结点的地址。



# 散列存储结构

- **散列存储结构**：就是根据结点的关键字直接计算出该结点的存储地址。
- 假设以地区名作关键字，地区名以汉语拼音的字符表示，可以取这样的散列函数：求关键字的**第一个**和**最后一个**字母在字母表中的**序号之和**，然后判别这个值，若比30（表长）大，则减去30。

key	BEIJING 北京	TIANJIN 天津	HEBEI 河北	SHAXNXI 山西	SHANGHAI 上海	SHANDONG 山东	HENAN 河南	SICHUAN 四川
h2(key)	09	04	17	28	28	26	22	03



# 数据的运算

- 数据的运算

- 定义在逻辑结构上的一系列操作以及这些操作在存储结构上的实现；**数据的运算是定义在逻辑结构上的，而具体的实现是基于存储结构。**

- 常用的运算：检索、插入、删除、定位、修改、排序等；只是在抽象的数据上所施加的一系列抽象的操作。

- 所谓**抽象的操作**，是指我们只知道这些操作是“做什么”，而无须考虑“如何做”。只有确定了存储结构之后，才考虑如何具体实现这些运算。

本课程中讨论的各种数据结构皆按照三个方面进行：

- 逻辑定义（逻辑结构）

- 存储结构及其各种运算的实现





# 数据结构中涉及的主要结构

**线性表：**线性表中各元素之间是一种简单的“线性”关系。

**顺序表和链表：**是两种常用的实现线性表的数据结构。

**字符串：**字符串也是一种特殊的线性结构，它以字符为元素。

**堆栈：**堆栈元素的存入和取出按照后进先出原则，最先取出的总是在此之前最后放进去的那个元素；

**队列：**而队列实现先进先出的原则，最先到达的元素也最先离开队列。



**树与二叉树：**树和二叉树都属“树形结构”，在逻辑上表示了结点的层次关系。

**字典：**字典是一种二元组的集合，每个二元组包含着一个关键码和一个值。抽象地看，一个字典就是由关键码集合到值集合的一个映射。按关键码进行检索是字典中最常用的操作。

➤ **静态字典：**有些字典一经建立就基本固定不变，主要的操作就是字典元素的检索

➤ **动态字典：**经常需要改动的字典称为“动态字典”

**图：**包括一个结点集合和一个边集合，边集合中每条边联系着两个结点。

**实际应用：**公路网络、通信网络、不同事物间的联系等。



# 第一章 绪论

- 本课程在计算机问题求解中的作用
- 数据结构的基本概念和术语
- 算法与算法评价
- 总结



# 算法

- **算法**是对特定问题求解方法和步骤的一种描述，它是指令的一组有限序列，其中每个指令表示一个或多个操作。
- **算法+数据结构=程序**：揭示了算法与数据结构的关系
- 算法的**五个重要特性**
  - **输入**：0或多个外界输入，初始值
  - **输出**：一个或多个输出
  - **有穷性**：一个算法必须总是（对任何合法输入值）在执行有穷步之后结束，且每一步都可在有穷时间内完成；
  - **确定性**：每条指令有明确的含义，无二义性，相同的输入得到相同结果
  - **可行性**：算法必须能执行，所有的操作都可以通过已经实现了的基本运算执行有限次来实现。





# 算法的设计要求

- 解决问题：选择恰当的数据结构，设计一个算法，再进行编程实现。  
如何设计一个好的算法？
  - **正确性**：经得起一切输入数据的考验；能够正确实现预想的目标。
  - **可读性**：让人阅读得懂，便于交流。注意注释行的作用；
  - **健壮性**：输入数据错误时，进行必要的处理。不能得到莫名其妙的结果；
  - **高效率**：执行时间尽可能地短，对存储要求尽可能地少，既省时有节省空间。

综合考虑，时间、空间往往相互抵制，快速往往需要高存储要求，低存储要求往往导致执行时间效率下降。时间换空间，空间换时间…



# 描述算法的工具

- 自然语言：易理解，但不精确，易有二义性
- 数学语言：精确
- 约定的符号描述：流程图（直观清晰并且比较精确，但是不容易实现）、伪代码（较严谨且简洁，易用程序实现）
- 计算机高级语言描述：最终形式：**C、Pascal、C++、Java...**

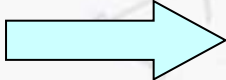


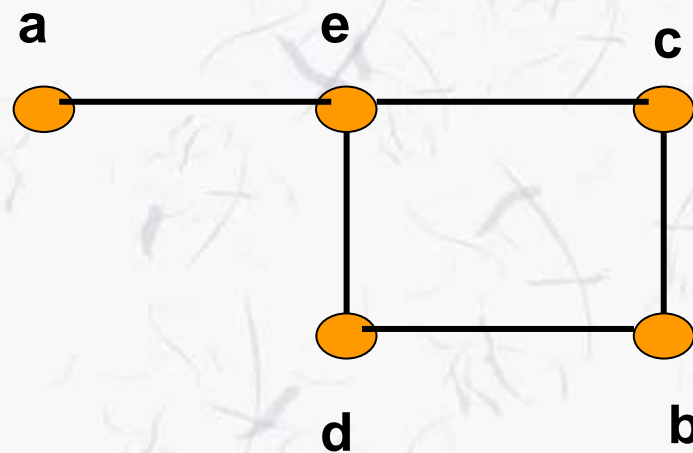
# 算法的分类

- **贪心法**[分步完成，局部最优得到整体最优]
- **分治法**[问题规模缩小，分而治之。如折半检索等]
- **动态规划法**[问题分解（缩小规模），得到各个分解结果，再自底往上求最后结果, 最佳二叉排序树]
- **回溯法**[彻底搜索，深度优先试探求得]
- **分支界限法**[彻底搜索，广度优先试探求得]
- 教材第十章详细讨论。



# 贪心法

- 贪心法思想：期望各阶段的局部最优  整体最优。





# 分治法

- 算法思想：把一个规模为 $n$ 的问题分成两个或多个较小的与原问题类型相同的子问题，通过对子问题的求解，并把子问题的解合并起来从而构造出整个问题的解，即对问题分而治之。
- 二分检索  $\log_2 n$ 次比较
- 顺序：  $(n+1) / 2$



# 动态规划法

- 算法思想：分解的子问题较多，而且相互包含，为了重用已经计算的结果，要把计算的中间结果全部保存起来，通常是自底向上进行。



# 回溯法和分枝界限法

- 算法思想：在表示问题解空间的树上进行系统搜索的方法，回溯法采用的是深度优先的策略。分枝界限法采用的是广度优先的策略。



# 算法性能的评价

- 空间效率：运行算法需要耗费的存储空间，其中主要考虑辅助空间；
    - 算法的存储空间 = 存储算法的空间 + 算法的输入、输出数据占用空间 + 程序执行过程占用的临时辅助空间
- 因此，讨论算法的空间效率主要关心执行算法需要的额外辅助空间的数量。
- 时间效率：执行算法耗费的时间；
  - 理解、阅读、编写、调试的难易等。





# 算法性能估计的两种方法

## 1. 事后统计法

- a. 必须先运行程序
- b. 所得时间统计量依赖于计算机软硬件等环境因素

## 2. 事前估算法

影响因素：

- a. 算法的策略
- b. 问题的规模 (n)
- c. 程序语言
- d. 编译程序所产生的机器代码的质量
- e. 机器执行指令的速度



# 算法的评价指标

- 空间复杂度 (**space complexity**)

当被解决问题的规模(以某种单位计算)由1增至 $n$ 时, 解该问题的算法所需占用的空间也以某种单位由 $S(1)$ 增至 $S(n)$ , 这时我们称该算法的空间代价是 $S(n)$ 。

- 时间复杂度 (**time complexity**)

当问题规模以某种单位由1增至 $n$ 时, 对应算法所耗费的时间也以某种单位由 $T(1)$ 增至 $T(n)$ , 这时我们称该算法的时间代价是 $T(n)$ 。



# 空间与时间单位

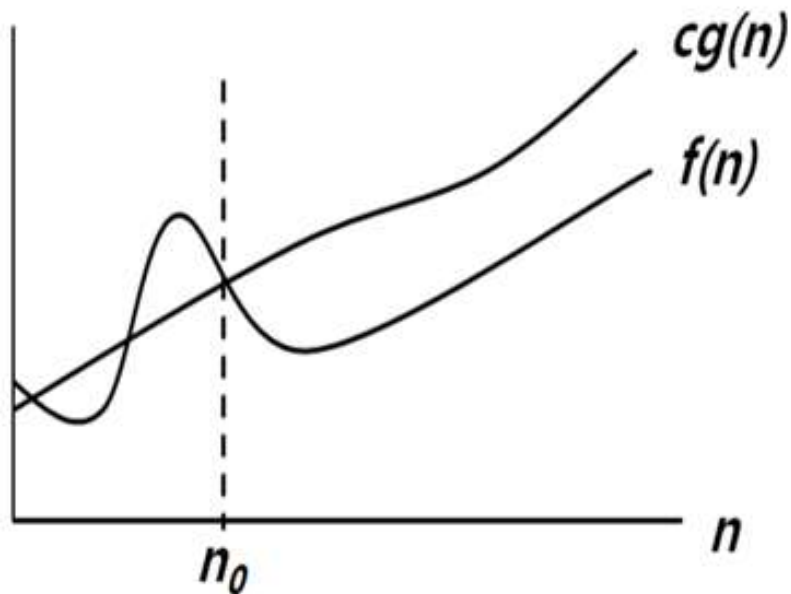
- **空间单位**：一般规定为一个简单变量(如整型、实型等)所占存储空间的大小；
- **时间单位**：一般规定为执行一个简单语句(如赋值语句、判断语句等)所用时间。**基本操作的重复执行次数。**
- **如何选择基本操作？** 循环最内层的原操作



# “大O (Big-Oh)” 表示法

- 定义1: 如果存在正的常数 $c$ 和 $n_0$ , 当问题的规模 $n \geq n_0$ 后, 该算法的时间(或空间)代价 $f(n) \leq c \cdot g(n)$ 。则称该算法的时间代价(或空间代价)为 $O(g(n))$ , 这时也称该算法的时间(或空间)代价的增长率为 $g(n)$ 。

- $f(n) = n^2 + 2n$ , 当 $n \geq 0$ 时,  
 $n^2 + 2n \leq 3n^2$ , 可得  
 $f(n) = O(n^2)$



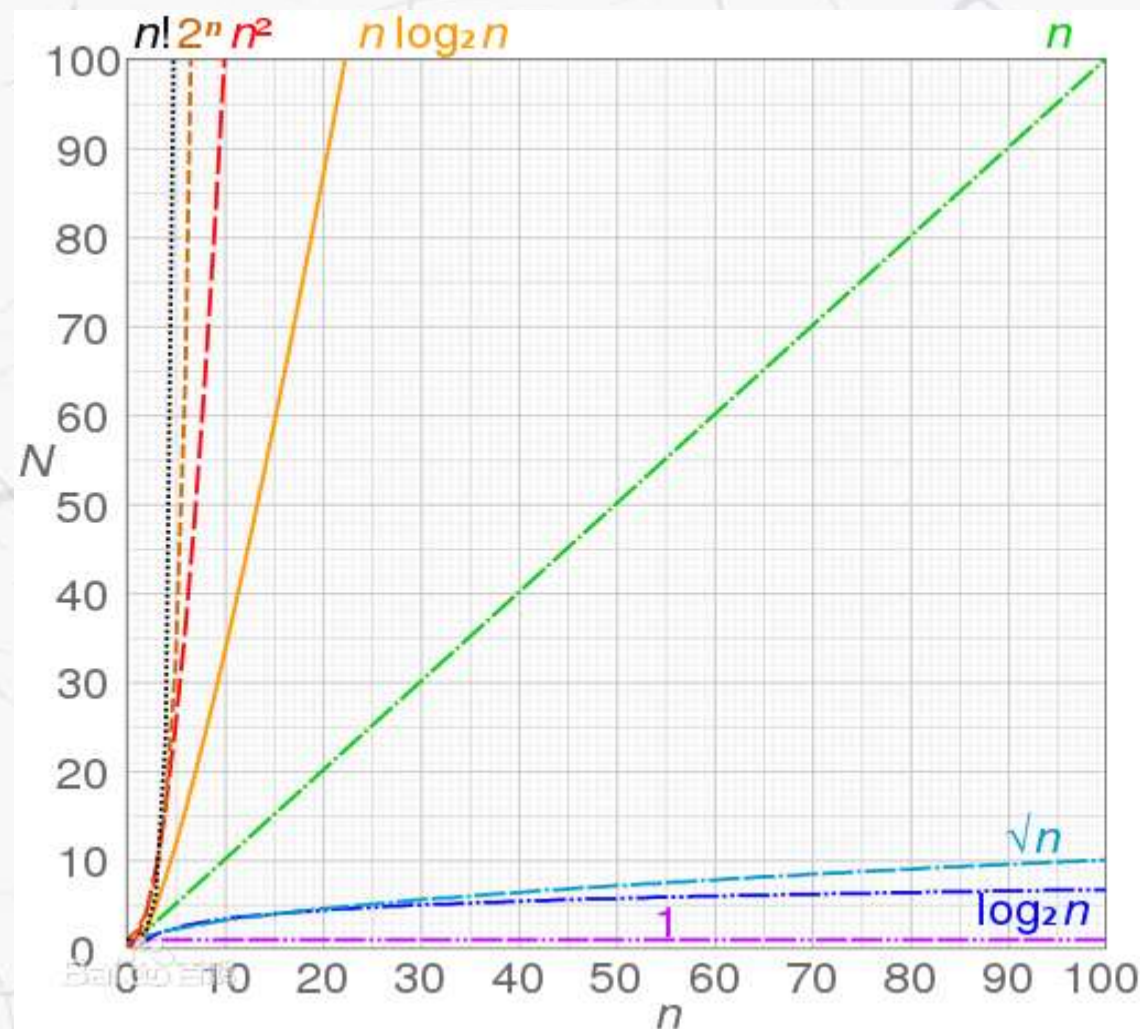


# 常见的几种时间复杂度渐进表示法

Big-Oh	说明
$O(1)$	常数时间
$O(n)$	线性时间
$O(\log_2 n)$	对数时间
$O(n^2)$	平方时间
$O(n^3)$	立方时间
$O(2^n)$	指数时间
$O(n \log_2 n)$	线性对数时间

当 $n > 16$ 时，按数量级递增排列依次为：

$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < k$ 次方阶 $O(n^k) < \text{指数阶 } O(2^n)$ 。



- 复杂度的渐进表示法:在描述算法的时间性能时, 人们只考虑宏观渐近性质, 即当输入问题规模  $n$  “充分大” 时, 观察算法复杂度随着  $n$  的“增长趋势”

❖ 常用函数增长表

函数	输入规模 $n$					
	1	2	4	8	16	32
1	1	1	1	1	1	1
$\log_2 n$	0	1	2	3	4	5
$n$	1	2	4	8	16	32
$n \log_2 n$	0	2	8	24	64	160
$n^2$	1	4	16	64	256	1024
$n^3$	1	8	64	512	4096	32768
$2^n$	2	4	16	256	65536	4294967296
$n!$	1	2	24	40320	2092278988000	$26313 \times 10^{33}$



# 大O表示法的计算规则

## 1. 加法准则

$$\begin{aligned}T(n) &= T_1(n) + T_2(n) \\ &= O(f_1(n)) + O(f_2(n)) = O(\max(f_1(n), f_2(n)))\end{aligned}$$

## 2. 乘法准则

$$\begin{aligned}T(n) &= T_1(n) \times T_2(n) \\ &= O(f_1(n)) \times O(f_2(n)) = O(f_1(n) \times f_2(n))\end{aligned}$$

- **大O表示法的作用**：主要关注复杂性的量级，而忽略量级的系数，这使我们在分析算法的复杂度时，可以忽略零星变量的存储开销和循环外个别语句的执行时间，重点分析算法的**主要代价**。



# 大O表示法的性质

- 如果函数  $f(n)$  是  $O(g(n))$  的,  $g(n)$  是  $O(h(n))$ , 那么  $f(n)$  是  $O(h(n))$  的;
- 如果函数  $f(n)$  是  $O(h(n))$  的,  $g(n)$  是  $O(h(n))$ , 那么  $f(n) + g(n)$  是  $O(h(n))$  的;
- 函数  $an^k$  是  $O(n^k)$  的,  $a$  不依赖于  $n$ ;
- 若  $f(n) = cg(n)$ , 则  $f(n)$  是  $O(g(n))$  的
- 对于任何正数  $a$  和  $b$ , 且  $b \neq 1$ , 函数  $\log_a n$  是  $O(\log_b n)$  的。即, 任何对数函数无论底数为何, 都具有相同的增长率
- 对任何正数  $a \neq 1$ , 都有  $\log_a n$  是  $O(\lg n)$  的, 其中  $\lg n = \log_2 n$
- 注: 若符号  $a$  是不依赖于  $n$  的任意常数

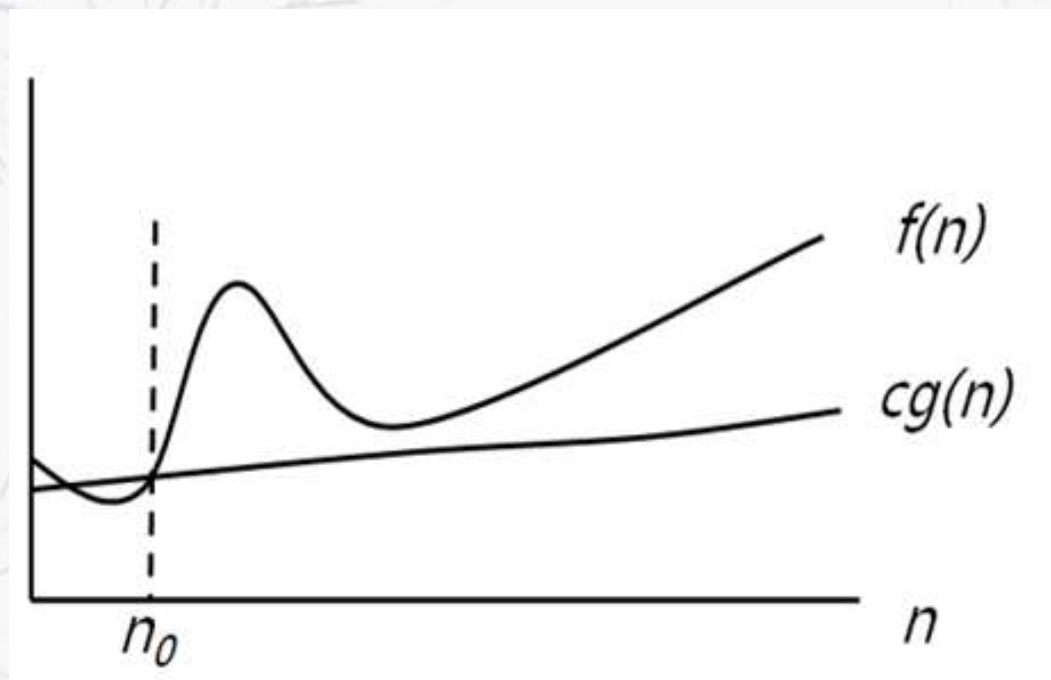




# $\Omega$ (Omega)表示法

- 定义2 : 如果存在正数 $c$ 和 $n_0$ , 使得对所有的 $n \geq n_0$ , 都有 $f(n) \geq cg(n)$

- $f(n)=n^2+2n$ , 当 $n \geq 0$ 时,  $n^2+2n \geq n^2$ , 可得 $f(n)=\Omega(n^2)$



# $\Theta$ (Theta)表示法

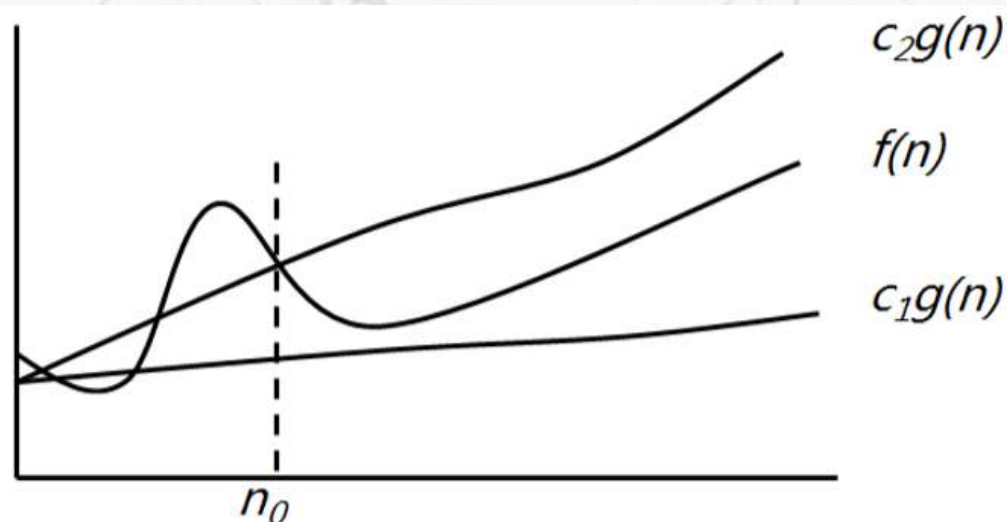
- 当上、下界相同时则可用  $\Theta$ 表示法
- 如果一个函数既在集合  $O(g(n))$  中又在集合  $\Omega(g(n))$  中，则称其为  $\Theta(g(n))$ 。即存在正常数  $c_1, c_2$ ，以及正整数  $n_0$ ，使得对于任意的正整数  $n > n_0$ ，有下列两不等式同时成立：
$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

- $f(n)=n^2+2n$ ，当  $n \geq 0$  时， $n^2+2n \leq 3n^2$ ，可得  $f(n)=O(n^2)$ 。
- 当  $n \geq 0$  时， $n^2+2n \geq n^2$ ，可得  $f(n)=\Omega(n^2)$ 。
- $f(n)=n^2+2n=\Theta(n^2)$

$$n^2+3n+2 \in \Theta(n^2)$$

$$n(n-1)/2 \in \Theta(n^2)$$

$$4n^2+5 \in \Theta(n^2)$$



# Example-1

- 例：下面是一个程序段

```
i=1; k=0;  
while(i<n)  
{  
    k=k+10*i;  
    i++;  
}
```

◆  $T(n)=2(n-1)$

∴  $T(n)=O(n)$

- ◇ 这个函数是按线性阶递增的



# Example-2

- 例：下面是一个程序段

```
a=0; b=1; ①  
for(i=2;i<=n;i++) ②  
{  
    s=a+b; ③  
    b=a; ④  
    a=s; ⑤  
}
```

- 语句①的频度为2，语句② 的频度为n，③ ④⑤的频度为n-1

◆  $T(n)=2+n+3*(n-1)=4n-1$

∴  $T(n)=O(n)$

- ◇ 这也是线性阶递增的





# Example-3

- 例：下面是一个程序段

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

- 语句①频度 $f(n)$ ，则有 $2^{f(n)} \leq n$ ，  
即 $f(n) \leq \log_2 n$ ，取最大值 $f(n) = \log_2 n$ 
  - ◆  $T(n) = 1 + \log_2 n$
- ∴  $T(n) = O(\log_2 n)$
- 这是一个按对数阶递增的函数。



# Example-4

例：下面是一个程序段

```
for (i=1 ;i<=(n-1);i++)  
{  y=y+1; ①  
  for (j=0;j<=2*n;j++)  
    x=x+1; ②  
}
```

问题规模：  $2*n$

语句①的重复执行的次数是  $f_1(n) = (n-1)$

语句②的重复执行的次数是  $f_2(n) = (n-1)*(2n+1) = 2n^2 - n - 1$

则该程序段的时间复杂度为：

$$T(n) = O(\max(f_1(n), f_2(n))) = O(n^2)$$



# Example-5

例：两个N\*N矩阵相乘的算法

```
for(i=0; i<n; ++i)
  for(j=0; j<n; ++j)
  {
    c[i][j] = 0;
    for(k=0; k<n; ++k)
      c[i][j] += a[i][k] * b[k][j];  /* 原操作 */
  }
```

问题规模：n

原操作：  $c[i][j] += a[i][k] * b[k][j];$

基本操作重复执行的次数：  $f(n) = n^3$

该算法时间度量记作  $T(n) = O(f(n)) = O(n^3)$

时间复杂度：  $O(n^3)$



- 有的情况下，算法中基本操作重复执行的次数还随问题的输入数据集的不同而不同。

例：冒泡排序法

```
void Bubble_Sort(int a[], int n)
{
    int i, change=TRUE;
    for(i=n-1; i >1 && change; --i)
    {
        change = FALSE;
        for(j = 0; j < i; j++)
            if(a[j] > a[j+1])
            {
                swap(&a[j], &a[j+1]);
                change = TRUE;
            }
    }
}
```





- 有些算法（如排序等）基本操作的执行次数除了与问题的规模 $n$ 有关外，还与输入数据有关。此时，往往用基本操作的**平均执行次数**衡量算法的时间效率。另外，也常常用算法在输入数据集最坏情况下，基本操作的**最多执行次数**作为算法的时间效率度量。
- **最坏情况下的代价**：对同样规模的问题所花费的最大代价（待排文件为反序时）
- **平均情况下的代价**：对同样规模的问题所花费的平均代价
- **最好情况下的代价**：对同样规模的问题所花费的最小代价（待排文件为正序时）



# 运行时间估算

- 例：假设CPU每秒处理 $10^6$ 个指令，对于输入规模为 $10^8$ 的问题，时间代价为 $2n^2$ 的算法要运行多长时间？
  - 操作次数为 $T(n)=T(10^8)=2 \times (10^8)^2=2 \times 10^{16}$
  - 运行时间为 $2 \times 10^{16} / 10^6 = 2 \times 10^{10}$ 秒
  - 每天有86,400秒，因此需要231,480 天（634年）



# 运行时间估算

- 例：假设CPU每秒处理 $10^6$ 个指令，对于输入规模为 $10^8$ 的问题，时间代价为 $n \log n$  的算法要运行多长时间？

- 操作次数为

$$T(n) = T(10^8) = 10^8 \times \log 10^8 = 2.66 \times 10^9$$

- 运行时间为 $2.66 \times 10^9 / 10^6 = 2.66 \times 10^3$ 秒，即44.33分钟



# 规定时间内处理问题的规模

- 设CPU每秒处理 $10^6$ 个指令，则每小时能够解决的最大问题规模
$$T(n) / 10^6 \leq 3600$$
- 对 $T(n) = 2n^2$ ,
  - 即 $2n^2 \leq 3600 \times 10^6$
  - $n \leq 42,426$
- $T(n) = n \log n$ 
  - 即 $n \log n \leq 3600 \times 10^6$
  - $n \leq 133,000,000$






# 加快硬件速度？

T(n)	处理输入规模为 $n=10^8$		1小时内解决的问题规模	
	$10^6$ 指令/秒	$10^8$ 指令/秒	$10^6$ 指令/秒	$10^8$ 指令/秒
$n\log n$	44.33 秒	0.4433秒	1.33亿	100亿
$2n^2$	634年	6.34年	42,426	424,264

- CPU每秒处理 $10^8$ 个指令（快**100**倍）
  - 处理时间降为原来的**1/100**
  - 解决问题的规模
    - 对 $2n^2$ ，规模增加**10**倍
    - 对 $n\log n$ ，规模增加**75**倍



# 第一章 绪论

- 为什么要学习数据结构
- 数据结构的基本概念和术语
- 算法与算法评价
- 总结 



# 主要的教学内容

<ul style="list-style-type: none"><li>• 概论 (2)</li><li>• 复习C语言 (0~2)</li></ul>	<ul style="list-style-type: none"><li>• 树与二叉树 (8~10)</li><li>• 图 (4~6)</li></ul>
<ul style="list-style-type: none"><li>• 线性表 (4~6)</li><li>• 字符串 (2~4)</li><li>• 栈与队列 (4~6)</li></ul>	<ul style="list-style-type: none"><li>• 字典 (检索) (5~6)</li><li>• 排序 (4~6)</li><li>• 算法设计技术 (2~4)</li></ul>

非线性结构

线性结构

- 复习课 (2) 机动 (2)



# 绪论

- **教学目的：**了解本章介绍的各种基本概念和术语，掌握算法描述和分析的方法
- **教学重点：**了解数据结构的逻辑结构、存储结构及数据的运算三方面的概念及相互关系
- **教学难点：**算法复杂度的分析方法。





# 总结

- 求解问题的主要步骤
- 求解问题的两大类
- 什么是数据结构？
- 四类逻辑结构
- 四类存储结构
- 算法的设计取决于逻辑结构，实现依赖于存储结构
- 算法的定义，算法与程序的主要区别
- 算法的描述方法
- 算法的五个重要特性和设计算法的四个基本要求
- 空间效率、时间效率的相互制约
- 时间效率的度量方法（基本操作的执行次数）
- 空间效率的度量方法（额外辅助空间的数量）



## 思考练习题：

- P26~27： 1、 2题（思考并掌握，可以不用写）。

