计算机专业导论

战德臣

哈尔滨工业大学 教授.博士生导师 教育部大学计算机课程教学指导委员会委员



第1讲 计算机、计算与计算思维

战德臣

哈尔滨工业大学 教授.博士生导师 教育部大学计算机课程教学指导委员会委员



本讲学习什么? ---计算机、计算与计算思维

战德臣

哈尔滨工业大学 教授.博士生导师 教育部大学计算机课程教学指导委员会委员

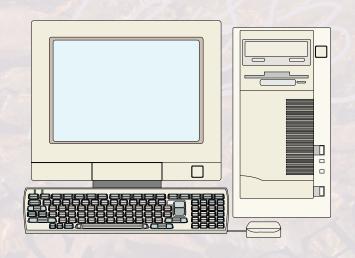


计算机、计算与计算思维 课程引语及本讲学习什么?



高度决定视野、角度改变观念、尺度把握人生

----某广告词



专业: 计算机科学与技术?

为什么要学?

战德臣

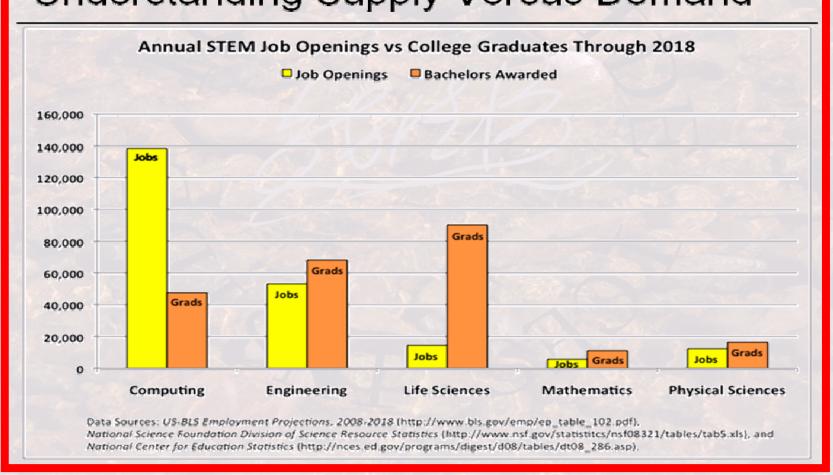
哈尔滨工业大学 教授.博士生导师 教育部大学计算机课程教学指导委员会委员



(1)为什么要学习计算机科学与技术专业?

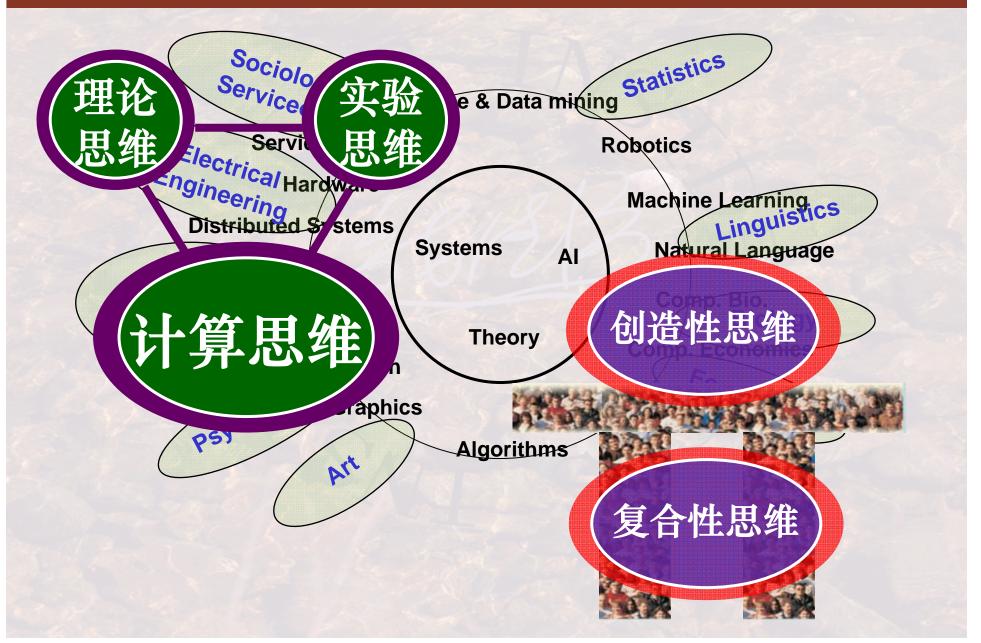


Understanding Supply Versus Demand



为什么要学? (2)为什么要学本门课程?





学什么?

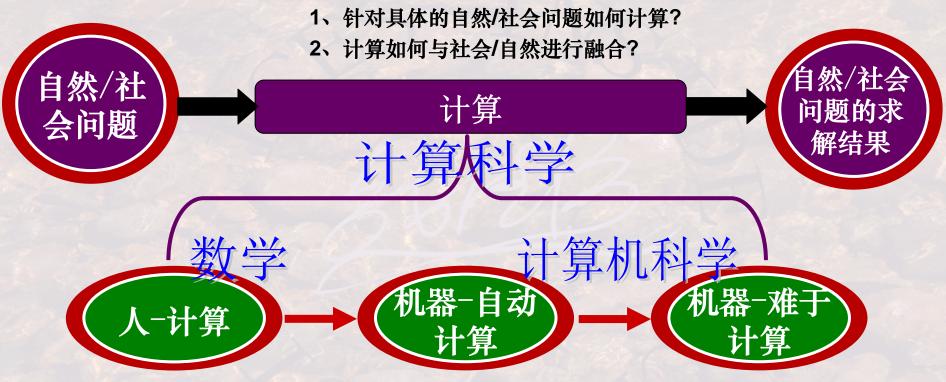
战德臣

哈尔滨工业大学 教授.博士生导师 教育部大学计算机课程教学指导委员会委员



(1)课程内容组织的基本脉络?





- 1、程序是如何被机器自动执行的:程序vs.系统?
- 2、如何编写机器可以执行的程序:语言vs.编译?
- 3、怎样构造求解问题的算法:问题→算法→程序?
- 1、可求解 vs.难求解?
- 2、如何降低计算量: 计算vs.算法?
- 3、怎样研究算法?

学什么?

(2)计算(机)科学有哪些计算思维需要学习?



计算/求解

的自然化

ΟŰ

社会/自然 的计算化

计算物理学

计算化学

Phil Trans. R. Soc. A (2008) 366, 3717-3725 Published online 31 July 2008

nal thinking and tl量品计算/ 光子计算

生物计算 计算生物学

计算语言

算法 The

移动计算

智能计算

自动化

系统

企业计算

服务网;

域网:机

抽象

模型

与数据库

数据分析与 数据仓库

数据聚集

3717

物联网: 知识网;

信息网络

广义资源

网络

战德臣

哈尔滨工业大学 教授.博士生导师 教育部大学计算机课程教学指导委员会委员







战德臣 教授



(1)知识构建次序



社会计算 计算语言学 计算广告学

计算农林学

数据化思维 企业计算 嵌入计算 网络化思约 服务计算 移动计算 之算法与程 云计算 普适计算

冯·诺依曼计算机 符号化-计算化- 计化(0和1) TELLE 拉爾拉

构造之基本 组合-抽象与 自动化产段: 迭代 构造:程序 与系统

面向某类学科 方向的计算 计算或计算·

计算生物学 医药学计算

(一說 在冀 七成 7) | | 被在草村五

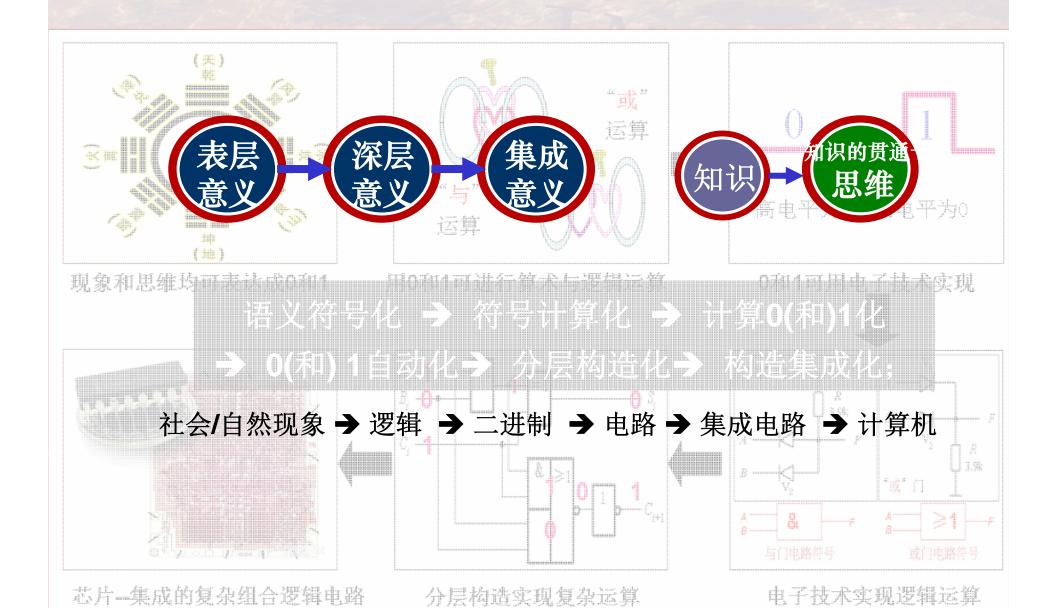
计算物理学 计算化学

计算经济学 计算金融学 计算统计学

智能计算 神经计算 感知计算

(2)贯通的知识才是思维?

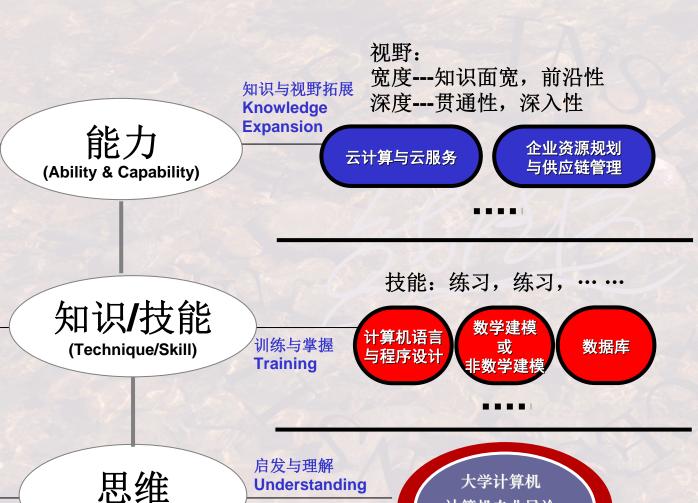




(3)知识 vs. 思维 vs. 能力?

(Thought/Thinking)





Understanding

打通知识脉络, 融贯各门课程, 内功强化基础, 外功灵活应变。

•能力----内功(贯通的脉络)

•实践----锻炼,使脉络贯通

•思维---脉络(穴位链)

•知识----穴位

思维:好奇、思考、联想、贯通

计算机专业导论

(计算思维导论)

战德臣

哈尔滨工业大学 教授.博士生导师 教育部大学计算机课程教学指导委员会委员



(1) "人"计算与"机器"计算的差别?



例如:求**ax²+bx+c=0**的根

人-求解

机器-求解

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

利用上述公式 计算得到x值

- (1)从-n到n,产 生x的每一个整 数值;
- (2)将其依次代 入到方程中计 算;
- (3)如果其值使 方程式成立, 则即为其解; 否则不是

人进行计算:

- ●一条规则可能很复杂,但计算 量却可能很小
- ●人需要知道具体的计算规则
- ●特定规则,只能求:

$$a_1 x^2 + a_2 x = c$$

机器-自动计算:

- ●每条规则可能很简单,但计算 量却很大
- ●机器也可以采用人所使用的 计算规则
- ●一般性的规则,可以求任意: $a_1X_1^{b_1}+a_2X_2^{b_2}+...+a_nX_n^{b_n}=C$

(2)基本计算规则 vs. 程序 vs. 机器?



_	n	n²	一阶差分 α _n = n²-(n	-1)²	二阶差分 $\beta_n = \alpha_n$ -		
	0	0				X ² +2x+3	
	1	1	1			x=0, 3	
	2	4	3		2	x=1, 6,	3
	3	9	5		2	x=2, 11	5
	4	16	7		· 2	x=3, 18	7
						x=4, 27	9
	5	25	9	}	2	x=5, 38	11
						x=6, 51	13
	(n+	$1)^2 =$	$n^2 + \alpha_n$	+ [3 _n	●初始值不一	·样,i

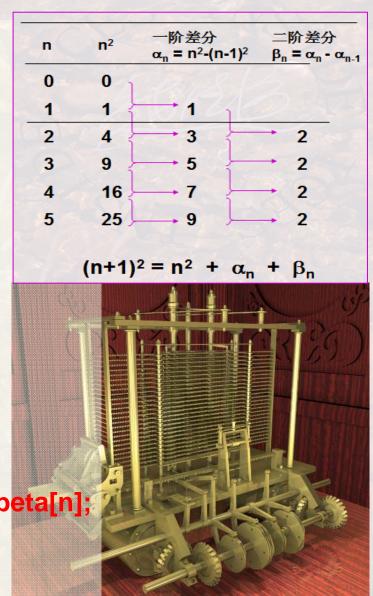
●初始值不一样,计算的 多项式也是不一样的;

(2)基本计算规则 vs. 程序 vs. 机器?



战德臣 教授

```
/*类C语言表达的计算规则—程序
Main()
 int k, n, square[], alpha[], beta[];
 input k;
                  输入不同的初
 square[0]=0;
                  始值便可计算
 square[1]=1;
                  不同的一元二
 square[2]=4;
                  次多项式的值
 alpha[1] = 1;
 for n=2 to k-1
    alpha[n] = square[n] - square[n-1];
    beta[n] = alpha[n] - alpha[n-1];
    square[n+1] = square[n] + alpha[n] + beta[n];
 output square[k];
```







```
/*类C语言表达的计算规则—程序
Main()
 int k, n, square_nminus1,square_n, alpha_nminus1, alpha_n, beta_n;
 input k;
 square_nminus1=1; square_n=4; alpha_nminus=1;
 for n=2 to k-1
    alpha_n = square_n - square_nminus1;
    beta_n = alpha_n - alpha_nminus1;
    square_nplus1 = square_n + alpha_n + beta_n;
    square_nminus1 = square_n;
    square_n = square_nplus1;
    alpha_nminus1 = alpha_n;
 output square_n;
```

alpha_nminus1 = alpha_n;

output square_n;





```
/*类C语言表达的计算规则—程序
Main()
 int k, n, square_nminus1, square_n, alpha_nminus1, alpha_n, beta_n;
 input k;
 square_nminus1=1; square_n=4; alpha_nminus=1;
 for n=2 to k-1
    alpha_n = square_n - square_nminus1;
    beta_n = alpha_n - alpha_nminus1;
    square_nplus1 = square_n + alpha_n + beta_n;
    square_nminus1 = square_n;
    square_n = square_nplus1;
```

n	square_ nplus1	square_ n	square_ nminus1	alpha_n	alpha_ nminus1	beta_n
2	9	4	1	3	1	2
3	16	9	4	5	3	2
4	25	16	9	7	5	2
5	36	25	16	9	7	2
		36	25		9	





```
/*类C语言表达的计算规则—程序
Main()
 int k, n, square_nminus1, square_n, alpha_nminus1, alpha_n, beta_n;
 input k;
 square_nminus1=1; square_n=4; alpha_nminus=1;
 for n=2 to k-1
                                                             square
                                                                                                    alpha_
                                                square
                                                                          square_
                                                                                       alpha_n
                                                                                                                 beta n
                                                nplus1
                                                                          nminus1
                                                                                                    nminus1
                                                             n
   alpha_n = square_n - square_nminus1;
                                            2
                                                   9
                                                               4
   beta_n = alpha_n - alpha_nminus1;
                                                               9
                                            3
                                                  16
                                                                                           5
                                                                                                       3
   square_nplus1 = square_n + alpha_n + beta_n;
   square_nminus1 = square_n;
                                                                                                       5
                                                  25
                                                              16
                                                                              9
   square_n = square_nplus1;
                                            5
                                                  36
                                                              25
                                                                             16
                                                                                                                     2
                                                                                           9
   alpha_nminus1 = alpha_n;
                                                              36
                                                                             25
                                                                                                       9
 output square_n;
```

机器自动计算需要解决的问题?

战德臣

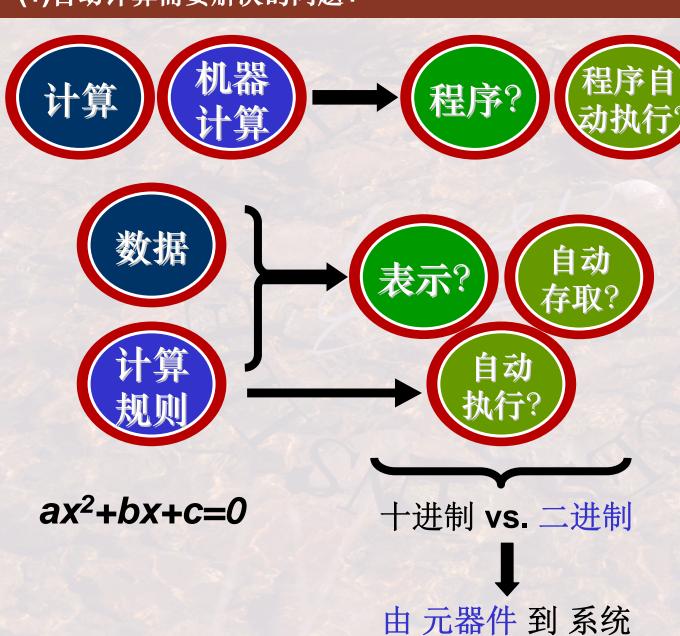
哈尔滨工业大学 教授.博士生导师 教育部大学计算机课程教学指导委员会委员

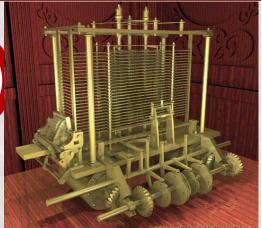


自动计算需要解决的问题?

(1)自动计算需要解决的问题?











自动计算需要解决的问题?





自动计算中的元器件的发展

◆从表示-自动存储-自动执行的角度



超大规模集成 电路(VLSI)



集成电路:可自动实现一定变换的元件



晶体管

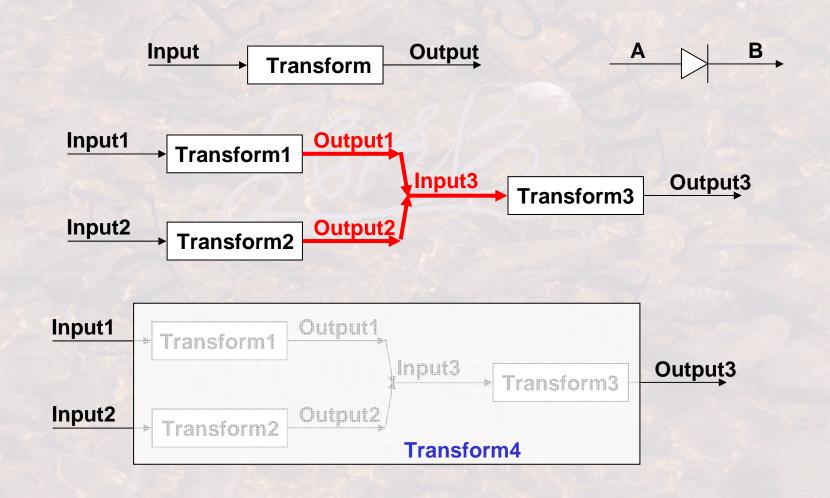
电子管:可自动控制0和1变化的元件

自动计算需要解决的问题?



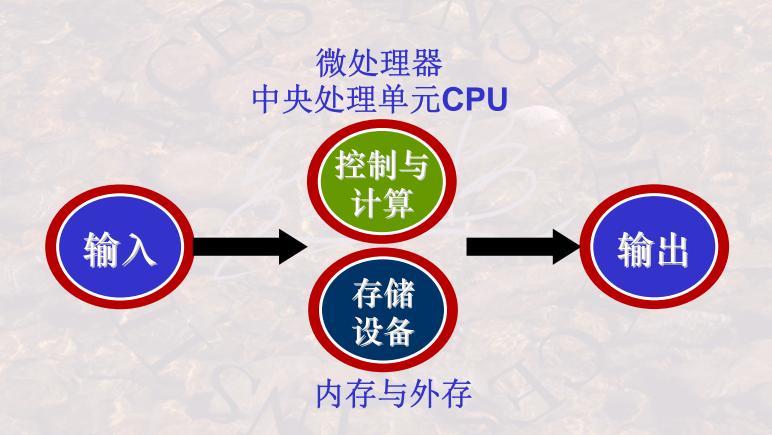
(3) 元器件发展的轨迹是怎样的,每个阶段要解决什么问题?

自动计算中的元器件的发展启示



自动计算需要解决的问题? (4)计算 vs. 计算系统?





自动计算需要解决的问题? (5)计算系统?



计算机系统的发展

发展水平 (64位; >10G; >10亿颗) 纳米存储/量子存储 感知输入 3D显示器 (接触式感知输入 (64位; 10G; 3亿颗) 固為硬盘 非接触式感知输入) **USB** Removable disk 数字显示器 (液晶、等离子技术) **FlashRAM** (32位; 1G;2800万颗) 鼠标输入 光盘存储(CD-ROM, (点输入) CRT: 数字光栅扫描显示器 CD R/W, DVD) (基于内存的显示) (32位; 60M;300万颗) 磁盘存储(便盘与软盘) 键盘输入 CRT: 字符发生器 (16位; 20M;10万颗) (符号输入) (向量式模拟显示器) 半导体存储(ROM/RAM) 磁带/磁芯/磁鼓存储 (8位; 5M;1万颗) 穿孔纸带 CRT: 阴极射线管 (0/1輸入) (模拟显示器: 黑白与彩色) 汞延迟线 (字长;主频;晶体管数目) 输出设备 类别 微处理器 输入设备 存储设备 (显示器)