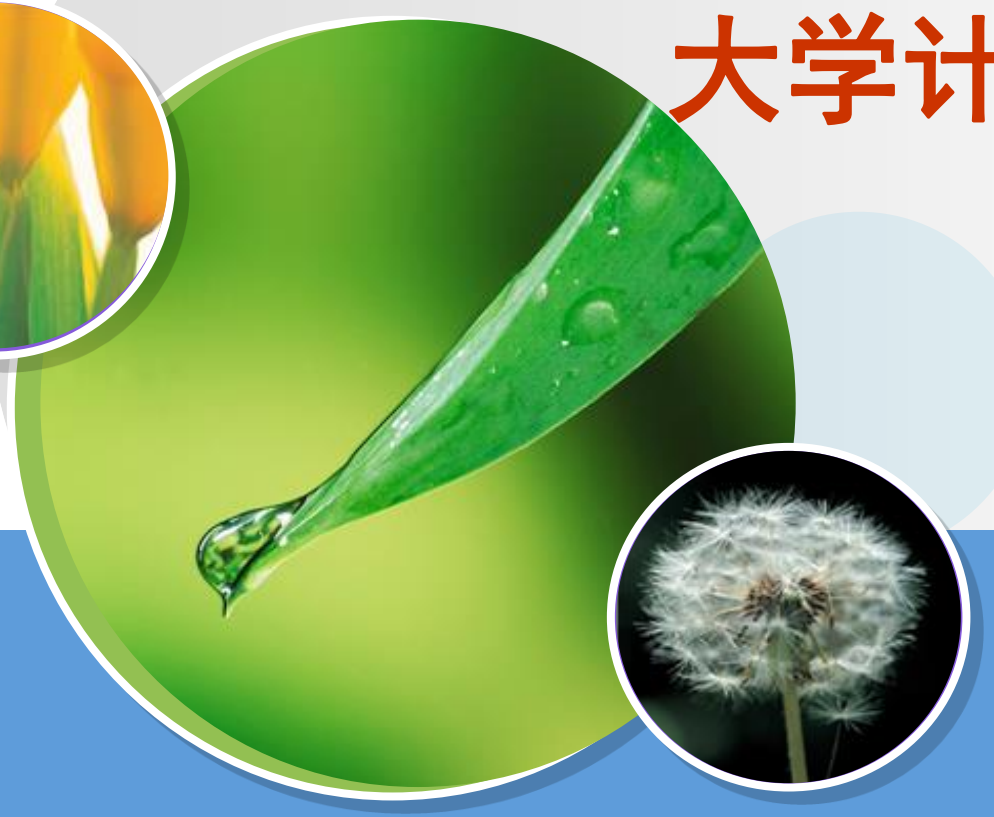
预习课件

北京航空航天大學
BEIHANG UNIVERSITY

沙河高校联盟共享课程

大学计算机基础（Python编程）



北京航空航天大学

第1章 计算思维与信息在计算机中的表示

共1学时

1.1 信息和计算的概念

1.2 计算思维定义与主要方法

1.3 信息在计算机中的表示（部分自学）

预习任务

1、请认真阅读**《课程大纲》**，了解教学内容、基本要求、考核方式等。

2、请认真阅读**《教学日历》**，了解教学安排、学习要求等。

3、自学中国大学MOOC上北航《大学计算机基础》MOOC的**课程导学**、**第1讲、第2讲**视频和课件，及时完成**单元测验**和**课堂讨论**。

4、预习**教材**《面向计算思维的大学计算机基础》**第1章**“计算思维与计算机模型”中**1.1、1.2**和**1.4节**。

5、预习课件“**2022春沙河联盟课《大学计算机基础》 -课程简介**”。

6、预习课件“**2022春沙河联盟课《大学计算机基础》-第1章-计算思维与信息在计算机中的表示【预习用】**”。

本章重点和难点

重点

- 理解什么是计算
- 把握计算思维的本质，了解其主要思维方法
- 了解计算机中的数据及其单位
- 掌握进制转换方法

难点

- 理解计算思维思想
- 掌握不同进制之间的转换方法

需要思考的若干问题

- **请带着以下问题学习本章内容**

(1) 计算机科学中的信息（Information）指什么?

(2) 什么是计算?

(3) 什么是计算思维？每个人都必须具备计算思维吗？其本质是什么?

(4) 在计算机中数据是如何表示的?

1.1 信息和计算的概念

- 美国数学家、信息论创始人**克劳德·艾尔伍德·香农**（Claude Elwood Shannon，1916-2001）：“**信息是用来消除随机不确定性的东西**”
- **信息是对客观世界中各种事物的运动状态和变化的反映，是客观事物之间相互联系和相互作用的表征，表现的是客观事物运动状态和变化的实质内容**

计算机科学中的信息

问题1：计算机科学中的信息指什么?

- **信息是能够用计算机处理的有意义的内容或消息**，它们以数据的形式出现，**如：文本（数字、字符）、声音、图形、图像、视频等**
- 数据是信息的载体
- 在现实世界中，**一切信息都可以用0和1的组合来表示**

计算的诞生

- 计算机出现之前，人们从事科学研究，大多采用**理论研究**或进行**实验**
- 计算机的问世使得科学研究又产生了新的方法——**计算**
  - **计算是指“数据”在运算符的操作下，按“规则”进行的数据变换**
  - **计算**最初指“**数值计算**”，即**有效使用数字计算机求数学问题近似解的方法与过程，以及由相关理论构成的学科**
  - 应用**计算机**处理科学研究和工程技术中**数值计算**的方法称为**计算科学**（或**科学计算**）

什么是计算？

问题2：什么是计算?

- **计算是一种符号串的转换：是将输入符号串转换为输出符号串的过程**
- **简单计算**：“数据”在运算符的操作下，按“规则”进行的数据变换
  - 如4+5=9，4×5=20
  - 学习和训练运算符的“**规则**”及其组合应用
- **复杂计算**：f(x)，函数
  - 如$\mathrm{f(x)} = \mathrm{a}x^2 + bx + \mathrm{c}$，求得其零点$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
  - 学习**计算规则**及其**简化**计算方法，应用规则求解问题

复杂计算

- 有些**复杂计算**：人知道规则但无法得到计算结果，怎么办？
  - **方法一**：研究其简化的等效计算方法（数学），使人可以计算
  - **方法二**：设计简单的规则，让机械代替人按照“规则”**自动**计算

【**例**】判断丢番图方程$a_1{x_1}^{b1} + a_2{x_2}^{b2} + \cdots + a_n{x_n}^{bn} = c$是否有整数解？

【课堂讨论1】

【课堂讨论1】

(1) 丢番图方程可称为什么方程?

(2) 可以用什么方法解丢番图方程?

请在公共聊天区写出你的答案

【课堂讨论1】答案

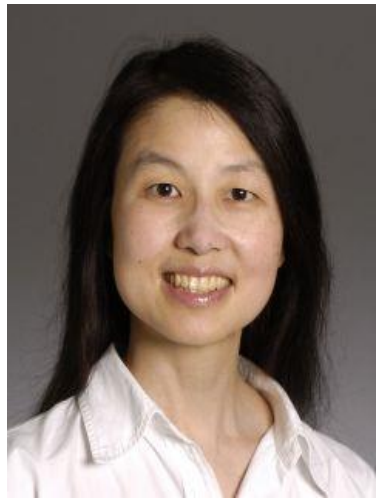
(1) 丢番图方程可称为**不定方程**——变量的个数大于方程的个数的方程

(2) 可以采用**枚举**的方法解丢番图方程

——列举可能解：逐一列举x1~xn的一组数，看能否使方程成立；如果成立，则采纳，否则丢弃

1.2 计算思维定义与主要方法

问题3：什么是计算思维？每个人都必须具备计算思维吗？其本质是什么？



- 2006年，美国卡内基·梅隆大学**周以真**（Jeannette M. Wing）教授在《Communications of the ACM》上发表论文“**Computational Thinking**”，首次明确提出计算思维的概念
  - **计算思维**（Computational Thinking，CT）是运用**计算机科学**的基础概念进行**问题求解**、**系统设计**、以及**人类行为理解**等涵盖计算机科学之广度的一系列**思维活动**

计算思维仅属于计算机科学家？

- **每个人的基本技能**
  - 计算思维面向所有的人，所有地方
  - 当计算思维真正融入人类活动的整体时，它作为一个问题解决的有效**工具**，**人人都应当掌握**，**处处都会被使用**。

图灵奖得主**Edsger Dijkstra**：***“我们所使用的工具影响着我们的思维方式和思维习惯，从而也将深刻地影响着我们的思维能力。”*** ——**工具影响思维**

周以真：*“计算思维不仅仅是计算机科学家应具备，而是每个人都应该具备的基本能力。”*

伟大的计算原理

- 2003年，计算机科学家、美国ACM 前主席**Peter J. Denning**（彼得 J.丹宁）发表论文“**Great Principles of Computing（伟大的计算原理）**”
  - 计算原理可以被归为**7个**类别，每个类别都从一个独特的视角去看待计算本身
  - **计算、通信、协作、记忆、自动化、评估**和**设计**

7个计算原理

- **计算**（Computation）是**执行**一个**算法**的过程
- **通信**（Communication）是指信息从一个过程或者对象**传输**到另一个过程或者对象
- **协作**（Coordination）是为确保多方参与的计算过程（如多人会话）最终能够得到确切的结论而对整个过程中各**步骤序列**先后顺序进行的**时序控制**
- **记忆**（Recollection）是指通过实现有效搜索数据的方法或者执行其他操作对数据进行**编码**和**组织**
- **自动化**（Automation）是计算在**物理系统自身运作**过程中的表现形式（镜像）
- **评估**（Evaluation）是对数据进行**统计**分析、**数值**分析或者**实验**分析
- **设计**（Design）是利用学科中的抽象、模块化、聚合和分解等方法对一个**系统**、**程序**或者**对象**等进行组织

计算思维的表述体系

分类	关注点	核心概念
计算	什么能计算，什么不能计算	计算的表示、表示的转换、状态和状态转换、按空间排序、按时间排序；可计算性、计算的复杂性
抽象	关注对象的本质特征	概念模型与形式模型、抽象层次、抽象结构、虚拟机
自动化	信息处理的算法发现	形式化方法、程序、算法、迭代、递归、搜索、推理；强人工智能、弱人工智能
设计	可靠和可信系统的构建	模块化、信息隐藏与封装、层次聚集；一致性和完备性、重用、安全性、可靠性、折中与结论
通信	不同位置间的可靠信息移动	信息及其表示、信息量、编码与解码、信息压缩、信息加密；校验与纠错
协作	多个自主计算机的有效使用	同步、并发、死锁、仲裁；事件以及处理、流和共享依赖，协同策略与机制
记忆	媒体信息的表示、存储和恢复	存储体系、对象与存储的动态绑定、层次命名、检索与索引；局部性与缓存、抖动（trashing）
评估	复杂系统的性能评价	模型方法、模拟方法、基准测试；可视化建模与仿真、预测与评价、服务网络模型；负载、吞吐率、反应时间、瓶颈、容量规划

计算思维的本质

计算思维的本质

计算思维最根本的内容是抽象（Abstraction）与自动化（Automation）

抽象

- 是指将要求解的问题**形式化**地表示为计算机所能理解的**符号模型**，进而用程序语言描述该模型

自动化

- 是指计算机**自动**执行程序，实现问题求解

计算思维的主要方法

（1）通过**约简、嵌入、转化和仿真**等方法，把一个看来困难的问题重新阐释成一个我们知道问题怎样解决的方法。

- **【例】计算机为什么能执行加、减、乘、除各种运算?**
  - ✓ 在机器数表示中引入**补码**，则将减法转化为加法运算；乘法用移位和加法；除法通过若干次“加、减、移位”循环实现，使得各种**算术运算**均以**加法器**为基础，从而**简化CPU的设计**

原码、反码和补码

原码、反码和补码

- **原码**是一种计算机对数值数据的二进制定点表示方法。其**最高位**为**符号位**（0表示正数，1表示负数），其余位表示数值的绝对值大小

  $[13]_{原} = 0\ 0001101$，$[-7]_{原} = 1\ 0000111$

- **反码**的表示方法：正数的反码与其原码相同；**负数**的**反码**是将原码的**数值部分各位取反**（0变1，1变0），**符号位**为**1**

  $[13]_{反} = 0\ 0001101$，$[-7]_{反} = 1\ 1111000$

- **补码**的表示方法：正数的补码与其原码相同；**负数**的**补码**由在其**反码**的最低有效位上**加1**获得

  $[13]_{补} = 0\ 0001101$，$[-7]_{补} = 1\ 1111001$

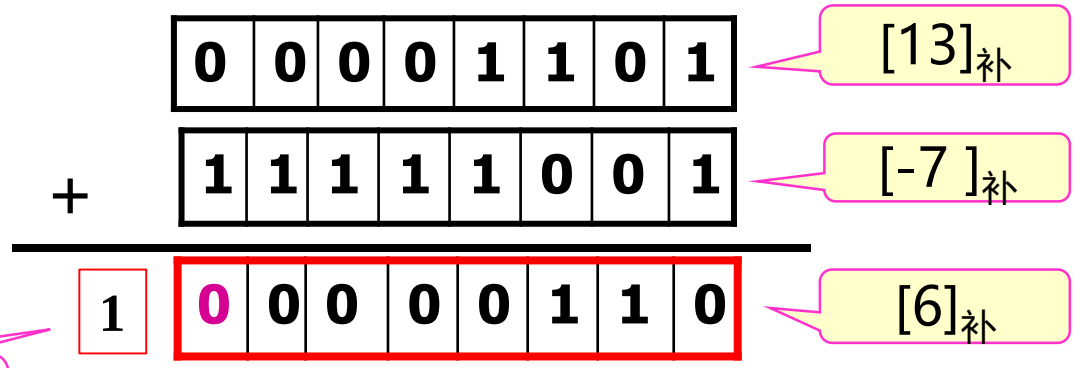
【例1.1】补码加法

【例1.1】采用8位补码，计算13-7。

解：$[13]_{原}$ = 0 0001101，$[13]_{反}$ = 0 0001101，$[13]_{补}$ = 0 0001101

$[-7]_{原}$ = 1 0000111，$[-7]_{反}$ = 1 1111000，$[-7]_{补}$ = 1 1111001

根据补码的运算规则（补码的**符号位**和数值一起**参加运算**；若符号位产生了进位，则**将进位舍弃**），有：



即：$[13]_{补}$ + $[-7]_{补}$ = 0 0001101 +1 1111001 = 1 00000110

舍弃向最高位的进位“1”，结果为**00000110**，即为6的补码，故**13-7 = 6**

计算思维的主要方法（续1）

（2）**递归思维**；**并行处理**；把代码译成数据（**译码**）或把数据译成代码（**编码**）；多维分析归纳的**类型检查**方法。

- **递归**是指**在函数的定义中调用函数自身的方法**
- **【例1.2】求阶乘F(n)=n! =n*(n-1)*(n-2)*……*1**

✓ **将数学表达式n*(n-1)*(n-2)*……1写成递归表达式（递归的定义）**

$$n! = \begin{cases} 1 & \text{当} n = 0 \text{或} 1 \text{时} \\ n * (n-1)! & \text{当} n > 1 \text{时} \end{cases}$$

递归基础——递归计算的起点

递归步骤——递归计算的步骤

阶乘的递归函数定义和调用

✓ 定义递归函数

$$f(n)=\begin{cases}1 & \text{当}n=0\text{或}1\text{时}\\ n*f(n-1) & \text{当}n>1\text{时}\end{cases}$$

例1.2-factorial_recursion.py

```
#定义递归函数
def factorial(n):
    if n == 0 or n==1:
        return 1
    elif n>1:
        return n* factorial(n-1)
    else:
        print('请输入正整数')

x=int(input ('Enter n:'))
fac_result=factorial(x)

print ('factorial result =', fac_result)
```

调用递归函数

```
>>>
Enter n:3
factorial result = 6
>>> ====================
>>>
Enter n:5
factorial result = 120
```

计算思维的主要方法（续2）

（3）是一种采用**抽象**和**分解**来控制庞杂的任务或进行巨大复杂系统设计的方法，是基于**关注点分离**的方法。

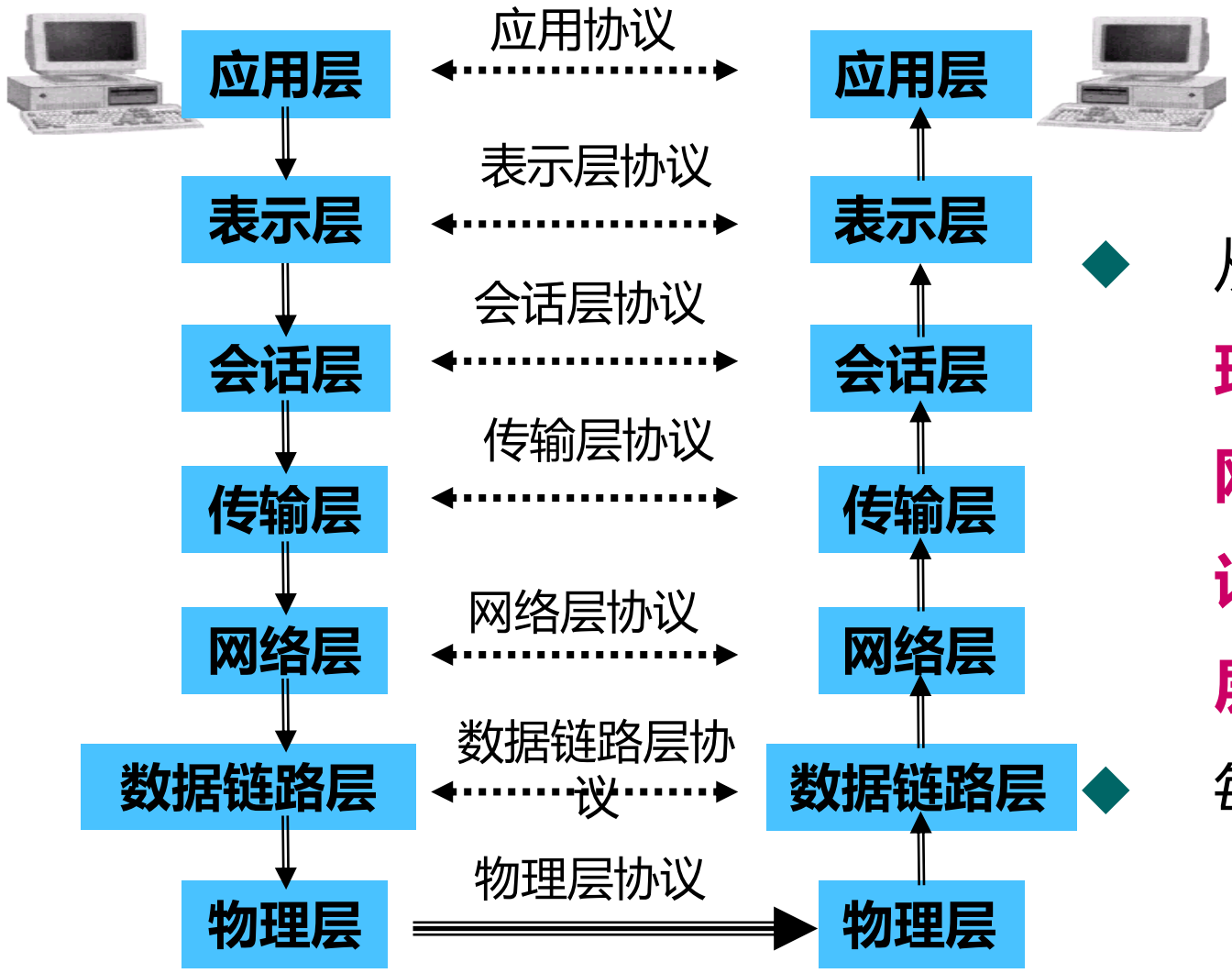
- **【例】OSI参考模型**定义了**计算机网络**的7层结构
  - ✓ 采用**分层结构**技术，将网络结构分为**7层**，每一层的**功能相对独立**
  - ✓ 定义了每一层的**协议**：**某一层与远方的一个对等层通信所使用的一套规则和约定**
  - ✓ 定义了**每一层实体为相邻的上一层实体提供的通信功能**——**服务**
- 层次化结构，使复杂的系统简单化，蕴含了**抽象、分解、关注点分离、建模**等计算思维方法

参见教材“1.6.2 计算机网络访问过程蕴含的计算思维”

OSI 参考模型的7层框架

应用层
为应用程序提供接口和网络服务

数据流层
负责网络通信



- 从下至上依次为：**物理层、数据链路层、网络层、传输层、会话层、表示层、应用层**
- 每层完成独立的功能

计算思维的主要方法（续3）

- **建模**
- **冗余、容错、纠错**
- **启发式推理寻求解答**
- 在**时间和空间**之间，在**处理能力和存储容量**之间进行**折衷**
- ……

现代计算机的**三级存储系统**：Cache、主存和辅存

参见教材“1.5.1 计算机硬件系统”

计算思维的核心方法

- 利用计算机自动求解一个问题的过程

抽象—建模—数据组织—算法—编程实现

计算思维的核心方法

本课程重点内容

1.3 信息在计算机中的表示

- 1.3.1 计算机中的数据及其单位
- 1.3.2 进位计数制
- 1.3.3 不同进制间的转换方法

1.3.1 计算机中的数据及其单位

- 计算机内部，一切数据均是用**0**和**1**的组合（**二进制**）来表示
- 计算机中的数据分为**数值型**数据和**非数值型**数据

数值型数据

- **是表示数量、可以进行数值运算的数据类型**
  - ✓ 如年龄18岁，身高1.75m，速度12秒/100m，高考成绩满分750分
  - ✓ 18在计算机内存储时表示为**0001 0010**
  - ✓ 1.75在计算机内存储时可采用**浮点数**表示

非数值型数据

非数值型数据

- **是表示字符（英文字母、汉字、标点符号等）、声音、图形、图像、视频等信息，不能进行数值运算的数据类型**
  - ✓ 如汉字“**中**”在计算机中存储时表示为**1101 0110 1101 0000——汉字机内码**
  - ✓ 在程序中或文档中为方便书写，一般写成十六进制$D6D0_H$

数据表示的单位

位

- 在计算机中，往往需要用多位二进制数码表示一个数，其中每一个数码称为1**位**（bit，比特）
  - **位**是计算机中数据的**最小单位**
  - 如正整数77表示为100 1101，其长度为7bit

字节

- 8个二进制位称为1个**字节**（Byte），**1Byte = 8bit**
  - **字节**是现代计算机中数据存储和处理的**基本单位**

存储容量的单位

- **存储器的容量统一以字节（Byte，B）为单位**

字节	1 Byte = 8 bit
千字节	1 KB = 1024 B = 2^{10}B
兆字节	1 MB = 1024 KB = 2^{20}B
吉字节	1 GB = 1024 MB = 2^{30}B
太字节	1 TB = 1024 GB = 2^{40}B
拍字节	1 PB = 1024 TB = 2^{50}B
艾字节	1 EB = 1024 PB = 2^{60}B

思考：相邻的两个存储单位之间是怎样的倍数关系?

- **目前机械硬盘容量最大已达到16TB 、20TB**
- **单条内存容量最大达到32GB**

1.3.2 进位计数制

- **进位计数制**（简称**进位制**，**进制**或**数制**）：**是用数码和带有权值的数位来表示有大小关系的数值型信息的表示方法**
  - 如果采用R个基本符号（例如：0，1，2，…，R−1）表示数值，则称**R进制**
  - 日常生活中：**十进制**，**十二进制**、**六十进制**等
  - 计算机内部：**二进制**
  - 编程或撰写文档：**八进制**或**十六进制**

进位制中的三个要素

任意一个**R进制**数D均可展开为：

$$(D)_R = \sum_{i=-m}^{n-1} k_i \times R^i$$

数码：第i位的系数

位权

- **数码、基数和位权是进位制中的三个要素**
  - **数码**：**数制中表示基本数值大小的基本符号**
  - **基数**（Radix）：**某种进位制所包含的数字符号的个数**
  - **位权**：**数制中某一位上的R^i所表示数值的大小（数码所处位置的价值）**

    称为第i位的**位权**（权值）
    - ✓ **若某进制的基数为R，则第i位的位权为R^i**
    - ✓ 例如，十进制数“795”，其个位数5的位权是10^0，十位数9的位权是10^1，百位数7的位权是10^2

常用的进位制及表示方法

几种常用的进位制的表示

进位制	基数	数 码	位权	形式表示
二进制 (binary)	2	0，1	2^i	B
八进制 (octal)	8	0，1，2，3，4，5，6，7	8^i	O
十进制 (decimal)	10	0，1，2，3，4，5，6，7，8，9	10^i	D
十六进制 (hexadecimal)	16	0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F（小写也可）	16^i	H

不同进制数的表示方法

方法一

- 在数字电路和计算机中，用**括弧加进制的基数或字母下标**的方式表示不同进制的数
  - ✓ 如$(25)_{10}$、$(1101.101)_{2}$、$(37F.5B9)_{16}$
  - ✓ 或者$(25)_{D}$、$(1101.101)_{B}$、$(37F.5B9)_{H}$

方法二

- 在数的后面加上该进制英文单词的**首字母**
  - ✓ 如25**D**、1101.101**B** 、37F.5B9**H**

- 若数的后面没有任何附加的数字或字母，一般表示十进制
  - ✓ 如25

不同进制数的对照表

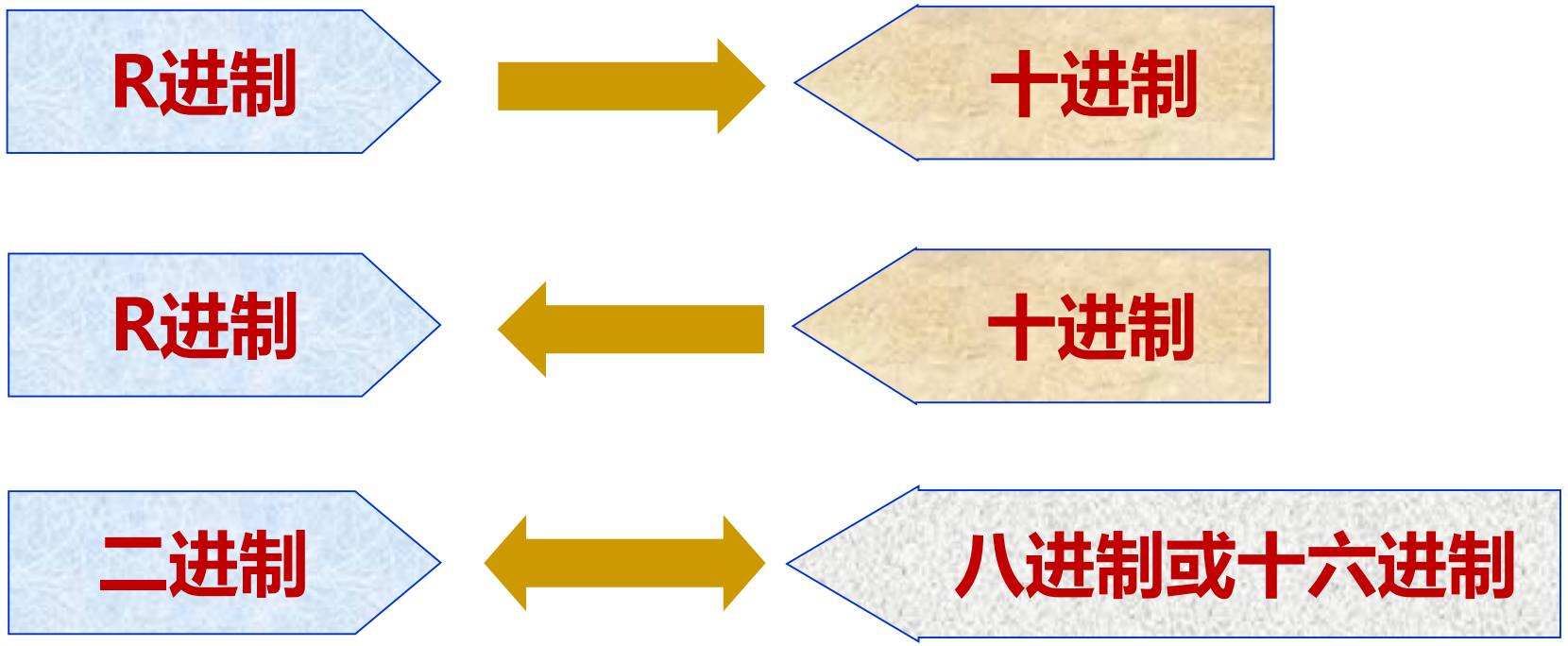
十进制	二进制	八进制	十六进制
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	**1001**	**11**	**9**
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

基数越大，使用的位数越少

- 同一个数值（例如**9**）用**不同**的**进制**表示，会表达成**不同**的**数码串**；
- 一个相同的数码串（如**11**），因其使用的**进制不同**而表示**不同**大小的**数值**

在书写数值型数据时，一定要标明其使用的进制！

1.3.3 不同进制间的转换方法



- **转换基本原则**
  - 对**整数**部分和**小数**部分分别进行转换

1、R进制数转换为十进制数

任意一个R进制数D均可展开为：

$$(D)_R = \sum_{i=-m}^{n-1} k_i \times R^i$$

方法　**通式展开法——把R进制数按权展开求和**

【例】$(1101.011)_2$=

$1\times2^3+1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}+1\times2^{-3}$

$= 8+4+0+1+0.0+0.25+0.125=(13.375)_{10}$

【例】$(1FD.6C)_{16}$=

$1\times16^2+15\times16^1+13\times16^0+6\times16^{-1}+12\times16^{-2}$

$=(509.421875)_{10}$

【课堂讨论2】

- 4位八进制数$(1527)_8$从高到低各位的位权分别是多少?
- 试写出对应的十进制数

【课堂讨论2】答案

- 4位八进制数$(1527)_8$从高到低各位的位权分别是$\mathbf{8^3}$、$\mathbf{8^2}$、$\mathbf{8^1}$、$\mathbf{8^0}$
- $\mathbf{(1527)_8} = 1\times8^3+5\times8^2+2\times8^1+7\times8^0 = \mathbf{855}$

2、十进制数转换为R进制数

方法

（1）**整数**转换用“**除基取余法**”，**直到商为零**；每次相除所得余数为对应的R进制整数的各位数码。余数从**右**到**左**排列，**首次**取得的余数排在**最右**边（结果的最**低**位）。

（2）**小数**转换用“**乘基取整法**”，直到乘积的**小数部分为零**，或**达到所要求的位数**（当小数部分永不可能为零时）。每次相乘所得整数从小数点之后自**左**往**右**排列，取有效精度，**首次**取得的整数排在**最左**边。

【例1.3】十进制数转换为二进制数

【例1.3】将$(62.625)_{10}$转换为二进制数。

【例1.3】转换过程

- **先转换整数部分**

2	62	… 余数	=0 = k_0(LSB)
2	31	… 余数	=1 = k_1
2	15	… 余数	=1 = k_2
2	7	… 余数	=1 = k_3
2	3	… 余数	=1 = k_4
2	1	… 余数	=1 = k_5(MSB)
	0		

商为0，转换结束

- **余数**按从最高位到最低位的顺序写出来

$(111110)_2$

最后取得的余数

最先取得的余数

【例1.3】转换过程（续）

- **再转换小数部分**

```
   .625
 ×   2
 ------
1 .250      进位 “1 ” = k-1  (MSB)
0 .250
 ×   2
 ------
0 .500      进位 “0” = k-2
 ×   2
 ------
1 .000      进位 “1” = k-3   (LSB)
```

小数部分为0，转换结束

- **进位**按从最高位到最低位的顺序写出来

$(.101)_2$

最先取得　最后取得

【例1.3】转换结果

$(62.625)_{10}=(111110.101)_2$

$(k_5k_4k_3k_2k_1k_0.\ k_{-1}k_{-2}k_{-3})_2$

- 将整数部分除以R得到的余数按从**最高位**到**最低位**的顺序写出来，即为转换后的R进制**整数**部分
- 将小数部分乘以N得到的向整数的进位按从**最高位**到**最低位**的顺序写出来，即为转换后的R进制**小数**部分

【课堂练习】十进制数转换为十六进制数

【课堂练习】将十进制数$(197.734375)_{10}$转换成十六进制数 。

三分钟内请在公共聊天区写出你的答案

【课堂练习】答案

- **先转换整数部分**

16 | 197 ··· 余数 = 5 = k_0(LSB)

16 | 12 ··· 余数 =12 = k_1(MSB)

0

商为0，转换结束

- **余数**按从最高位到最低位的顺序写出来

$197 = (C5)_{16}$

最后取得的余数

最先取得的余数

【课堂练习】答案（续）

- **再转换小数部分**

.734375
× 16
11.75　　进位“11 ”= k_{-1}　(MSB)

0.75
× 16
12.00　　进位“12”= k_{-2}　(LSB)

小数部分为0，转换结束

- **进位**按从最高位到最低位的顺序写出来

.734375 = $(.BC)_{16}$

最先取得　最后取得

转换结果

- **整合转换结果**

$(197.734375)_{10}=(C5.BC)_{16}$

$(k_1k_0.\ k_{-1}k_{-2})_{16}$

- 将整数部分除以R得到的余数按从**最高位**到**最低位**的顺序写出来，即为转换后的R进制**整数**部分
- 将小数部分乘以R得到的向整数的进位按从**最高位**到**最低位**的顺序写出来，即为转换后的R进制**小数**部分

3、二进制数与八进制数相互转换

自学

- 当一个数很大时，若用二进制表示，位数太多，可以用**八进制**或**十六进制**表示
- $2^3=8$，因此3位二进制数对应1位八进制数，而1位八进制数对应3位二进制数

- **二进制数转换成八进制数**
  - 以小数点为界，向左或向右，**每3位**二进制数用相应的**一位**八进制数取代
  - **注意**：**两头不足3位的二进制数先用0补足**
- **八进制数转换成二进制数**
  - 以小数点为界，向左或向右，每**一位**八进制数用相应的**3位**二进制数取代

【例1.4】二进制数转换成八进制数

自学

【例1.4】将二进制数11010011.1101101转换成八进制数。

解： $(011\ 010\ 011.110\ 110\ 100)_2=(323.664)_8$

整数高位补0

小数低位补0

$(\ 3\quad 2\quad 3\ .\ 6\quad\quad 6\quad \textcircled{4}\ )_8$

思考：为什么转换成八进制后小数部分的最后1位是4而不是1?

【例1.5】八进制数转换成二进制数

自学

【例1.5】将八进制数174.536转换成二进制数。

$$(1\ 7\ 4.5\ 3\ 6)_8$$

解： $(174.536)_8=(001\ 111\ 100.101\ 011\ 110)_2$

4、二进制数与十六进制数相互转换

自学

- 二进制数-十六进制数之间的转换方法跟二进制数-八进制数之间的转换类似
- 区别
  - 4位二进制数对应1位十六进制数
  - 3位二进制数对应1位八进制数

5、八进制数与十六进制数相互转换

自学

八进制 ⟷ 十六进制

- **思考：用什么方法最简单？以什么作为桥梁？**

方法

（1）八进制数转换为十六进制数

- 以**二进制**为桥梁，首先将**八**进制数转换为**二**进制数，再将二进制数转换为**十六**进制数

（2）十六进制数转换为八进制数

- 也以**二进制**为桥梁，先将**十六**进制数转换为**二**进制数，再将二进制数转换为**八**进制数

本章小结

北京航空航天大学

1.1 信息和计算的概念

信息

- **克劳德•艾尔伍德•香农：信息是对客观世界中各种事物的运动状态和变化的反映，是客观事物之间相互联系和相互作用的表征**
- **计算机中的信息：信息是能够用计算机处理的有意义的内容或消息，**它们以数据的形式出现，**如：文本、声音、图形、图像、视频等**

计算

- **计算是一种符号串的转换：是将输入符号串转换为输出符号串的过程**
- **简单计算：** 利用运算符，按“运算规则” **手动**进行计算
- **复杂计算**：设计算法（规则），让计算机按照“规则” **自动**计算

1.2 计算思维定义与主要方法

定义

计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动

本质

抽象与**自动化**

主要方法

约简、递归、并行处理、抽象和分解、建模、冗余、容错、纠错……

1.3 信息在计算机中的表示

信息在计算机中的表示

- 在计算机内部，一切数据（文本、声音、图形、图像、视频）都是用**二进制**表示的
- **位**：计算机中数据的**最小单位**
- **字节**：现代计算机中数据**存储**和**处理**的基本单位

1 Byte = 8 bit，1 KB = 1024 B

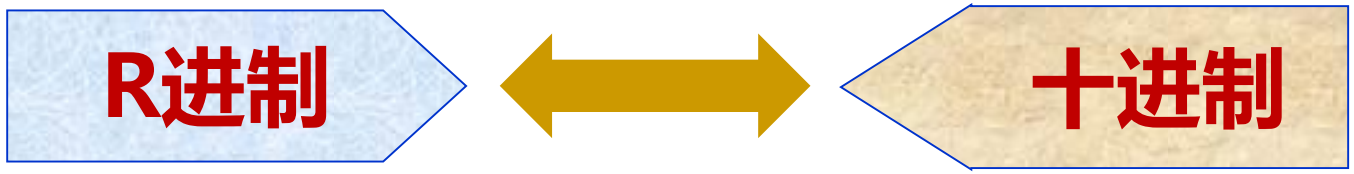
常用的存储单位

- 字节，千字节，兆字节，吉字节
- 太字节，拍字节，艾字节
- **相邻的两个存储单位之间是2^{10}的倍数关系**

进位制

- **进位制**：**是用数码和带有权值的数位来表示有大小关系的数值型信息的表示方法**
- 进位制中的**三个要素**
  - **数码**：数制中表示基本数值大小的**基本符号**
  - **基数**：某种进位制所包含的数字符号的**个数**
  - **位权**：数制中某一位上的R^i所表示**数值**的**大小**（**数码所处位置的价值**）
- 几种常用的进位制
  - **二进制，八进制**
  - **十进制，十六进制**

不同进制之间的转换方法（1/4）



1、R进制数转换为十进制数

- **通式展开法**——把R进制数按权展开求和

2、十进制数转换为R进制数

- **整数**部分的转换采用**除基取余法**，**直到商为零**
- **小数**部分的转换采用**乘基取整法**，直到乘积的**小数部分为零**，或**达到所要求的位数**

不同进制之间的转换方法（2/4）

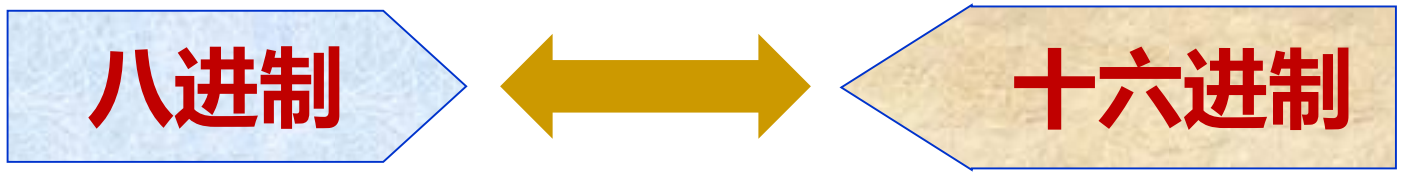
3、二进制数转换成八进制数

- 以小数点为界，向左或向右，**每3位**二进制数用相应的一位八进制数取代
- **注意**：**两头不足3位的二进制数先用0补足**

4、八进制数转换成二进制数

- 以小数点为界，向左或向右，每一位八进制数用相应的**3位**二进制数取代

不同进制之间的转换方法（3/4）



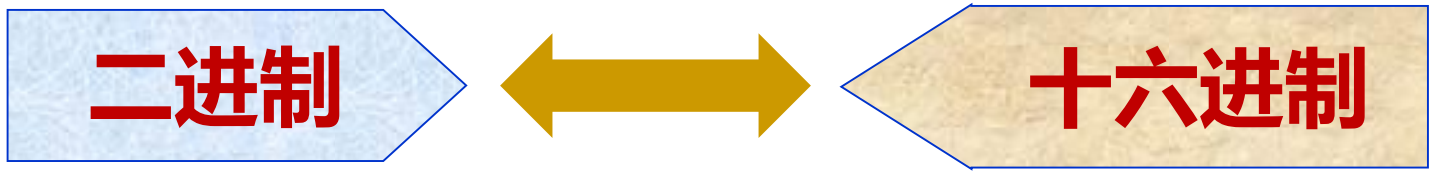
5、八进制数转换为十六进制数

- 以**二进制**为桥梁，首先将**八**进制数转换为**二**进制数，再将二进制数转换为**十六**进制数

6、十六进制数转换为八进制数

- 以**二进制**为桥梁，先将**十六**进制数转换为**二**进制数，再将二进制数转换为**八**进制数

不同进制之间的转换方法（4/4）



7、二进制数转换成十六进制数

- 以小数点为界，向左或向右，**每4位**二进制数用相应的一位十六进制数取代
- **注意**：**两头不足4位的二进制数先用0补足**

8、十六进制数转换成二进制数

- 以小数点为界，向左或向右，每一位十六进制数用相应的**4位**二进制数取代