计算机基础知识

# 计数法与进制

## 二进制

### 二进制计数法

#### 数与计数法

计数法是一种表示数的方法，我们平时一般使用10进制计数法，这种方法逢十进一。

与十进制一样，二进制也是一种计数方法。在十进制里逢十进位，在二进制里是逢二进位。

数（number）本身具有一个不依赖于计数法的真正的量值。比如这里有三个苹果，跟用什么计数法表示没有关系。不管是写成十进制的3还是二进制的11，苹果的个数都是一样的（还可以写成三进制的10，四进制的3，五进制的3 。。。）。



在我们都使用的十进制中，数中的每一位都代表不同的值（数位）。分别是个位，十位，百位，千位，等等。



例如，9在百位上，表示900。

在二进制中，每一位也表示一个值。但它不是每位乘以10，而是乘以2。因此它们从低到高是1位，2位，4位，8位，等等。



例如，数9在二进制下，就是1001。

为了计算它，我们需要加1次8，0次4，0次2，和1次1。



#### 二进制与十进制转换

##### 二进制转十进制

1111 1001 10000001 11111111

11.01 101.011

##### 十进制转二进制

5 15 99

4.5 8.75 9.875 1.2 2.05

### 二进制算术

二进制算术与普通的（十进制）算术的基本思想是一样的，唯一的区别就是采用二进制，而不是十进制，就是说逢二进一，借一当二。

加法

1101 + 10 = 1111 (13 + 2 = 15）

1110 + 1011 = 11001 (14 + 11 = 25)

减法

1101 – 101 = 1000 (13 – 5 = 8)

1100 – 111 = 101 (12 – 7 = 5)

乘法

1001 × 11 = 11011 （9 × 3 = 27）

1111 × 1011 = 10100101 （15 × 11 = 165）

除法

1111 / 11 = 101 （15 / 3 = 5）

### 八进制、十六进制

除了二进制，编程中也经常使用八进制和十六进制。  
  
八进制有0~7共8个数字，基数为8，逢八进一，借一当八；

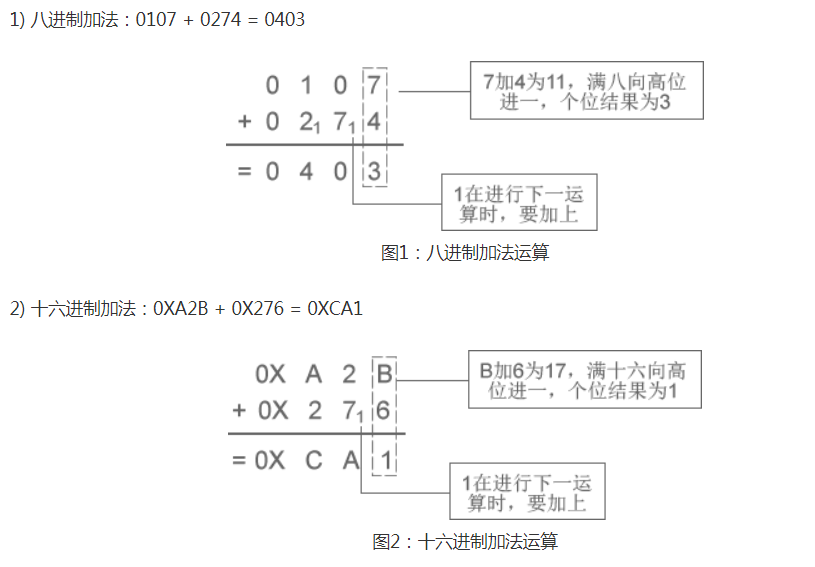
十六进制中，用A来表示10，B表示11，C表示12，D表示13，E表示14，F表示15，因此有0~F共16个数字，基数为16，逢16进1，借1当16。

例如：

* 八进制 3072 = 3×83 + 0×82 + 7×81 + 2×80 = 1536 + 0 + 56 + 2 = 1594
* 十六进制 E3F9 = 14×163 + 3×162 + 15×161 + 9×160 = 57344 + 768 + 240 + 9 = 58361

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 二进制、八进制、十进制、十六进制的对应关系 | | | | | | | |
| **十进制** | **二进制** | **八进制** | **十六进制** | **十进制** | **二进制** | **八进制** | **十六进制** |
| 0 | 0 | 0 | 0 | 10 | 1010 | 12 | A |
| 1 | 1 | 1 | 1 | 11 | 1011 | 13 | B |
| 2 | 10 | 2 | 2 | 12 | 1100 | 14 | C |
| 3 | 11 | 3 | 3 | 13 | 1101 | 15 | D |
| 4 | 100 | 4 | 4 | 14 | 1110 | 16 | E |
| 5 | 101 | 5 | 5 | 15 | 1111 | 17 | F |
| 6 | 110 | 6 | 6 | 16 | 10000 | 20 | 10 |
| 7 | 111 | 7 | 7 | 17 | 10001 | 21 | 11 |
| 8 | 1000 | 10 | 8 | 18 | 10010 | 22 | 12 |
| 9 | 1001 | 11 | 9 | 19 | 10011 | 23 | 13 |

#### 加法



### 小数的表示

### 负数的表示

在表示负数的时候，因为符号只有正负两个值，很自然想到可以用一位比特来表示符号。直接使用一位比特表示符号，0表示正，1表示负，这就是原码表示法。但是在实践中又发展了反码和补码。补码是在现代计算机中广泛采用的方法。

#### 原码、反码以及补码

①  原码：

　　原码是指将最高位作为符号位(0表示正，1表示负)，其它数字位代表数值本身的绝对值的数字表示方式。

　　例如：数字6 在计算机中原码表示为：0 000 0110

　　其中，第一个数字0是符号位，0表示正数，0 000110是数字6的二进制数据表示。

　　数字－6 在计算机中原码表示为：1 000 0110

　　以上是在8位计算机中的原码表示，如果在32位或16位计算机中，表示方法也是一样的，只是多了几个数字0而已。

　　例如，在32位 计算机中数字6 的原码表示为：00000000 0000 0000 0000 0000 0000 0110

　　在16位 计算机中数字6 的原码表示为：00000000 0000 0110

②   反码：

　　反码表示规则为：如果是正数，则表示方法和原码一样；如果是负数，符号位不变，其余各位取反，则得到这个数字的反码表示形式。

　　例如，数字6 在8位 计算机中的反码就是它的原码：00000110

　　数字－6 在 8位计算机中的反码为：11111001

③   补码

　　补码是计算机表示数据的一般方式，其规则为：如果是正数，则表示方法和原码一样；如果是负数，则将数字的反码加上1(相当于将原码数值位取反然后在最低位加1)。

　　例如：数字6 在8位 计算机中的补码就是它的原码：00000110

    数字－6 在8 位 计算机中的补码为：1111 1010

**注意：0的反码、补码都为零**

#### 原码、反码、补码的产生、应用以及优缺点有哪些？

<https://www.zhihu.com/question/20159860>

数字在自然界中抽象出来的时候，一棵树，两只猪，是没有正数和负数的概念的

计算机保存最原始的数字，也是没有正和负的数字，叫没符号数字

如果我们在内存分配4位（bit）去存放无符号数字，是下面这样子的



后来在生活中为了表示“欠别人钱”这个概念，就从无符号数中，划分出了“正数”和“负数”

正如上帝一挥手，从混沌中划分了“白天”与“黑夜”

为了表示正与负，**人们发明了"原码"，把生活应该有的正负概念，原原本本的表示出来**

把左边第一位腾出位置，存放符号，正用0来表示，负用1来表示

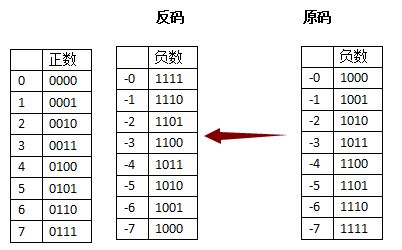
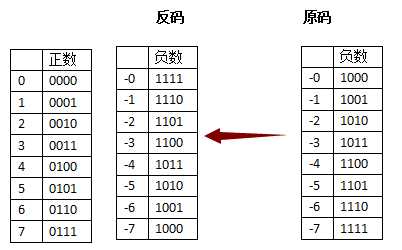
  
但使用“原码”储存的方式，方便了看的人类，却苦了计算机  
https://pic3.zhimg.com/50/7cec066778fbb42aa57598d64336f8b8_hd.jpghttps://pic3.zhimg.com/80/7cec066778fbb42aa57598d64336f8b8_hd.jpg  
我们希望 （+1）和（-1）相加是0，但计算机只能算出0001+1001=1010 (-2)

这不是我们想要的结果 (╯' - ')╯︵ ┻━┻

另外一个问题，这里有一个（+0）和（-0）

**为了解决“正负相加等于0”的问题，在“原码”的基础上，人们发明了“反码”**

“反码”表示方式是用来处理负数的，符号位置不变，其余位置相反

  
当“原码”变成“反码”时，完美的解决了“正负相加等于0”的问题

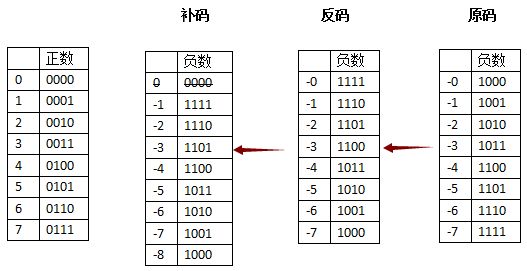
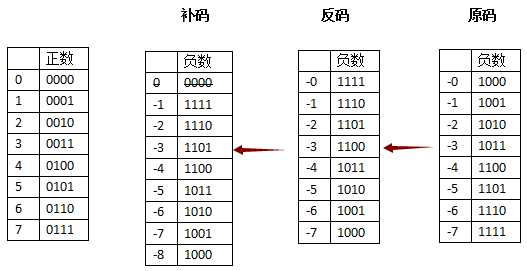
过去的（+1）和（-1）相加，变成了0001+1101=1111，刚好反码表示方式中，1111象征-0

人们总是进益求精，历史遗留下来的问题—— 有两个零存在，+0 和 -0

**我们希望只有一个0，所以发明了"补码"**，同样是针对"负数"做处理的

"补码"的意思是，从原来"反码"的基础上，补充一个新的代码，（+1）

我们的目标是，没有蛀牙（-0）

有得必有失，在补一位1的时候，要丢掉最高位

我们要处理"反码"中的"-0",当1111再补上一个1之后，变成了10000，丢掉最高位就是0000，刚好和左边正数的0，完美融合掉了

这样就解决了+0和-0同时存在的问题

另外"正负数相加等于0"的问题，同样得到满足

举例，3和（-3）相加，0011 + 1101 =10000，丢掉最高位，就是0000（0）

同样有失必有得，我们失去了(-0) , 收获了（-8）

以上就是"补码"的存在方式

**结论：保存正负数，不断改进方案后，选择了最好的"补码"方案**

## 科学计数法

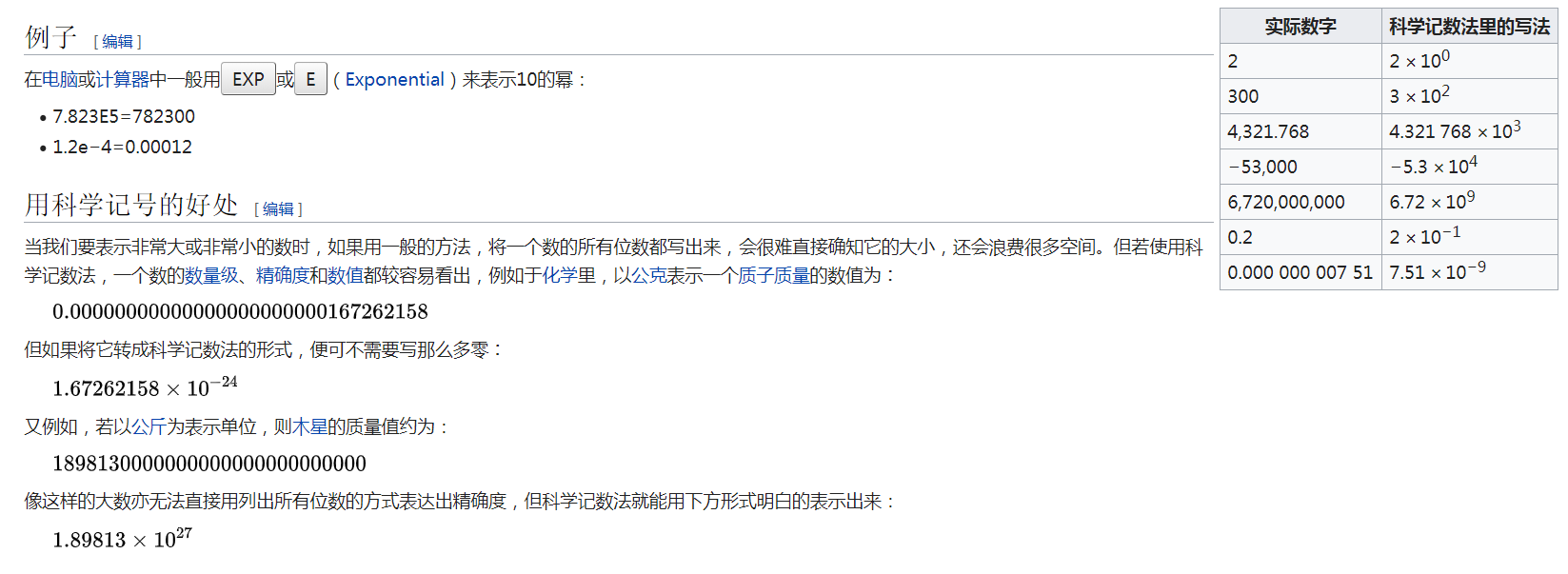
**科学记数法**（英语：Scientific notation，英国则称为 Standard form），又称为**科学记数法**或**科学记法**，是一种数字的表示法。科学记数法最早由[阿基米德](https://zh.wikipedia.org/wiki/%E9%98%BF%E5%9F%BA%E7%B1%B3%E5%BE%B7)提出。

科学记数法是一种记数的方法。把一个数表示成a与10的n次幂相乘的形式（1≤a<10，n为整数），这种记数法叫做**科学记数法。[2]**

例如：19971400000000=1.99714×1013。

计算器或电脑表达10的幂是一般是用E或e，也就是1.99714E13=19971400000000。

{\displaystyle n\,}



## 基本存储单位

信息的最小单位：比特Bit，一个二进制位

8位比特：字节Byte，ASCII码，可以表示256个不同符号

KB： Kilo Byte 210=1024 约等于1000

MB： Mega Byte 220=1024 × 1024 约等于1000，000

GB： Giga Byte 230=1024 × 1024 × 1024 约等于1000，000，000

## 练习：

1. 二进制算术（加减乘除）

1001 + 1101

1001 – 111

1011 × 101

1111 / 11

1. 二/十进制转换，

二进制转十进制 1101 1111 10000000

十进制转二进制 31 97 987 255

1. 二/十六进制转换

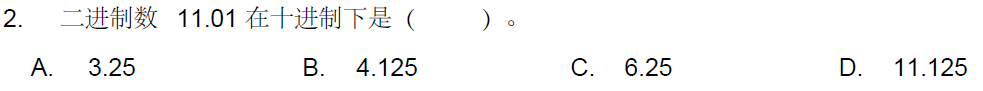
二进制到十六进制 11011111 10000001 11111111

十六进制到二进制 AF 10 1AF

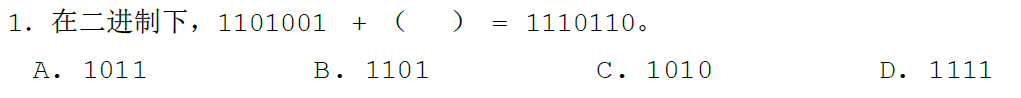
1. 求一个自然数N的个位数、十位数和百位数。

## 历年考题：

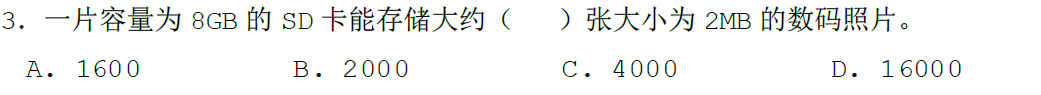
2013年



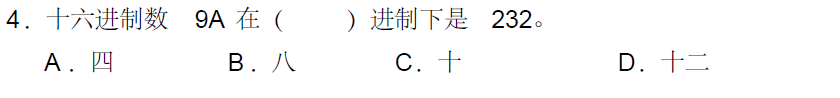
2011年







2012年



# 简单逻辑学、逻辑演算、布尔代数

## 逻辑论证

### 概念、判断和推理

概念：

三角形，圆

判断：

两点之间，线段最短

三角形内角之和等于180度，

等边三角形三个内角相等

三角形面积等于底乘以高除以二

圆上的任意一个点到圆心的距离都等于圆的半径。

推理：

三角形两边之和大于第三条边

直角三角形面积等于两条直边的乘积除以二

等边三角形三个内角都等于60度

A，B是圆上的两个点，所以OA=OB

### 逻辑三段论

所有人都是必死的。[苏格拉底](https://zh.wikipedia.org/wiki/%E8%8B%8F%E6%A0%BC%E6%8B%89%E5%BA%95)是人。苏格拉底是必死的。

所有的三角形内角之和都是180度。直角三角形是三角形。直角三角形内角之和等于180度。直角三角形两个非直角之和等于90度。

### 反证法

先假设命题是错误的，通过推理得出矛盾，这就证明命题是正确的。

南方某风水先生到北方看风水，恰逢天降大雪。乃作一歪诗：“天公下雪不下雨，雪到地上变成雨；早知雪要变成雨，何不当初就下雨。”他的歪诗又恰被一牧童听到，亦作一打油诗讽刺风水先生：“先生吃饭不吃屎，饭到肚里变成屎；早知饭要变成屎，何不当初就吃屎。”

王戎小时候，爱和小朋友在路上玩耍。一天，他们发现路边的一棵树上结满了李子，小朋友一哄而上，去摘李子，独有王戎没动。等到小朋友们摘了李子一尝，原来是苦的！他们都问王戎：“你怎么知道李子是苦的呢？”王戎说：“假如李子不苦的话，早被路人摘光了，而这树上却结满了李子，所以李子一定是苦的。”

“我肯定不会错”

“是人都是会错的”

## 逻辑演算

一般的，在数学中把用语言、符号或式子表达的，可以判断**真**假的陈述句叫做**命题**（判断题）。

**命题真**值只能取两个值：**真**或假。**真**对应判断正确，假对应判断错误。

比如：

“直角三角形面积等于两条直边的乘积除以二”，

“三角形两边之和大于第三条边”，

“所有的自然数都是整数”。

逻辑运算又称[布尔运算](https://baike.baidu.com/item/%E5%B8%83%E5%B0%94%E8%BF%90%E7%AE%97/10865631)。布尔用[数学方法](https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E6%96%B9%E6%B3%95/1747958)研究逻辑问题，成功地建立了[逻辑演算](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E6%BC%94%E7%AE%97/1861075)。他用[等式](https://baike.baidu.com/item/%E7%AD%89%E5%BC%8F/3517693)表示判断，把推理看作等式的变换。这种变换的有效性不依赖人们对符号的解释，只依赖于符号的组合规律 。这一逻辑理论人们常称它为[布尔代数](https://baike.baidu.com/item/%E5%B8%83%E5%B0%94%E4%BB%A3%E6%95%B0/525084)。20世纪30年代，[逻辑代数](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E4%BB%A3%E6%95%B0/1933102)在电路系统上获得应用，随后，由于[电子技术](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E6%8A%80%E6%9C%AF/2470)与计算机的发展，出现各种复杂的大系统，它们的变换规律也遵守布尔所揭示的[规律](https://baike.baidu.com/item/%E8%A7%84%E5%BE%8B/3311038)。逻辑运算 (logical operators) 通常用来测试真假值。最常见到的逻辑运算就是[循环](https://baike.baidu.com/item/%E5%BE%AA%E7%8E%AF/607158)的处理，用来判断是否该离开循环或继续执行循环内的指令。

### 表示方法

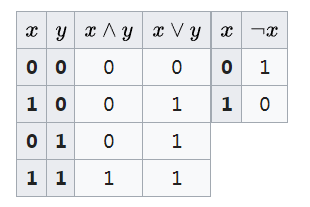
"∧" 表示"与"，"∨" 表示"或"， "┐"表示"非"，"=" 表示"等价"，1和0表示"真"和"假"

### 基本运算

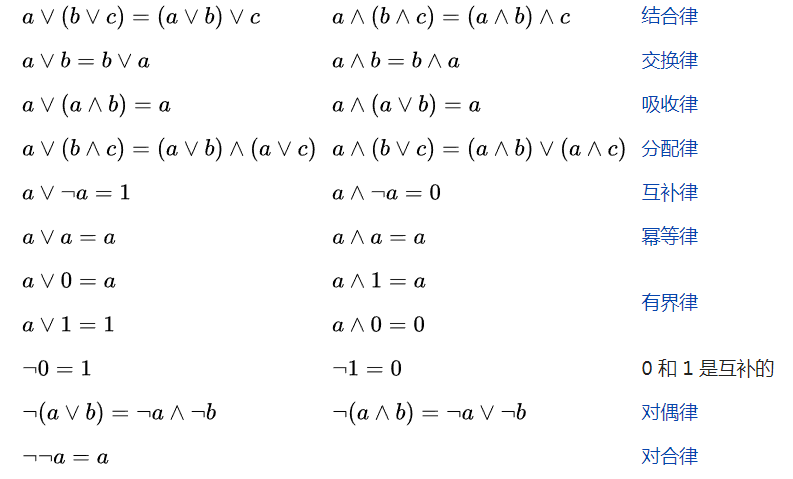
逻辑代数的基本运算如下。

* 与（[合取](https://zh.wikipedia.org/wiki/%E5%90%88%E5%8F%96)），记作 *x*∧*y*（有时记作 *x* AND *y* 或 K*xy*），在 *x* = *y* = 1 情况下，满足 *x*∧*y* = 1；其他情况下 *x*∧*y* = 0。
* 或（[析取](https://zh.wikipedia.org/wiki/%E6%9E%90%E5%8F%96)）, 记作 *x*∨*y*（有时记作 *x* OR *y* 或 A*xy*），在 *x* = *y* = 0 情况下，满足 *x*∨*y* = 0；其他情况下 *x*∨*y* = 1。
* 非 （[否定](https://zh.wikipedia.org/wiki/%E5%90%A6%E5%AE%9A)）, 记作 ¬*x*（有时记作 NOT *x*, N*x* 或 !*x*），在 *x* = 1 情况下，满足 ¬*x* = 0；而在 ¬*x* = 1 情况下，*x* = 0。

此外可以用制作[真值表](https://zh.wikipedia.org/wiki/%E7%9C%9F%E5%80%BC%E8%A1%A8)来表示 *x*∧*y*, *x*∨*y* 和 ¬*x* 的值：



两个主要的二元运算的符号定义为 {\displaystyle \land } (逻辑与)和 {\displaystyle \lor } (逻辑或)，把单一的一元运算的符号定义为 {\displaystyle \lnot }(逻辑非)。我们还使用值 0 (逻辑假)和 1 (逻辑真)。逻辑代数有下列性质:



### 三、练习

**例题1．**若A=True，B=False，C=True，D=False，以下逻辑运算表达式真的有（）。

A．（A∧B）∨（C∧D∨¬A）             B．（（¬A∧B）∨C）∧¬B

C．（B∨C∨D）∨D∧A                D．A∧（D∨¬C）∧B

**题解：**一个个算结果，比如A选项（A**∧**B）**∨**（C**∧**D**∨¬**A） ，根据运算级的比较，我们可以定下运算的顺序，然后按运算顺序计算结果。注意，这类题是有个小技巧的。比如A选项可以先看中间的**∨**，为什么呢？**因为∨的左右有一边是真就行，可以不去看另外一边。**

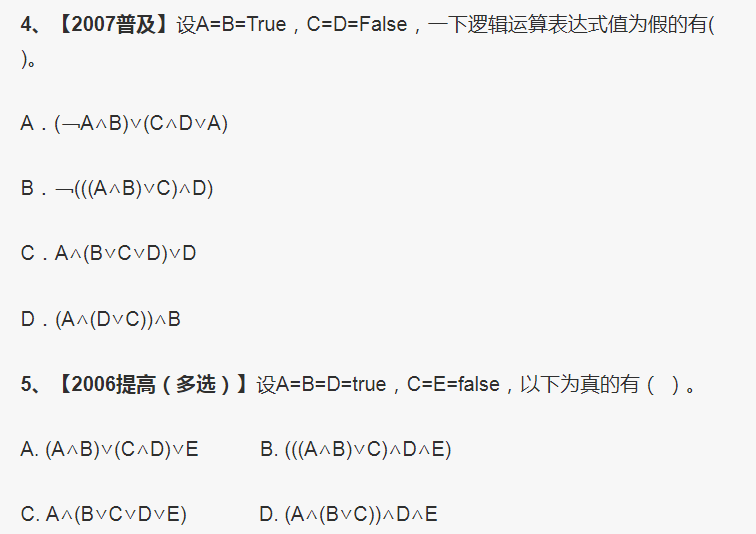
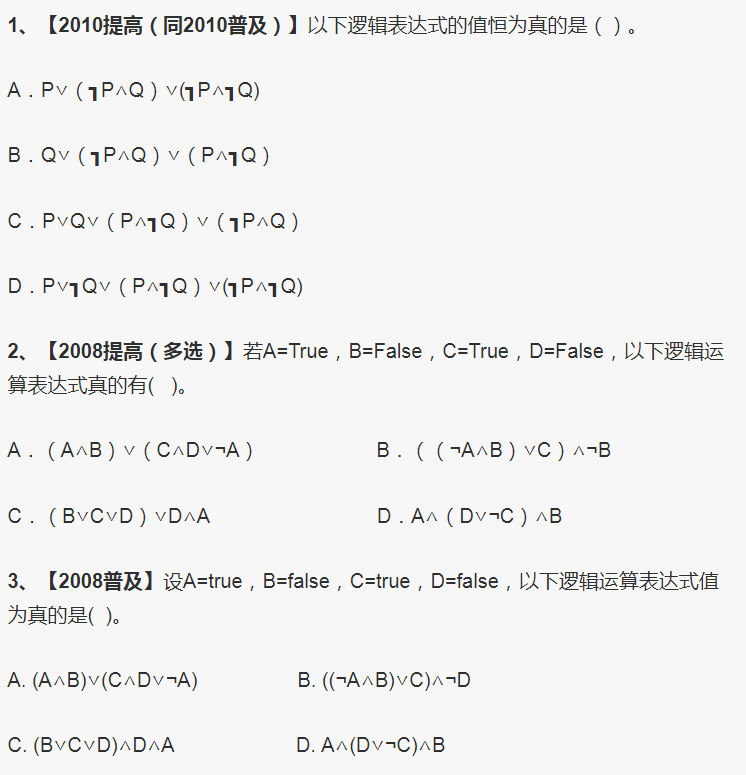
**A选项的结果是：**（A∧B）∨（C∧D∨¬A），（A∧B）=假，（C∧D∨¬A）中C∧D =假，¬A=假，所以（C∧D∨¬A）=假。于是A选项可以简写为：假∨（假∨假）=假。

**B选项的结果是：**（（¬A∧B）∨C）∧¬B，如果¬B是假那么就可以不去看前面的（（¬A∧B）∨C），可惜的是¬B是真，那么就要看（（¬A∧B）∨C），发现C是真，所以不看（¬A∧B），于是B选项可以简写为：（？∨真）∧真=真。

**C选项的结果是：**（B∨C∨D）∨D∧A，D∧A=假，所以不得不看前面部分（B∨C∨D），只要BCD有一个是真，那么（B∨C∨D）=真，而容易发现C=true。所以C选项可以简写为：真∨假=真。

**D选项的结果是：**A∧（D∨¬C）∧B，我们很容易发现D选项的特殊结构为 ？∧？∧？，三个？有一个是假，那么D为假，A和B不用计算便可看出，所以先发现B=假，所以D=假。

**历年题目**



答案：

1～5  A、BC、B、D、AC

### 逻辑门电路实现

与门

或门

非门

异或门

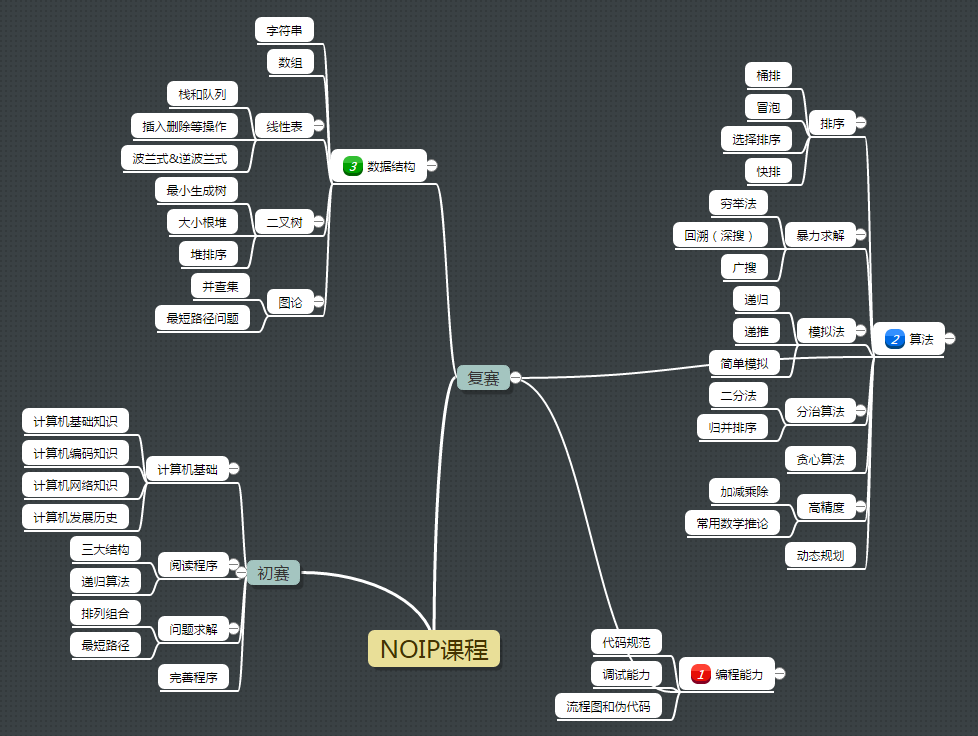
电子管

晶体管

### 布尔代数与二进制算术的关系

二进制加法

使用逻辑门电路实现二进制算术运算



参考：

### 计算机发展

（电子管，晶体管，集成电路，超大规模集成电路VLSI）

### 历年考题

