



进制转换与位运算

disangan233

2023/8/2



www.luogu.com.cn

进制

进制也就是进位计数制，是人为定义的带进位的计数方法。

对于任何一种进制—— X 进制，就表示每一位上的数运算时都是逢 X 进一位。十进制是逢十进一，十六进制是逢十六进一，二进制就是逢二进一。以此类推， X 进制就是逢 X 进位， X 称为基数。

现在最常用的是十进制，通常使用 10 个阿拉伯数字 0~9 进行记数。

对于任何一个数，我们可以用不同的进位制来表示。

比如：十进制数 $(57)_{10}$ ，用二进制表示为 $(111001)_2$ ，用五进制表示为 $(212)_5$ ，用八进制表示为 $(71)_8$ ，用十六进制表示为 $(39)_{16}$ ，它们所代表的数值都是一样的。

进制

X 进制中用 $0 \sim X-1$ 来计数，对于 $X > 10$ 的情况，我们用字母 A,B,C,…… 来代表 10,11,12,……

例如十六进制由数字 $0 \sim 9$ 加上字母 A~F 组成，大写字母用完后用小写字母，然后是符号。

本质上就是用字符去表示 10 及以上的整数。

需要重点提的是二进制、八进制和十六进制。

进制

二进制: Binary (bin)

二进制由两个数码 0,1 组成, 二进制数运算规律是逢二进一。

计算机使用的编码就是二进制, 二进制天然兼容逻辑运算。

在 C++14 中, 可以用 0b 为前缀书写二进制, 例如 7 写成 0b111。

进制

八进制: Octal (oct)

八进制由数码 0 ~ 7, 并且每个数码刚好对应三位二进制数, 所以八进制能很好地反映二进制, 此性质也会在进制转换时用到。

在 C++ 中, 可以用 0 为前缀书写八进制, 例如 12 写成 014。

进制

十六进制: Hexadecimal (hex)

十六进制由数字 0 ~ 9 加上字母 A~F 组成, 并且每个数码刚好对应四位二进制数, 所以十六进制也能很好地反映二进制。

在 C++ 中, 可以用 0x 为前缀书写十六进制, 例如 15 写成 0xf, 此时字母的大小写不影响结果。

十进制: Decimal (dec)

进制转换

对于 n 位 X 进制整数，考虑我们刚刚所讲的满 X 进 1 的定义，我们可以从其 X 进制中求出其十进制值。

令其从左往右第 i 位数码为 a_i ，有

$$(a_1 a_2 \cdots a_n)_X = a_1 X^{n-1} + a_2 X^{n-2} + \cdots + a_n X^0$$

所以从低位枚举到高位，计算累加答案即可。

进制转换

对于 n 位 X 进制小数，有与整数类似的计算方法。

小数点后的数码依然满足 $0 \sim X-1$ 的要求，计算方式跟整数相同。

令其从左往右第 i 位整数数码为 a_i ，小数数码为 b_i ，有

$$(a_1 \cdots a_n . b_1 \cdots b_m)_X = a_1 X^{n-1} + \cdots + a_n X^0 + b_1 X^{-1} + \cdots + b_m X^{-m}$$

对于负数次幂，有以下运算法则

$$X^{-a} = \frac{1}{X^a}$$

依然从低位枚举到高位，计算累加答案即可。

进制转换

我们已经讲解完了 X 进制转十进制的做法，接下来是一点练习：

二进制数 10110 转十进制

十六进制数 1AC 转十进制

二进制数 110.101 转十进制

进制转换

我们已经讲解完了 X 进制转十进制的做法，接下来是一点练习：

二进制数 10110 转十进制

$$(10110)_2 = 2^4 + 2^2 + 2^1 = 22$$

十六进制数 1AC 转十进制

$$(1AC)_{16} = 16^2 + 10 \times 16^1 + 12 = 428$$

二进制数 110.101 转十进制

$$(110.101)_2 = 2^2 + 2^1 + 2^{-1} + 2^{-3} = 6.625$$

进制转换

考虑十进制整数转 X 进制，采用进制的定义即可，使用短除法。
具体地，每次将当前数除以 X 的余数记下，然后将该数除以 X 后下取整，重复操作直到当前数变为 0。

将记下的所有余数反着排列，就可以得到转换后的 X 进制数。

例如将十进制整数 13 转为二进制

2		13		
2		6	1
2		3	0
2		1	1
		0	1

即： $(13)_{10} = (1101)_2$

进制转换

考虑十进制小数转 X 进制，整数部分与小数部分可以分别计算，于是整数部分按照刚刚的方法计算即可。

考虑数码从左到右位数变小、X 的幂次也变小，可以发现小数部分将除法改为乘法即可。

具体地，提出小数部分 p，每次将 p 乘上 X，然后记下并减去整数部分，重复计算直到 p 为 0。记下的数码顺序排列就是小数部分。

例如对于十进制小数 13.375，整数部分化为 1101，小数 0.375 提出计算。

$$0.375 \times 2 = 0.75, \quad 0.75 \times 2 = 1.5, \quad 0.5 \times 2 = 1$$

记下的数字分别为：0,1,1，可得 0.375 化为二进制为 0.011。

于是 13.375 化为二进制即为 1101.011。

进制转换

我们已经讲解完了十进制转 X 进制的做法，接下来是一点练习：

十进制数 127 转二进制

十进制数 53.48 转五进制

进制转换

我们已经讲解完了十进制转 X 进制的做法，接下来是一点练习：

十进制数 127 转二进制

$$127 \div 2 = 63 \dots\dots 1, 63 \div 2 = 31 \dots\dots 1, 31 \div 2 = 15 \dots\dots 1$$

$$15 \div 2 = 7 \dots\dots 1, 7 \div 2 = 3 \dots\dots 1, 3 \div 2 = 1 \dots\dots 1$$

$$(127)_{10} = (1111111)_2$$

十进制数 53.48 转五进制

$$53 \div 5 = 10 \dots\dots 3, 10 \div 5 = 2 \dots\dots 0$$

$$0.48 \times 5 = 2.4, 0.4 \times 5 = 2$$

$$(53.48)_{10} = (203.22)_5$$

进制转换 综合练习

$(10110110101)_2$ 转成八进制是多少?

十进制有限小数转二进制小数一定是有限小数吗?

C++ 表达式 `printf("%o", 0x10+010+0b10)` 会输出什么?

进制转换 综合练习

$(10110110101)_2$ 转成八进制是多少？

三位一转， $(10\ 110\ 110\ 101)_2 = (2665)_8$

十进制有限小数转二进制小数一定是有限小数吗？

不一定，比如 0.3 会有循环节，本质上是用 2 的次幂去拟合小数。

反过来是一定的，因为 2 没有 10 没有的质因数。

C++ 表达式 `printf("%o", 0x10+010+0b10)` 会输出什么？

$0x10+010+0b10=16+8+2=26=032$ ，所以输出 32。

输出技巧：`%d,%i,%o,%x` 和 `ios_base`

位运算

程序中的所有数在计算机内存中都是以二进制的形式储存的，位运算就是直接对整数在内存中的二进制位进行操作。

C++ 语言提供了六种位运算符来进行位运算操作：

1. 按位与 &
2. 按位或 |
3. 按位异或 ^
4. 按位非 ~
5. 左移 <<
6. 右移 >>

我们将分别对这六种位运算符进行讲解。

位运算

按位与：C++ 中写作 `&`，数学公式中写作 `and`。

将参与运算的两操作数各对应的二进制位进行与操作，只有对应的两个二进制位均为 1 时，结果的对应二进制位才为 1，否则为 0。

例如： $10 \text{ and } 13 = (1010)_2 \text{ and } (1101)_2 = (1000)_2 = 8$ 。

按位与的应用：

- 通常用来将某变量中的某些位变成 0 且同时保留其他位不变。
例如 `n&=0xffffffff00` 可以将 int 型后 8 位清 0。
- 用来获取某变量中的某一位。
例如判断 `n` 的从低到高第 3 位是否是 1 可以用 `n&4` 来判断。

代码中 `if(x&1)` 等价于 `if(x%2==1)`。

注意：`&&` 是对 bool 进行的逻辑与，`5&&7=true&&true=1`

位运算

按位或：C++ 中写作 `|`，数学公式中写作 `or`。

将参与运算的两操作数各对应的二进制位进行或操作，只有对应的两个二进制位均为 0 时，结果的对应二进制位才为 0，否则为 1。

例如： $10 \text{ or } 13 = (1010)_2 \text{ or } (1101)_2 = (1111)_2 = 15$ 。

按位或的应用：

- 通常用来将某变量中的某些位变成 1 且同时保留其他位不变。
例如 `n |= 0x0000ffff` 可以将 `int` 型后 16 位变成 1。

注意：`||` 是对 `bool` 进行的逻辑或，`5 || 0 = true`，`|| false = 1`。

位运算

按位异或：C++ 中写作 \wedge ，数学公式中写作 xor。

将参与运算的两操作数各对应的二进制位进行异或操作，只有对应的两个二进制位不同时，结果的对应二进制位才为 1，否则为 0。

例如： $10 \text{ xor } 13 = (1010)_2 \text{ xor } (1101)_2 = (0111)_2 = 7$ 。

异或具有可差分性，即 $a \wedge b = c$ 可以推出 $c \wedge a = b, c \wedge b = a$ 。

代码中 $\text{if}(x \wedge 1)$ 等价于 $\text{if}(x \neq 1)$ 。

位运算

按位非（取反）：C++ 中写作 \sim ，是单目运算符。

其功能是将操作数中的二进制位 0 变成 1，1 变成 0。

例如对有符号 short 型整数 86 按位非，有

$$\begin{aligned}\sim 86 &= \sim(0000\ 0000\ 0101\ 0110)_2 = (1111\ 1111\ 1010\ 1001)_2 \\ &= -87\end{aligned}$$

通过编码的知识，我们可以知道 $\sim x = -(x+1)$ 。

$\text{if}(\sim x)$ 和 $\text{if}(x \neq -1)$ 等价。

位运算

左移运算符：C++ 中写作 `<<`。

`a<<b` 得到的值是将 `a` 的二进制位全部左移 `b` 位后得到的值，左移时高位丢弃，低位补 0，`a` 的值不因运算而改变。

实际上左移一位就是 $\times 2$ ，于是左移 n 位就是 $\times 2^n$ ，左移操作比乘法操作快得多。

例如：`3<<2=12`。

位运算

右移运算符：C++ 中写作 `>>`。

`a>>b` 得到的值是将 `a` 的二进制位全部右移 `b` 位后得到的值，右移时移出最右边的位丢弃，高位补 0，`a` 的值不因运算而改变。

实际上右移一位就是 $\div 2$ 后向下取整，右移 n 位就是 $\div 2^n$ 后向下取整。

例如：`7>>2=3>>1=1`。

有符号数的右移是算术右移，右移后符号位会赋为原符号位。

所以对于负数的右移，实际上是 `10xxx` 右移一位变为 `110xxx`。

运算符优先级

不同运算符共同作用时，需要区分运算符的优先级。

常用的运算符优先级如下：

1. `() [] -> ++ --`
2. `! ~ ++ -- - + * &`
3. `* / %`
4. `+ -`
5. `<< >>`
6. `& ^ |`
7. `&& ||`

不同优先级从小到大，同一优先级从左到右。

运算符优先级

本质上就是算术运算优先，其次再是位运算，最后是逻辑运算。

注意位运算非的优先级很高。

一些可以记忆的比如异或在与和或中间。

一些练习：

计算 C++ 表达式 $3 \& 1 | 4^2$ 的值

计算 C++ 表达式 $7 \ll 2 | 1$ 的值

运算符优先级

本质上就是算术运算优先，其次再是位运算，最后是逻辑运算。

注意位运算非的优先级很高。

一些可以记忆的比如异或在与和或中间。

一些练习：

计算 C++ 表达式 $3\&1|4^2$ 的值

$$3\&1|4^2=(3\&1)|(4^2)=3|6=7$$

计算 C++ 表达式 $7\<\<2|1$ 的值

$$7\<\<2|1=28|1=2$$

注意 $7\<\<2|1$ 和 $7\<\<2+1$ 的结果并不相等

位运算 综合练习

$x \& -x$ 和 $x - (x \& x - 1)$ 的值相等吗？它们的值代表什么？

C++ 语言表达式 $3 \& 0xA \sim 2$ 的值为？

C++ 语言表达式 $x -= p, x += x \gg 31 \& p$ 有什么用？

位运算 综合练习

$x \& -x$ 和 $x - (x \& x - 1)$ 的值相等吗？它们的值代表什么？
相等。代表 x 二进制最低位 1 及其后的 0 组成的数。
 $x \& -x$ 被称作 `lowbit(x)`，后面会学到这个的运用。

C++ 语言表达式 $3 \& 0xA - \sim 2$ 的值为？

$3 \& 0xA - \sim 2 = 3 \& (0xA - (-3)) = 3 \& 13 = 3$

C++ 语言表达式 $x -= p, x += x \gg 31 \& p$ 有什么用？

在 $[0, 2^p)$ 范围内实现 $x \% p$

定时练习

内容：15 道选择题，考察范围为所讲内容

形式：团队考试，spj 评测，OI 赛制

时间：30min

考完试休息 10min 后讲评，讲评完下课

定时练习 讲评

1. 对于一个二进制数 101011 转换成十进制应该是多少？（B）

对于二进制数 101011，转换为十进制得

$$2^5 + 2^3 + 2^1 + 2^0 = 32 + 8 + 2 + 1 = 43$$

定时练习 讲评

2. 对于一个十进制下的数 140 转十六进制应该是多少？（D）

对于十进制数 140，转换为十六进制有

$$140 \div 16 = 8 \cdots \cdots 12$$

所以对应的十六进制数为 8C

定时练习 讲评

3. $(1010)_2$ 转成八进制是多少? (A)

二进制转八进制可以 3 位转换一次

010 转为八进制为 2, 1 转为八进制为 1

故二进制 $(1010)_2$ 转八进制为 $(12)_8$

定时练习 讲评

4. 下列哪个数可能存在？（C）

- A. $(114514)_5$ ，五进制里不能出现 5
- B. $(1919)_9$ ，九进制里不能出现 9
- C. $(FFF)_{16}$ ，存在
- D. $(000)_1$ ，不存在标准一进制

定时练习 讲评

5. 对于一个二进制数 11.11 转换成十进制应该是多少? (C)

整数部分为 $2^1 + 2^0 = 3$

小数部分为 $\frac{1}{2^1} + \frac{1}{2^2} = 0.75$

所以答案为 3.75

定时练习 讲评

6. 对于一个十进制数 11.375 转换成二进制应该是多少？(B)

整数部分有 $11 \div 2 = 5 \cdots 1$, $5 \div 2 = 2 \cdots 1$, $2 \div 2 = 1 \cdots 0$,
 $1 \div 2 = 0 \cdots 1$; 所以整数部分为 1011

小数部分有 $0.375 \times 2 = 0.75$, $0.75 \times 2 = 1.5$, $0.5 \times 2 = 1$,
所以小数部分为 0.011

所以答案为 1011.011

定时练习 讲评

7. 对于 C++ 语言表达式 $2\&4|3$ 的值是? (C)

与运算的优先级高于或运算

所以先计算 $2\&4=0$ ，再有 $0|3=3$

定时练习 讲评

8. 对于 C++ 语言表达式 $2^1 \& 4 \mid 3$ 的值是? (C)

与运算的优先级高于异或运算，再高于或运算
所以先计算 $1 \& 4 = 0$ ，再有 $2^0 = 2$ 与 $2 \mid 3 = 3$

定时练习 讲评

9. 对于 C++ 语言表达式 $x \ll 3 | 1$ 的值等价于? (D)

先执行左移运算再执行或运算

左移三位相当于乘上 $2^3 = 8$, 此时二进制的末位为 0, 因此再或 1 相当于加上 1, 即答案为 $x * 8 + 1$

定时练习 讲评

10. 以下 C++ 语言表达式中值最大的是? (A)

A. $4 \ll 3 \mid 4 \gg 1 = (4 \ll 3) \mid (4 \gg 1) = 32 \mid 2 = 34$

B. $2 * 2 + 1 - 3 = 2$

C. $2^{31-1} = 2^{(31-1)} = 28$

D. $2 * 2 * 2 = 8$

定时练习 讲评

11. $(11101101001)_2$ 转换成十六进制数是多少? (B)

二进制转十六进制可以 4 位转换一次

1001 转为十六进制为 9, 0110 转为十六进制为 6, 0111 转为十六进制为 7

故二进制 $(11101101001)_2$ 转十六进制为 $(769)_{16}$

定时练习 讲评

12. 以下 C++ 语言表达式中值最大的是？（C）

- A. 0xA 为十六进制，转为十进制后为 10
- B. 011 为八进制，转为十进制后为 9
- C. 十进制的 11
- D. 012 为八进制，转为十进制后为 10

定时练习 讲评

13. C++ 语言表达式 $213 \gg 3$ 的值为? (B)

213 的二进制为 11010101, 右移 3 位后为 $(11010)_2 = (26)_{10}$

- A. 030 为八进制, 转为十进制后为 24
- B. 0x1A 为十六进制, 转为十进制后为 26
- C. 0x19 为十六进制, 转为十进制后为 25
- D. 十进制的 25

定时练习 讲评

14. C++ 语言表达式 $6\&\sim 7+3$ 的值为? (A)

先运算优先级最高的取反, 然后为加法, 最后才为位运算

$$6\&\sim 7+3 = 6\&((\sim 7)+3) = 6\&3 = 2$$

定时练习 讲评

15. C++ 语言表达式 $6\&012^0xA-2*3|8>>1$ 的值为? (C)

算术运算最先, 先乘后减

再然后才是位运算

$$\begin{aligned}6\&012^0xA-2*3|8>>1 &= 6\&012^0xA-(2*3)|(8>>1) \\ &= 6\&10^4|4 \\ &= ((6\&10)^4)|4 \\ &= 6\end{aligned}$$