Review of binary number representation

Number systems and bases

A given number in a β-system is represented as

$$(a_n \dots a_2 a_1 a_0 . b_1 b_2 b_3 \dots)_\beta = \sum_{k=0}^{n} a_k \beta^k + \sum_{k=1}^{\infty} b_k \beta^{-k}$$

Examples:

- Decimal base:

$$(426.97)_{10} = 4\times10^2 + 2\times10^1 + 6\times10^0 + 9\times10^{-1} + 7\times10^{-2}$$

- Binary base:

$$(1011.001)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$$

Integer numbers in a computer

From decimal to binary: $(39)_{10}$

Method 1:

#/2	Quotient	Remainder
39/2	19	1
19/2	9	1
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1

Method 2:

	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	64	32	16	8	4	2	1
39	39	7	7	7	3	1	0
#	0	1	0	0	1	1	1

$(39)_{10} = (100111)_2$

Integer numbers in a computer

From binary to decimal: $(10111)_2$

$$(10111)_2 = \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 1 & 1 & 1 \\ \hline \end{array}$$

$$2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

$$= 1\times16 + 0\times8 + 1\times4 + 1\times2 + 1\times1 = 23$$

$$(10111)_2 = (23)_{10}$$

Practice questions

Convert $(110101)_2$ to decimal number

A) 43
B) 53
C) 42
D) 52

Convert $(175)_{10}$ to binary number

A) $(01111101)_2$
B) $(10111110)_2$
C) $(11110101)_2$
D) $(10101111)_2$

Real numbers in a computer

Real numbers add an extra level of complexity. Not only do they have a leading integer, they also have a fractional part.

From decimal to binary: $(39.6875)_{10}$

Method 1:

Same as before for the integer part

$$(39)_{10} = (100111)_2$$

For the decimal part, use the following table:

#×2	Integer part	Fractional part
1.375	1	0.375
0.75	0	0.75
1.5	1	0.5
1.0	1	0

$$(39.6875)_{10} = (100111.1011)_2$$

Method 2:

	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	32	16	8	4	2	1	0.5	0.25	0.125	0.0625
#	1	0	0	1	1	1	1	0	1	1
39.6875	7.6875	7.6875	7.6875	3.6875	1.6875	0.6875	0.1875	0.1875	0.0625	0

Real numbers in a computer

From binary to decimal: $(101101.101)_2$

$$(101101.101)_2 = \underbrace{\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{1}}_{2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0}\ \underbrace{\boxed{1}\boxed{0}\boxed{1}}_{2^{-1}\ 2^{-2}\ 2^{-3}}$$

$$= 1\times32 + 0\times16 + 1\times8 + 1\times4 + 0\times2 + 1\times1 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$$

$$(101101.101)_2 = (45.625)_{10}$$

Practice questions

Convert $(11101.11)_2$ to decimal number

A) 19.75
B) 25.75
C) 23.75
D) 29.75

Convert $(67.125)_{10}$ to binary number

A) $(1000011.001)_2$
B) $(1100001.001)_2$
C) $(1100001.01)_2$
D) $(1000011.01)_2$

Convert $(23.3)_{10}$ to binary number

	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
	16	8	4	2	1	0.5	0.25	0.125	0.0625	0.03125
#	1	0	1	1	1	0	1	0	0	1
23.3	7.3	7.3	3.3	1.3	0.3	0.3	0.05	0.05	0.05	0.01875

	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}
	0.015625	0.0078125	0.00390625	0.00195313	0.000976563
#	1	0	0	1	1
0.01875	0.003125	0.003125	0.003125	0.00117188	0.000195313

$$(10111.010011001)_2 = (23.2998046875)_{10}$$

$$(a_n \dots a_2 a_1 a_0 . b_1 b_2 b_3 \dots)_\beta = \sum_{k=0}^{n} a_k \beta^k + \sum_{k=1}^{\infty} b_k \beta^{-k}$$

Looks like 23.3 is represented by an infinite series in the binary base!

Tips for conversion using Python:

There are more ways to do this! You can use your own favorite method.

Decimal Fraction (less than 1) to Binary

```
def decfrac_to_binary(fdec,n=23):
    f = ''
    dig = 0
    while ((fdec > 0) & (dig<n)):
        fdec = fdec*2
        temp_int = int(fdec)
        fdec = fdec - temp_int
        f += str(temp_int)
        dig += 1
    return f

decfrac_to_binary(0.42891717890,n=6)
```

```
'011011'
```

Integer to Binary

```
1  def integer_to_binary(x, n=23):
2      f = ''
3      dig = 0
4      while ((x>0) & (dig<n)):
5          remainder = x%2
6          x = int(x/2)
7          f += str(remainder)
8          dig += 1
9      return f[::-1]
10
11 integer_to_binary(2341)
```

```
'100100100101'
```

Much simpler method to convert integer to binary

Binary to Decimal

binary is normalized

```
def binary_to_decimal(f,m):
    decimal = 1
    for i,digit in enumerate(f):
        decimal += int(digit)*2**(-(i+1))
    decimal *= 2**(m)
    return decimal
```

```
binary_to_decimal('00010101',8)
```

```
277.0
```

Decimal Integer to Binary

```
bin(2341)[2:]
```

```
'100100100101'
```

Conversion of other systems (for example, ternary), can be obtained with a simple modification of these code snippets.