067 Add Binary

2018年3月29日 21:09

Question:

Given two binary strings, return their sum (also a binary string).
For example,
a = "11"
b = "1"
Return "100".

来自 <https://leetcode.com/problems/add-binary/description/>

给定两个二进制字符串，返回他们的和（用二进制表示）。

案例:

a = "11"
b = "1"

返回 "100" 。

Solution for Python3:

```
class Solution1:
    def addBinary(self, a, b):
        """
        :type a: str
        :type b: str
        :rtype: str
        """
        if not a or not b:
            return a + b
        s = ''
        i, j, c = len(a) - 1, len(b) - 1, 0
        while i >=0 or j >= 0 or c == 1:
            c += int(a[i]) if i >= 0 else 0
            c += int(b[j]) if j >= 0 else 0
            s = str(c % 2) + s
            c //= 2
            i -= 1
            j -= 1
        return s

class Solution2:
    def addBinary(self, a, b):
        """
        :type a: str
        :type b: str
        :rtype: str
        """
        if len(a) == 0:
            return b
        if len(b) == 0:
            return a
        if a[-1] == '1' and b[-1] == '1':
            return self.addBinary(self.addBinary(a[0:-1], b[0:-1]), '1') +
'0'
        if a[-1] == '0' and b[-1] == '0':
```

```
            return self.addBinary(a[0:-1], b[0:-1]) + '0'
        else:
            return self.addBinary(a[0:-1], b[0:-1]) + '1'

class Solution3:
    def addBinary(self, a, b):
        """
        :type a: str
        :type b: str
        :rtype: str
        """
        return bin(eval('0b' + a) + eval('0b' + b))[2:]

class Solution4:
    def addBinary(self, a, b):
        """
        :type a: str
        :type b: str
        :rtype: str
        """
        return '{:b}'.format(int(a, 2) + int(b, 2))
```

Solution for C++:

```
class Solution {
public:
    string addBinary(string a, string b) {
        if (a.empty() || b.empty()) {
            return a + b;
        }
        string s = "";
        int i = a.length() - 1, j = b.length() - 1, c = 0;
        while (i >= 0 || j >= 0 || c == 1) {
            c += i >= 0 ? a[i--] - '0' : 0;
            c += j >= 0 ? b[j--] - '0' : 0;
            s = char(c % 2 + '0') + s;
            c /= 2;
        }
        return s;
    }
};
```

Appendix:

eval(expression, globals=None, locals=None):

将字符串str当成有效的表达式来求值并返回计算结果。

globals和locals参数是可选的，如果提供了globals参数，那么它必须是dictionary类型；如果提供了locals参数，那么它可以是任意的map对象。

当一行代码要使用变量 x 的值时，Python 会到所有可用的名字空间去查找变量，按照如下顺序:

1）局部名字空间 - 特指当前函数或类的方法。如果函数定义了一个局部变量 x, 或一个参数 x，Python 将使用它，然后停止搜索。

2）全局名字空间 - 特指当前的模块。如果模块定义了一个名为 x 的变量，函数或类，Python 将使用它然后停止搜索。

3）内置名字空间 - 对每个模块都是全局的。作为最后的尝试，Python 将假设 x 是内置函数或变量。

python的全局名字空间存储在一个叫globals()的dict对象中；局部名字空间存储在一个叫locals()的dict对象中。我们可以用print (locals())来查看该函数体内的所有变量名和变量值。

locals()对象的值不能修改，globals()对象的值可以修改。

bin(int):

将整数转换为二进制字符串。如bin(4)返回字符串'0b100'。再通过bin(4)[2:]去掉开头'0b'。

int(str, int):

将几进制字符串转整数。如int('100',2)

str.format():基本语法是通过 {} 和 : 来代替以前的 % 。

1) "{1} {0} {1}".format("hello", "world") # 设置指定位置

2) 数字格式化：

数字	格式	输出	描述
3.1415926	{:.2f}	3.14	保留小数点后两位
3.1415926	{:+.2f}	+3.14	带符号保留小数点后两位
-1	{:+.2f}	-1.00	带符号保留小数点后两位
2.71828	{:.0f}	3	不带小数
5	{:0>2d}	05	数字补零 (填充左边, 宽度为2)
5	{:x<4d}	5xxx	数字补x (填充右边, 宽度为4)
10	{:x<4d}	10xx	数字补x (填充右边, 宽度为4)
1000000	{:,}	1,000,000	以逗号分隔的数字格式
0.25	{:.2%}	25.00%	百分比格式
1000000000	{:.2e}	1.00e+09	指数记法
13	{:10d}	13	右对齐 (默认, 宽度为10)
13	{:<10d}	13	左对齐 (宽度为10)
13	{:^10d}	13	中间对齐 (宽度为10)

11	`'{:b}'.format(11)`
	`'{:d}'.format(11)`
	`'{:o}'.format(11)`
	`'{:x}'.format(11)`
	`'{:#x}'.format(11)`
	`'{:#X}'.format(11)`
		1011
		11
		13
		b
		0xb
		0XB
			进制

^, <, > 分别是居中、左对齐、右对齐，后面带宽度， : 号后面带填充的字符，只能是一个字符，不指定则默认是用空格填充。

+ 表示在正数前显示 +，负数前显示 -； （空格）表示在正数前加空格

b、d、o、x 分别是二进制、十进制、八进制、十六进制。

此外我们可以使用大括号 {} 来转义大括号，如下实例：

print ("{} 对应的位置是 {{0}}".format("runoob"))

结果：runoob 对应的位置是 {0}