想法: a+b之後看哪邊有2就進位

11,1

[1,2]

class Solution:

def addBinary(self, a: str, b: str) -> str:

if len(a) < len(b):

a ,b = b, a

a = [int(i) for i in a]

b = [int(i) for i in b]

for i in range(len(b)):

i = i + 1

a[-i] = a[-i] + b[-i]

while 2 in a or 3 in a:

if a[0] == 2:

a[0] = 0

a.insert(0, 1)

elif a[0] == 3:

a[0] = 1

a.insert(0, 1)

for i in range(len(a)-1):

i = i + 1

if a[-i] ==2:

a[-i] = 0

a[-i-1] = a[-i-1] +1

elif a[-i] ==3:

a[-i] = 1

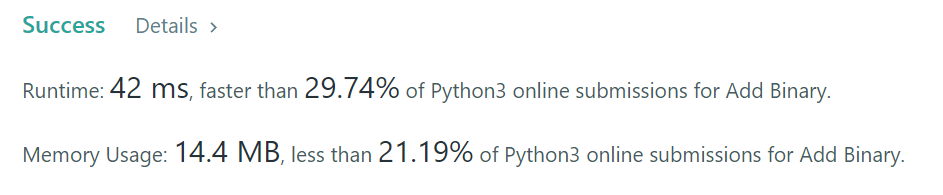
a[-i-1] = a[-i-1] +1

result = ""

for i in a:

result = result + str(i)

return(result)



Others:

1.

class Solution:

def addBinary(self, a, b):

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'

2.

class Solution:

def addBinary(self, a: str, b: str) -> str:

carry = 0

result = ''

a = list(a)

b = list(b)

while a or b or carry:

if a:

carry += int(a.pop())

if b:

carry += int(b.pop())

result += str(carry %2)

carry //= 2

return result[::-1]