1. class Solution:

def reverse(self, x: int) -> int:

a = []

d = 0

if math.copysign(1,x) == 1:

x = x

d = 1

elif math.copysign(1,x) == -1:

x = -x

d = -1

(c,b) = divmod(x,10)

a.append(b)

while c != 0:

(c,b) = divmod(c,10)

a.append(b)

R = 0

for i in range(len(a)):

R = R + a[i] \* int(math.pow(10,len(a)-1-i))

R = R \* d

if R < math.pow(-2,31) or R > (math.pow(2,31)-1):

return 0

else:

return R

2.

class Solution:

def reverse(self, x: int) -> int:

a = 0

d = 0

if math.copysign(1,x) == 1:

x = x

d = 1

elif math.copysign(1,x) == -1:

x = -x

d = -1

(c,b) = divmod(x,10)

e = len(str(x))-1

a = b\*int(math.pow(10,e))

while c != 0:

(c,b) = divmod(c,10)

a = a + b\*int(math.pow(10,e-1))

e = e - 1

a = a \* d

if a < math.pow(-2,31) or a > (math.pow(2,31)-1):

return 0

else:

return a

3.

class Solution:

def reverse(self, x: int) -> int:

a = 0

d = 0

if math.copysign(1,x) == 1:

x = x

d = 1

elif math.copysign(1,x) == -1:

x = -x

d = -1

(c,b) = divmod(x,10)

a = b

while c != 0:

(c,b) = divmod(c,10)

a = a\*10 + b

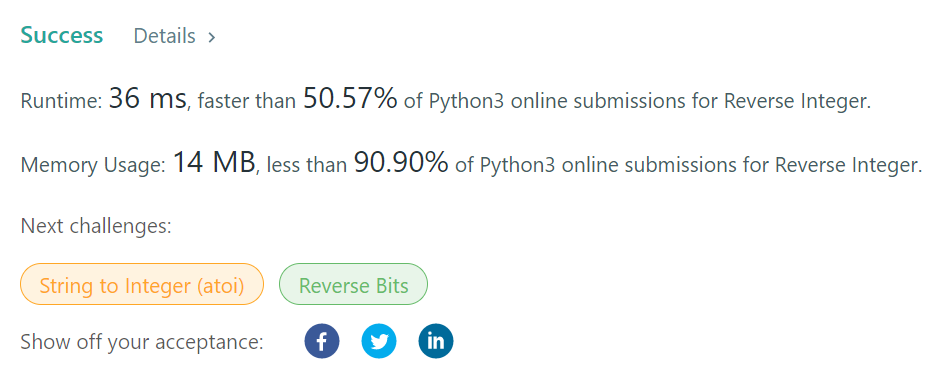
a = a \* d

if a < math.pow(-2,31) or a > (math.pow(2,31)-1):

return 0

else:

return a



Other one

class Solution:

def reverse(self, x):

range\_ = range(-2\*\*31, 2\*\*31-1)

y = str(abs(x))

if x == 0 or x not in range\_: x = 0 (maybe no need)

x = int('-' + y[::-1]) if '-' in str(x) else int(y[::-1])

return x if x in range\_ else 0