class Solution:

def isValid(self, s: str) -> bool:

# trans = {'(' : 1, ')' : 2, '{' : 3, '}' : 4, '[':5 , ']':6}

if len(s)%2 !=0:

return False

else:

previouslen\_1 = 0

previouslen\_2 = 0

slist = list(s)

while len(slist) !=0 and previouslen\_1 != len(slist):

previouslen\_1 = len(slist)

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

if slist[i] == '(' and slist[i+1] == ')':

del slist[i+1]

del slist[i]

break

elif slist[i] == '{' and slist[i+1] == '}':

del slist[i+1]

del slist[i]

break

elif slist[i] == '[' and slist[i+1] == ']':

del slist[i+1]

del slist[i]

break

while len(slist) !=0 and previouslen\_2 != len(slist):

previouslen\_2 = len(slist)

if slist[0] == '(':

for i in range(len(slist)):

if i%2 !=0 and slist[i] == ')':

del slist[i]

del slist[0]

break

elif slist[0] == '{':

for i in range(len(slist)):

if i%2 !=0 and slist[i] == '}':

del slist[i]

del slist[0]

break

elif slist[0] == '[':

for i in range(len(slist)):

if i%2 !=0 and slist[i] == ']':

del slist[i]

del slist[0]

break

else:

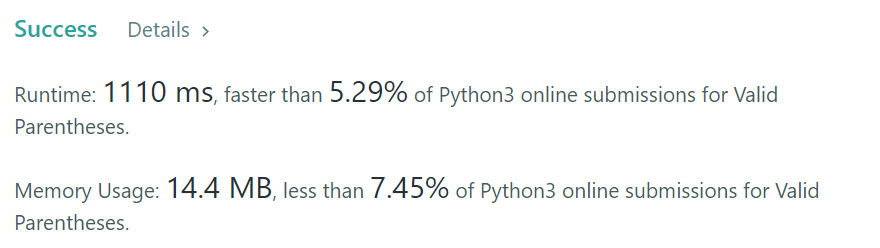
return False

if len(slist) ==0:

return True

else:

return False



Others:真的把stack用的熟

class Solution(object):

def isValid(self, s):

"""

:type s: str

:rtype: bool

"""

d = {'(':')', '{':'}','[':']'}

stack = []

for i in s:

if i in d: # 1

stack.append(i)

elif len(stack) == 0 or d[stack.pop()] != i: # 2

return False

return len(stack) == 0 # 3

# 1. if it's the left bracket then we append it to the stack

# 2. else if it's the right bracket and the stack is empty(meaning no matching left bracket), or the left bracket doesn't match

# 3. finally check if the stack still contains unmatched left bracket

