1.

5.1.1

from random import randint

def find\_coin\_2(L):

if len(L)<=1:

print("Error: coins are too few"); quit()

i = 0

while i < len(L):

if L[i] != L[i+1]:

if L[i]<L[i+1]:return i

else:return i+1

i+=2

n =int(input())

w\_normal = randint(2,5)

index\_faked = randint(0,n-1)

L=[]

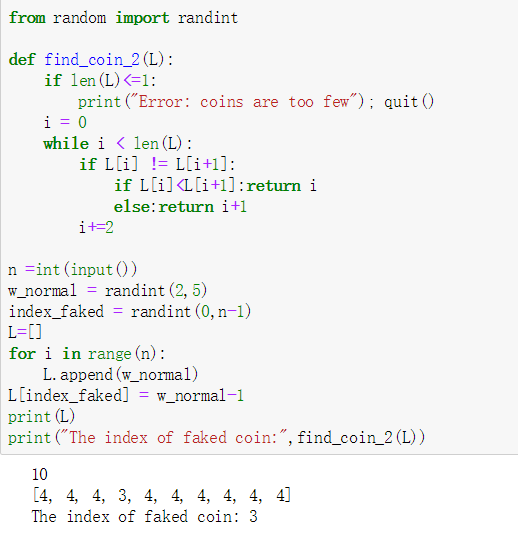
for i in range(n):

L.append(w\_normal)

L[index\_faked] = w\_normal-1

print(L)

print("The index of faked coin:",find\_coin\_2(L))



5.1.2

from random import randint

def find\_coin\_3(L):

if len(L)<=1:

print("Error: coins are too few"); quit()

else:

k = L

p = 0

while len(k)>1:

if len(k)%2:

if sum(k[1:len(k)//2+1])==sum(k[len(k)//2+1:]):

break

else:

if sum(k[1:len(k)//2+1])<sum(k[len(k)//2+1:]):

k = k[1:len(k)//2+1]; p+=1

else:k = k[len(k)//2+1:]; p+=len(k)+1

else:

if sum(k[:len(k)//2])<sum(k[len(k)//2:]):

k = k[:len(k)//2]

else: k = k[len(k)//2:];p+=len(k)

return p

n =int(input())

w\_normal = randint(2,5)

index\_faked = randint(0,n-1)

L=[]

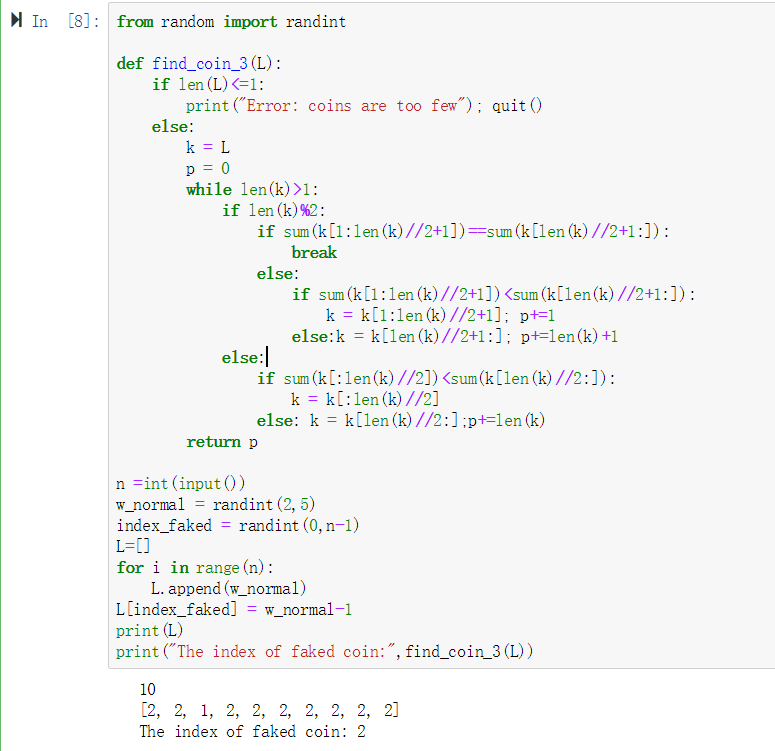
for i in range(n):

L.append(w\_normal)

L[index\_faked] = w\_normal-1

print(L)

print("The index of faked coin:",find\_coin\_3(L))



5.1.3

a可以指示L[0]在原始L里的index，return(-1)代表没有假币

5.1.4

能。

K分法时，k越大单层里需要比较的次数越多，但层数越少。

T(n)=)

当k越大时指数也越大，且只有k=2时T(k)=1，所以当n很大时k>=3是消耗很大的

故k>=3时没有意义

5.1.5

一一比较

from random import randint

def find\_coin(L):

if len(L)<=3:

print("Error: coins are too few"); quit()

i = 0

k=[]

while i < len(L)-1:

if L[i] != L[i+1]:

if L[i]<L[i+1]:k.append(i)

i+=1

return k[0],k[1]

n =int(input())

w\_normal = randint(2,5)

index\_faked1 = randint(0,n-1)

index\_faked2 = randint(0,n-1)

while index\_faked1==index\_faked2:index\_faked2 = randint(0,n-1)

L=[]

for i in range(n):

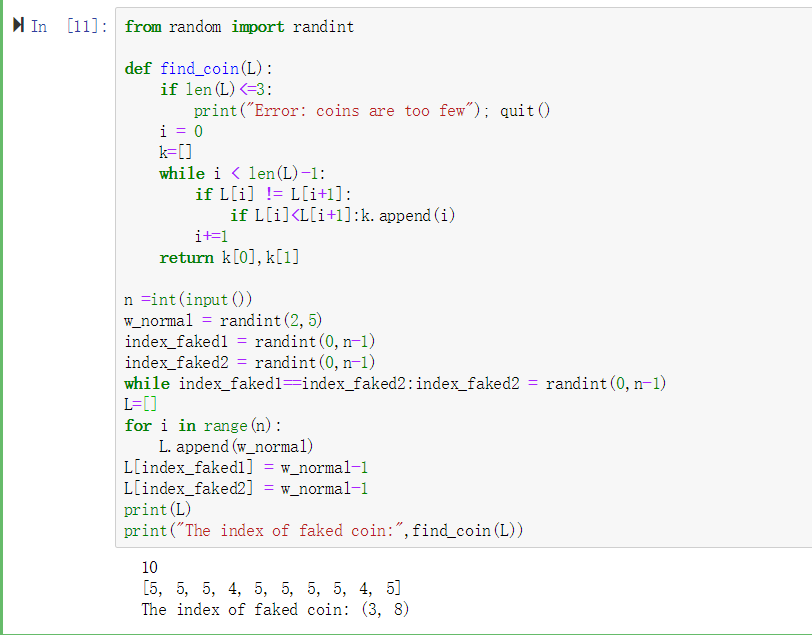
L.append(w\_normal)

L[index\_faked1] = w\_normal-1

L[index\_faked2] = w\_normal-1

print(L)

print("The index of faked coin:",find\_coin(L))



5.1.6

先比较前两项，如果一样，说明假币在后面，遍历列表与L[0]比较即可

如果不一样，说明假币就是这两个的一个，与L[2]比较，不一样的就是假币

2.

5.1.7

from random import randint

def find\_coin(L):

if len(L)<=2:

print("Error: coins are too few"); quit()

if L[0]!=L[1]:

if L[0]==L[2]:return 1

else:return 2

else:

for i in range(len(L)):

if L[i]!=L[0]:return i

n =int(input())

w\_normal = randint(2,5)

index\_faked = randint(0,n-1)

L=[]

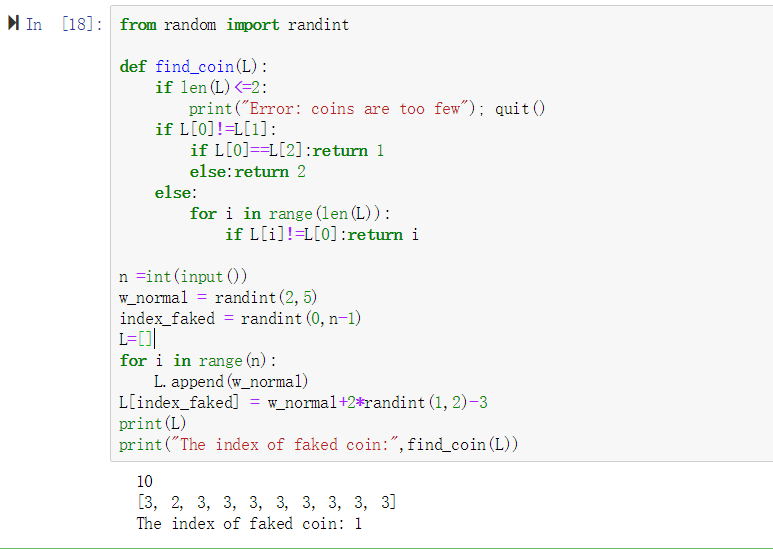
for i in range(n):

L.append(w\_normal)

L[index\_faked] = w\_normal+2\*randint(1,2)-3

print(L)

print("The index of faked coin:",find\_coin(L))



3.

def hanoi(n,a,b,c):

global L

if n<1:print('False')

elif n==1:

L.append([a,c])

elif n>1:

hanoi(n-1,a,b,c)

hanoi(n-1,c,a,b)

hanoi(1,a,b,c)

hanoi(n-1,b,a,c)

n = int(input())

L=[]

s = hanoi(n,0,1,2)

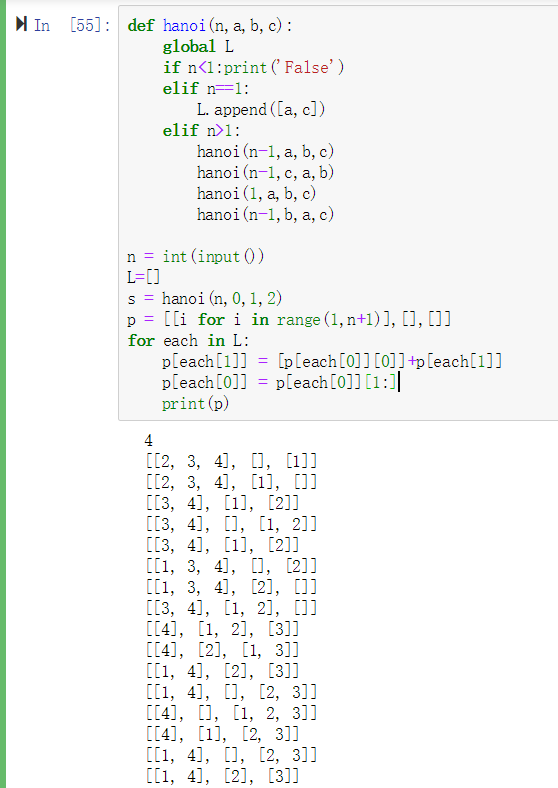
p = [[i for i in range(1,n+1)],[],[]]

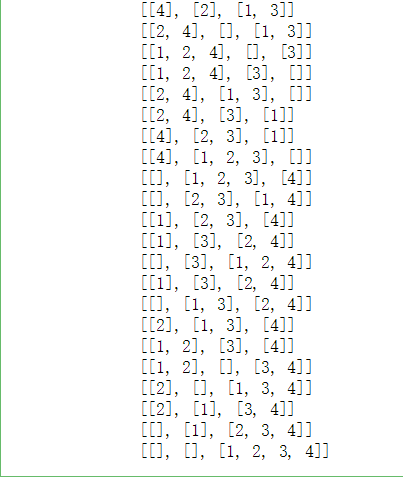
for each in L:

p[each[1]] = [p[each[0]][0]]+p[each[1]]

p[each[0]] = p[each[0]][1:]

print(p)





4.

5.3.1

总和乘积最大数可以，减法不可以，要满足交换律和结合律

5.3.2

#递归

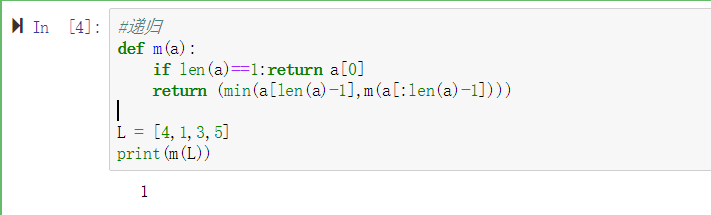
def m(a):

if len(a)==1:return a[0]

return (min(a[len(a)-1],m(a[:len(a)-1])))

L = [4,1,3,5]

print(m(L))



#分治法

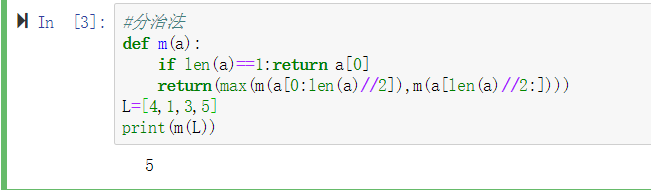
def m(a):

if len(a)==1:return a[0]

return(max(m(a[0:len(a)//2]),m(a[len(a)//2:])))

L=[4,1,3,5]

print(m(L))



5.

5.3.1

def msort(L):

k = len(L)

if k == 0:return(L)

if k == 1:return(L)

X1 = []; X2 = []

for i in L[1:]:

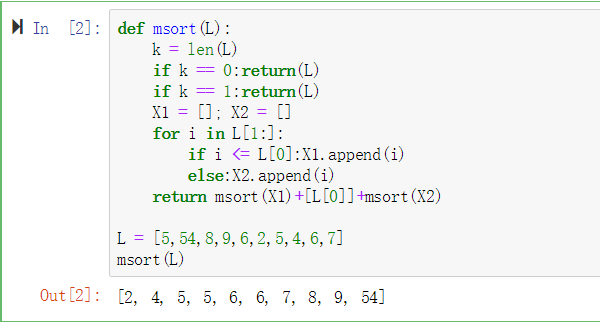
if i <= L[0]:X1.append(i)

else:X2.append(i)

return msort(X1)+[L[0]]+msort(X2)

L = [5,54,8,9,6,2,5,4,6,7]

msort(L)



5.3.2

def FA(a,b,c):

carry = (a and b) or (b and c) or (a and c)

sum = (a and b and c) or (a and not b and not c) or (not a and b and not c) or (not a and not b and c)

return carry,sum

def add\_divide(x,y,c=False):

while len(x)<len(y):x=[False]+x

while len(y)<len(x):y=[False]+y

if len(x)==1:

ctemp,stemp=FA(x[0],y[0],c)

return ctemp,[stemp]

if len(x)==0:return c,[]

c1,s1 = add\_divide(x[len(x)//2:],y[len(y)//2:],c)

x1,y1 = add\_divide(x[:len(x)//2],y[:len(y)//2],True)

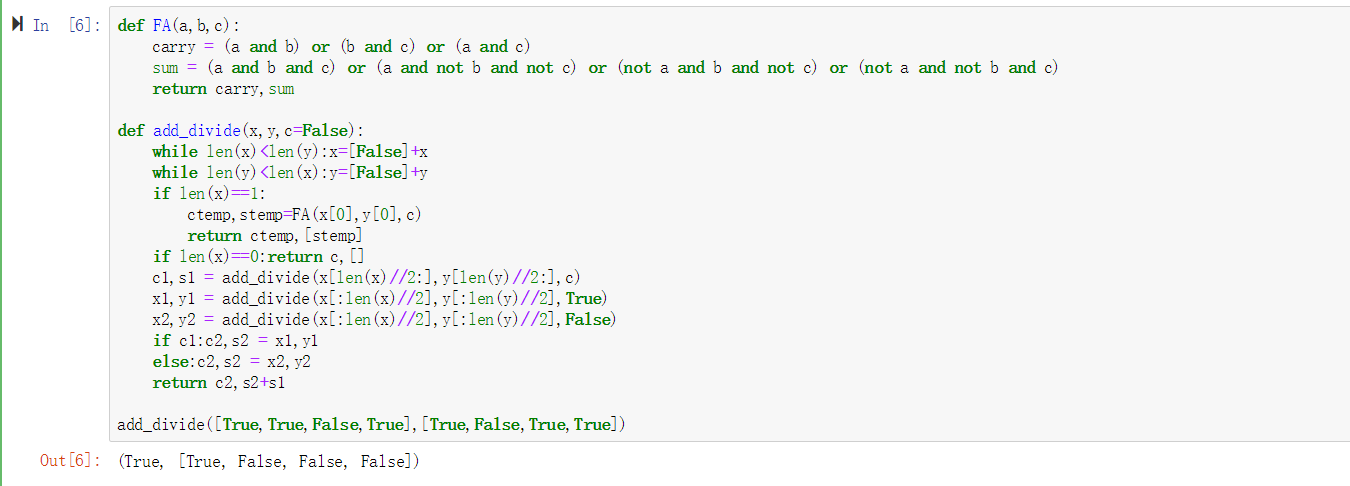
x2,y2 = add\_divide(x[:len(x)//2],y[:len(y)//2],False)

if c1:c2,s2 = x1,y1

else:c2,s2 = x2,y2

return c2,s2+s1

add\_divide([True,True,False,True],[True,False,True,True])



6.

7.

游戏名：star wars

成员：10185102232；10185102128；10185102134；10185102229

游戏介绍：玩家对战类，可控制子弹发射力度与角度