

文件分为三个模块，运行test.py即可获得三次生成假币列表寻找随机假币位置的图表

核心代码（coin\_compare）：

# coin\_dir = [1, 1, 1, 1, 5]

def sum\_coin(co\_list, fom, to):

sum\_co = 0

for i in range(fom, to+1, 1):

sum\_co += co\_list[i]

return sum\_co

def compare\_coin(co\_list, fom, to):

"""

:param co\_list: 硬币列表

:param fom: 硬币列表起始位置

:param to: 硬币列表末尾

:return: 假币位置

"""

# 硬币数

num = to - fom + 1

# 假币位置

res = 0

if num == 1:

if fom == 0:

front = co\_list[fom] == co\_list[fom + 1]

back = co\_list[fom] == co\_list[fom + 2]

if back and front:

res = 0

elif front and (not back):

res = fom + 2

elif (not front) and back:

res = fom + 1

else:

res = 0

elif fom == len(co\_list) - 1:

front = co\_list[fom] == co\_list[fom - 2]

back = co\_list[fom] == co\_list[fom - 1]

if back and front:

res = 0

elif front and (not back):

res = fom - 1

elif (not front) and back:

res = fom - 2

else:

res = fom

else:

front = co\_list[fom] == co\_list[fom - 1]

back = co\_list[fom] == co\_list[fom + 1]

if back or front:

res = 0

else:

res = fom

else:

# 判断硬币总数是否为偶数

if num % 2 == 0:

front = sum\_coin(co\_list, fom, int(fom + num / 2 - 1))

back = sum\_coin(co\_list, int(to - num / 2 + 1), to)

if front == back:

res = 0

else:

res1 = compare\_coin(co\_list, fom, int(fom + num / 2 - 1))

if res1 == 0:

res2 = compare\_coin(co\_list, int(to - num / 2 + 1), to)

if res2 != 0:

res = res2

else:

res = 0

else:

res = res1

else:

res3 = compare\_coin(co\_list, fom + 1, to)

if res3 == 0:

if co\_list[fom] != co\_list[fom + 1]:

res = fom

else:

res = res3

return res