



การบ้าน Programming



Araya Wiwatwanich • Aug 8

10 points

Due 11:59 PM

1. จงเขียนโปรแกรม random ตัวเลข 0 กับ 1 บรรจุใน list ไปจนกว่าจะเจอ 0 ติดกัน 3 ตัว เช่น
[0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0]

2. จงสร้างฟังก์ชันการคำนวณระยะระหว่างพิกัด (x1,y1) และ (x2,y2) โดยใช้สูตร Euclidean Distance
จากนั้นให้เขียนโปรแกรมตามขั้นตอนต่อไปนี้

- รับค่าพิกัดของจำนวนจริง (x, y) มา 3 พิกัดให้ชื่อว่าพิกัด A, B และ C
- คำนวณระยะระหว่างพิกัดทั้งสามแล้ว print ออกมาทีละคู่ ด้วยข้อความ "Distance between A and B is ..."
- เปรียบเทียบระยะของทั้งสามคู่ แล้วแสดงผลคู่ที่ระยะสั้นที่สุด ด้วยข้อความ "The shortest path is between ... and ..."

แต่ละข้อให้ capture หน้าจอ ให้เห็นตัว code, ผลการ run, account google ของคุณ, และ timestamp
ถ้าโปรแกรมยาวก็ capture มาหลายรูปได้

เตือนรบกวนอาจารย์เฉลยการบ้านทั้ง 2 ข้อด้วยนะครับ (โดยเฉพาะข้อ 1 ผมคิดว่าจะผิด เพราะรันแต่ละครั้งผลไม่เหมือนกันบ่อยเลยครับ ขอขอบคุณมากครับ)

1. จงเขียนโปรแกรม random ตัวเลข 0 กับ 1 บรรจุใน list ไปจนกว่าจะเจอ 0 ติดกัน 3 ตัว เช่น [0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0]

တၢ်မံၤတၢ်ဖျါလၢနး

```
import random
```

mylist = []

count = 0

```
while count < 3:
```

```
a = random.randint(0,1)
```

```
mylist.append(a)
```

if $a == 0$:

count = count + 1

else :

count = 0

```
print(mylist)
```

DSA_Linear Algebra day#1 (6-Aug-22).ipynb ☆

CommentShareSettingsUser Avatar

FileEditViewInsertRuntimeToolsHelpAll changes saved

Code + TextRAMDiskEditing

เขียนโปรแกรม random ตัวเลข 0 กับ 1 บรรจุใน list ไปจนกว่าจะเจอ 0 ติดกัน 3 ตัว เช่น
[0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0]

Print: ฟังก์ชัน random ตัวเลข 0 กับ 1 คือ

```
1 import random
2 list_random_numbers = []
3 new_list_with_3_zero = []
4 count_zero = 0
5 
6 for i in range(50) :
7     list_random_numbers.append(random.randint(0,1))
8 
9 print("First round : ",list_random_numbers)
10 
11 for num in range(len(list_random_numbers)) :
12     num_cur = list_random_numbers[num]
13     num_back_1 = list_random_numbers[num - 1]
14     num_back_2 = list_random_numbers[num - 2]
15     num_back_3 = list_random_numbers[num - 3]
16     if num_cur == 0 and num_back_1 == 0 and num_back_2 == 0 and num_back_3 == 0:
17         break
18     else :
19         new_list_with_3_zero.append(list_random_numbers[num])
20 
21 
22 print("Second round :",new_list_with_3_zero)
```

First round : [1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0]
Second round : [1, 1, 0, 0, 0]

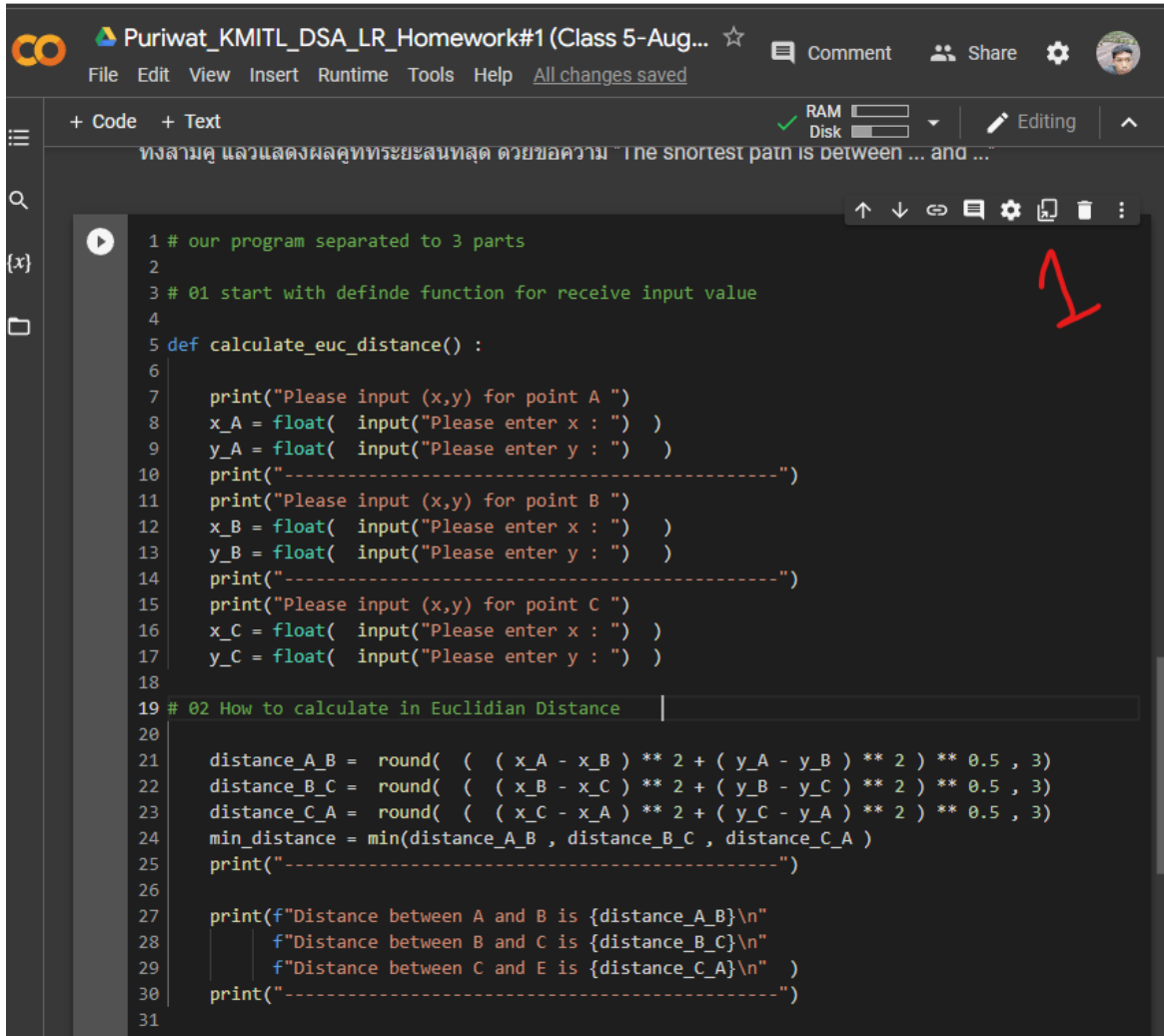
การเรียกใช้ Library Math

ฟังก์ชันทางคณิตศาสตร์ที่ใช้อยู่ เช่น max(), min(), abs(), pow() ได้ built-in มากับ Python แล้ว
ถ้ามีค่าอะไรก็ตามที่เราต้องการหาค่า เราสามารถใช้ library "math" เช่น ค่า pi ค่า e sin(x) cos(x) exp(x) log(x) ceil(x)
0s completed at 9:20 PM

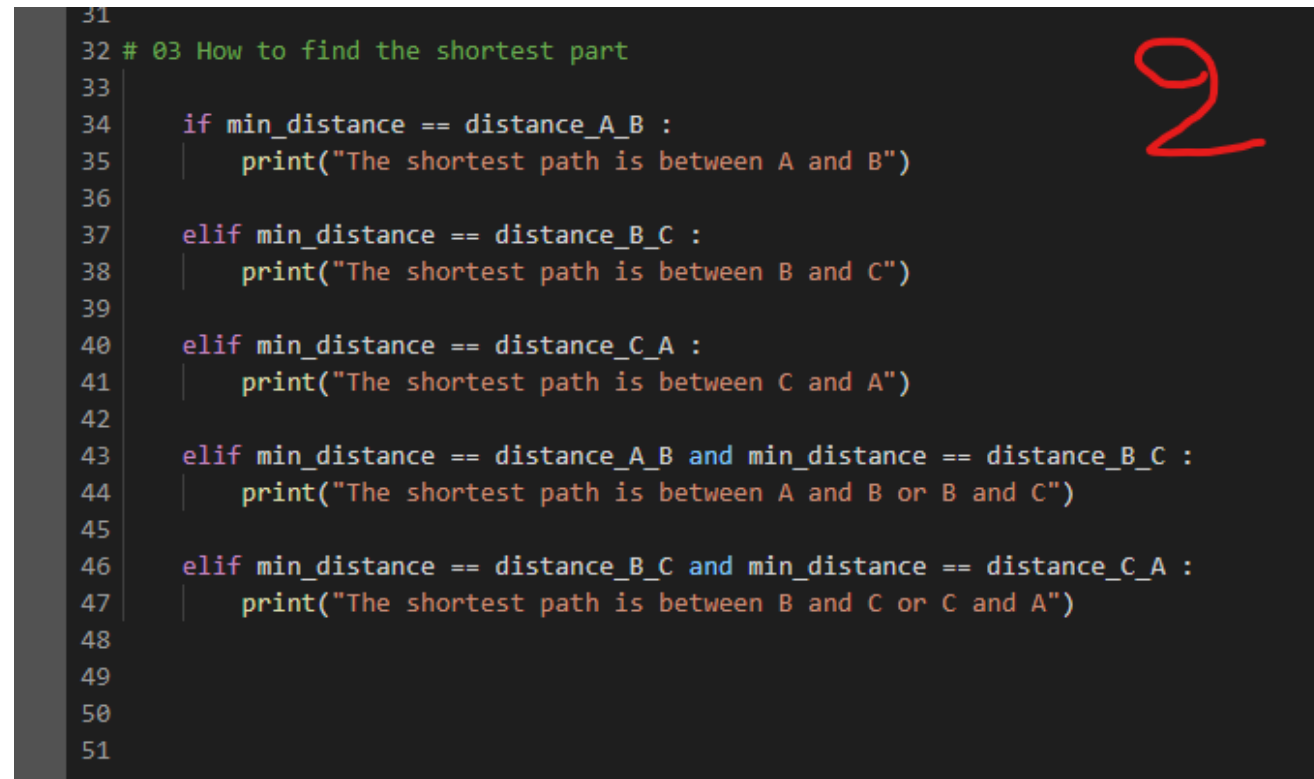
2. จงสร้างฟังก์ชันการคำนวณระยะระหว่างพิกัด (x1,y1) และ (x2,y2) โดยใช้สูตร Euclidean Distance จากนั้นให้เขียนโปรแกรมตามขั้นตอนต่อไปนี้

- รับค่าพิกัดของจำนวนจริง (x, y) มา 3 พิกัดให้ชื่อว่าพิกัด A, B และ C
- คำนวณระยะระหว่างพิกัดทั้งสามแล้ว print ออกมาทีละคู่ ด้วยข้อความ "Distance between A and B is ..."
- เปรียบเทียบระยะของทั้งสามคู่ แล้วแสดงผลคู่ที่ระยะสั้นที่สุด ด้วยข้อความ "The shortest path is between ... and ..."

Part 1 : Code



```
1 # our program separated to 3 parts
2
3 # 01 start with define function for receive input value
4
5 def calculate_euc_distance() :
6
7     print("Please input (x,y) for point A ")
8     x_A = float( input("Please enter x : ") )
9     y_A = float( input("Please enter y : ") )
10    print("-----")
11    print("Please input (x,y) for point B ")
12    x_B = float( input("Please enter x : ") )
13    y_B = float( input("Please enter y : ") )
14    print("-----")
15    print("Please input (x,y) for point C ")
16    x_C = float( input("Please enter x : ") )
17    y_C = float( input("Please enter y : ") )
18
19 # 02 How to calculate in Euclidian Distance
20
21 distance_A_B = round( ( ( x_A - x_B ) ** 2 + ( y_A - y_B ) ** 2 ) ** 0.5 , 3)
22 distance_B_C = round( ( ( x_B - x_C ) ** 2 + ( y_B - y_C ) ** 2 ) ** 0.5 , 3)
23 distance_C_A = round( ( ( x_C - x_A ) ** 2 + ( y_C - y_A ) ** 2 ) ** 0.5 , 3)
24 min_distance = min(distance_A_B , distance_B_C , distance_C_A )
25 print("-----")
26
27 print(f"Distance between A and B is {distance_A_B}\n"
28       f"Distance between B and C is {distance_B_C}\n"
29       f"Distance between C and E is {distance_C_A}\n" )
30 print("-----")
31
```



```
31
32 # 03 How to find the shortest part
33
34 if min_distance == distance_A_B :
35     print("The shortest path is between A and B")
36
37 elif min_distance == distance_B_C :
38     print("The shortest path is between B and C")
39
40 elif min_distance == distance_C_A :
41     print("The shortest path is between C and A")
42
43 elif min_distance == distance_A_B and min_distance == distance_B_C :
44     print("The shortest path is between A and B or B and C")
45
46 elif min_distance == distance_B_C and min_distance == distance_C_A :
47     print("The shortest path is between B and C or C and A")
48
49
50
51
```

Part 2 : Result

นี่ใช้ def
มาบวชมา
มาลบ 2 3 4 5
นี่ใช้ def

```
Puriwat_KMITL_DSA_LR_Homework#1 (Class 5-Aug... ☆
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text RAM Disk Editing
37 elif min_distance == distance_B_C :
38     print("The shortest path is between B and C")
39
40 elif min_distance == distance_C_A :
41     print("The shortest path is between C and A")
42
43 elif min_distance == distance_A_B and min_distance == distance_B_C :
44     print("The shortest path is between A and B or B and C")
45
46 elif min_distance == distance_B_C and min_distance == distance_C_A :
47     print("The shortest path is between B and C or C and A")
48
49
50
51

1 # call function , input value and wait to see the summary
2 calculate_euc_distance()

Please input (x,y) for point A
Please enter x : -10
Please enter y : 5
-----
Please input (x,y) for point B
Please enter x : 7
Please enter y : 2
-----
Please input (x,y) for point C
Please enter x : 8
Please enter y : 10
-----
Distance between A and B is 17.263
Distance between B and C is 8.062
Distance between C and E is 18.682
-----
The shortest path is between B and C

10s completed at 8:17 PM
```

นี่ใช้ numpy ใช่มั้ย
นี่ใช้ numpy ใช่มั้ย
def distance(x-1, y-1, x-2, y-2)
return np.sqrt((x-2-x-1)**2
+ (y-2-y-1)**2)
จากนั้นใช้ numpy ใช่มั้ย
distance_A_B = distance(x_A, y_A,
x_B, y_B)
และใช้ numpy ใช่มั้ย