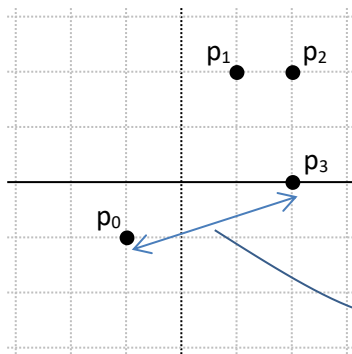


2559_2_NumPy_L2

จงเขียนฟังก์ชัน `all_pair_distances(p)` ที่รับ `array` สองมิติ `p` ที่มี `n` แถว 2 คอลัมน์ หนึ่งแถวแทนหนึ่งจุด คอลัมน์ 0 คือพิกัด `x` ของจุด คอลัมน์ 1 คือพิกัด `y` ของจุด เพื่อหา **ระยะห่างระหว่างทุกคู่จุดใน `p`**



```
p = np.array( [ [-1,-1], [1,2], [2,2], [2,0] ] )
d = all_pair_distances(p)
print(d)
```

```
[[ 0.          3.60555128  4.24264069  3.16227766]
 [ 3.60555128  0.          1.          2.23606798]
 [ 4.24264069  1.          0.          2.          ]
 [ 3.16227766  2.23606798  2.          0.          ]]
```

ให้เขียนคำสั่งในฟังก์ชันตาม `comment` ที่แนะนำในโครงโปรแกรมข้างล่างนี้ จะได้เสริมความเข้าใจการใช้ความสามารถของ `numpy`

```
import numpy as np

def all_pair_distances(p):
    x = _____ # พิกัด x ของทุก p ซึ่งคือข้อมูลในทุกแถว คอลัมน์ที่ 0 ของ p
    y = _____ # พิกัด y ของทุก p ซึ่งคือข้อมูลในทุกแถว คอลัมน์ที่ 1 ของ p
    X = np.array([x]) # สร้างเมทริกซ์ขนาด 1xn จากข้อมูลพิกัด x
    Y = np.array([y]) # สร้างเมทริกซ์ขนาด 1xn จากข้อมูลพิกัด y
    dX = _____ # ให้ dX คือ X ลบด้วย transpose ของ X
    dY = _____ # ให้ dY คือ Y ลบด้วย transpose ของ Y
    d = _____ # คำนวณระยะทางระหว่างทุกคู่จุดจากค่าของ dX กับ dY
    return d

exec(input().strip()) # do not remove this line
```

บรรทัดที่น่าสนใจเป็นพิเศษคือบรรทัดที่หาค่าของ `dx` และ `dy` นิสมศรลองพิมพ์ค่าภายในมาดูว่า `dx` เก็บอะไร ทำไมถึงได้แบบนั้น

ข้อมูลนำเข้า

คำสั่งในการทดสอบฟังก์ชันที่เขียน

ข้อมูลส่งออก

ผลที่ได้จากคำสั่งที่ป้อนเป็นข้อมูลนำเข้า

ตัวอย่าง

Input
<pre>p = np.array([[-1,-1], [1,2], [2,2], [2,0]]); print(all_pair_distances(p))</pre>
Output
<pre>[[0. 3.60555128 4.24264069 3.16227766] [3.60555128 0. 1. 2.23606798] [4.24264069 1. 0. 2.] [3.16227766 2.23606798 2. 0.]]</pre>