

กระบวนวิชา 229223

Lab	
HW	
Until	

การบ้านปฏิบัติการ 13

n -Dimensional Lists (20 คะแนน)

ข้อกำหนด

- i. การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ `import` ไปเรียกใช้งานจาก Script อื่น ๆ ได้
- ii. นักศึกษาสามารถสร้างฟังก์ชันย่อยต่าง ๆ เพิ่มเติมได้ตามความเหมาะสม
- iii. ฟังก์ชัน `main()` ควรอ่าน Input จากไฟล์ด้วยวิธี `Command Redirection` เพื่อความสะดวกในการทดสอบและหาข้อผิดพลาด

1) **4 คะแนน** (Lab13_1_6XXXXXXX.py) **[Attachment]** ให้เขียนฟังก์ชัน `matrix_mult(m1, m2)` เพื่อทำการหาผลคูณของเมทริกซ์ `m1` และ เมทริกซ์ `m2` (wikipedia: <https://goo.gl/S0DDZv>) โดยฟังก์ชันจะทำงานแบบ Non-destructive กล่าวคือจะคืนค่าผลคูณที่ได้โดยไม่เปลี่ยนแปลงเมทริกซ์ `m1` และ `m2` ที่อยู่ในในรูปแบบ List สองมิติ หากไม่สามารถหาผลคูณได้ให้คืนค่า `None`

"Dot Product"

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & \dots \\ \dots & \dots \end{bmatrix}$$

<u>Input</u>	<u>Output</u>
[[1, 2, 3], [4, 5, 6]] [[7, 8], [9, 10], [11,12]]	[[58, 64], [139, 154]]
[[1, 2, 3], [4, 5, 6]] [[7, 8, 5, 9, 3], [9, 10, -3, 7, 13], [11, 12, 6, 2, 9]]	[[58, 64, 17, 29, 56], [139, 154, 41, 83, 131]]

- การวิเคราะห์ปัญหา

• Input: จำนวนข้อมูล _____ ชนิดข้อมูล _____

• Output: (แสดงค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____

 (คืนค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____

2	7	6	→ 15
9	5	1	→ 15
4	3	8	→ 15

15 ← 15 ↓ 15 ↓ 15 ↘ 15

<pre>[[2, 7, 6], [9, 5, 1], [4, 3, 8]]</pre>	True
<pre>[[5, 5, 5], [5, 5, 5], [5, 5, 5]]</pre>	False
<pre>[[7, 12, 1, 14], [2, 13, 8, 11], [16, 3, 10, 5], [9, 6, 15, 4]]</pre>	True

- | | | | |
|-----------|-----------|-------------|------------|
| • Input: | | จำนวนข้อมูล | ชนิดข้อมูล |
| • Output: | (แสดงค่า) | จำนวนข้อมูล | ชนิดข้อมูล |
| | (คืนค่า) | จำนวนข้อมูล | ชนิดข้อมูล |

[[2, 3, 4], [1, 2, 3]]	[[2, 3, 4], [1, 2, 3], [0, 0, 0]]
---------------------------	---

<u>Input</u>	<u>Output</u>
[[1, 2], [1, 2, 3], [1, 2], [1, 2], [1]]	[[1, 2, 0, 0, 0], [1, 2, 3, 0, 0], [1, 2, 0, 0, 0], [1, 2, 0, 0, 0], [1, 0, 0, 0, 0]]

- การวิเคราะห์ปัญหา
 - Input: จำนวนข้อมูล_____ชนิดข้อมูล_____
 - Output: (แสดงค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____
 - (คืนค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____

4) **5 คะแนน** (HW13_2_6XXXXXXX.py) **[Attachment]** ให้เขียนฟังก์ชัน Destructive `bottom_up_m_sort(list_x, show_step=False)` เพื่อทำการเรียงลำดับสมาชิกใน Integer List `list_x` ตามลำดับจากน้อยไปมากโดยใช้ Merge Sort Algorithm โดยมี Optional Parameter `show_step` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration ทั้งนี้ไม่อนุญาตให้ใช้ฟังก์ชัน `sorted()` หรือ method `list.sort()` ในการแก้ปัญหา

<u>Function Call</u>	<u>Output</u>
<pre>list_x = [3, 7, 4, 9, 5, 2, 6] bottom_up_m_sort(list_x, True) print('-----') print(list_x)</pre>	<pre>[[3], [7], [4], [9], [5], [2], [6]] [[3, 7], [4, 9], [2, 5], [6]] [[3, 4, 7, 9], [2, 5, 6]] [[2, 3, 4, 5, 6, 7, 9]] --- [2, 3, 4, 5, 6, 7, 9]</pre>
<pre>list_x = [3, 7, 4, 9, 5, 2, 6, 1] bottom_up_m_sort(list_x) print('-----') print(list_x)</pre>	<pre>----- [1, 2, 3, 4, 5, 6, 7, 9]</pre>

- การวิเคราะห์ปัญหา
 - Input: จำนวนข้อมูล _____ ชนิดข้อมูล _____
 - Output: (แสดงค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____
 - (คืนค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____

5) 4 คะแนน (HW13_3_6XXXXXXX.py) **[Attachment]** ให้เขียนฟังก์ชัน `sum_d_product(m)` เพื่อคืนค่าผลบวกของผลคูณทแยงใน matrix m ที่มีขนาด $n \times n$ เมื่อ n สามารถเขียนในรูปของ 2^x (x เป็นจำนวนเต็มบวก)

a	b
c	d

โดยกรณี matrix m ขนาด 2×2 เช่น

1	2
3	4

ดังนั้น matrix จะมี $\text{sum_d_product}() = (1 \times 4) + (3 \times 2) = 10$

กรณีต้องการหา $\text{sum_d_product}()$ ของ matrix ขนาดใหญ่กว่า 2×2 ทำได้โดยการหา $\text{sum_d_product}()$

ของ matrix ย่อย ขนาด 2×2 ก่อน แล้วหา $\text{sum_d_product}()$ ของ matrix ผลลัพธ์อีกที

เช่นกรณี matrix ขนาด 4×4 จะมีขั้นตอนดังนี้

7	0	9	6
4	8	0	2
9	3	5	3
5	9	1	6

56	18
96	33

3576

หรือกรณี matrix 8×8

3	4	3	3	0	4	4	3
1	4	1	1	1	2	3	2
5	5	2	3	3	5	5	5
1	5	1	3	1	5	3	5
3	0	5	1	4	5	3	1
5	3	2	3	3	1	5	5
2	2	4	1	4	1	4	5
5	5	0	1	0	4	3	1

16	6	4	17
30	9	20	40
9	17	19	20
20	4	16	19

324	500
376	681

408644

Input

Output

[[3, 3, 3, 2], [2, 0, 3, 1], [2, 1, 2, 3], [1, 0, 2, -1]]	33
[[1, 1, 5, -1], [12, 2, -2, 0], [4, 8, 8, 12], [4, 12, 12, 15]]	3856
[[0, -1, -1, 3, 2, 3, -1, 3], [3, -1, -1, 2, 0, -1, 2, 1], [3, 0, 1, 2, 3, 1, 3, 1], [2, 2, 1, -1, -1, 2, 0, 3], [1, 3, 2, 1, 3, 2, 2, 1], [1, 2, 2, 1, 3, 3, 1, 3], [2, 2, 2, 2, 2, 2, 3, 3], [1, 3, 2, 3, 1, 1, 2, 2]]	-6290

การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล_____ชนิดข้อมูล_____
- Output: (แสดงค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____
- (คืนค่า) จำนวนข้อมูล_____ชนิดข้อมูล_____