

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....
กำหนดส่ง



กระบวนวิชา **229223**

Lab	
HW	
Until	

การบ้านปฏิบัติการ 12

Iterations - Part II (20 คะแนน)

ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ import ไปเรียกใช้งานจาก Script อื่น ๆ ได้
- นักศึกษาสามารถสร้างฟังก์ชันย่อยต่าง ๆ เพิ่มเติมได้ตามความเหมาะสม

$$2x^2 + 3 \cdot 2x^2 + x - 3 = 4x^4 + 2x^3 - 6x^2 + 6x^2 + 3x - 9$$

- 4 คะแนน (Lab12_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน `multiply_polynomials(p1, p2)` เพื่อคืนค่า list แทนผลคูณของ $p1$ และ $p2$ ซึ่งเป็น list แทน Polynomial (พหุนาม) ตัวอย่างเช่น $[2, 3, 0, 4]$ คือ $2x^3 + 3x^2 + 4$ ดังนั้น `multiply_polynomials([2, 0, 3], [4, 5])` จะแทนผลคูณ $(2x^2 + 3)(4x + 5) = 8x^3 + 10x^2 + 12x + 15$ และคืนค่า $[8, 10, 12, 15]$ ทั้งนี้ฟังก์ชันจะต้องไม่เปลี่ยนแปลง list $p1$ และ $p2$

Input

Output

$[2, 0, 3]$ $[4, 5]$	$[8, 10, 12, 15]$
-------------------------	-------------------

$$(2x^2 + 3)(2x^3 + x^2 + 2x + 5)$$

$$4x^5 + 2x^4 + 4x^3 + 10x^2 + 6x^2 + 3x^2 + 6x + 15$$

การวิเคราะห์ปัญหา

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

- 4 คะแนน (Lab12_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน `matching_sum(t, target_value)` เพื่อคืนค่า list ของจำนวนเต็มสองจำนวนใน tuple t ที่มีผลรวมเท่ากับจำนวนเต็ม $target_value$ ทั้งนี้สมาชิกใน tuple t จะเป็นจำนวนเต็มเท่านั้นและจะมีสมาชิกอย่างน้อย 1 ตัวเสมอ โดยหากไม่สามารถหาจำนวนสองจำนวนดังกล่าวได้ ให้คืนค่า list ว่าง และหากมีสมาชิกใน t มากกว่า 1 คู่ ที่สามารถเป็นคำตอบได้ ให้คืนค่าเพียงคำตอบเดียวเท่านั้น
Hint: การจับคู่สมาชิกทั้งหมด แล้วนำมาตรวจสอบผลบวกทีละคู่ จะทำให้เวลาบน Grader เกิน ควรแก้ปัญหาโดยใช้ collections ประเภท `set` หรือ `dict`

$$\begin{cases} (a1, a2) \\ key = a1 + a2 \end{cases}$$

นำไปใส่ใน list

①

ให้ p2a มีค่าน้อยกว่า p1a เสมอ

$p1a = \text{sorted}(p1)$

$p2a = \text{sorted}(p2)$

1. ① $p1a(\text{Value ต่ำสุด}) \times p2a(\text{Value ต่ำสุด})$
 $p1a(\text{key ต่ำสุด}) + p2a(\text{key ต่ำสุด})$
เก็บค่าใน result_dict

② $p1a(\text{Value ต่ำสุด}) \times p2a(\text{Value ที่ 2})$
 $p1a(\text{key ต่ำสุด}) + p2a(\text{key ที่ 2})$
เก็บค่าใน result_dict

2. ถ้า p2a ครบทุกตัวแล้ว → เปลี่ยน p1a เป็นตัวต่อไป

① $p1a(\text{Value ที่ 2}) \times p2a(\text{Value ต่ำสุด})$
 $p1a(\text{key ที่ 2}) + p2a(\text{key ต่ำสุด})$
เก็บค่าใน result_dict

ถ้า key เหมือนกัน เอา Value มาบวกกัน

Note: เมื่อเปลี่ยน ถ้า Value != 0 จะบวก

$p1a = \{2:2, 1:0, 0:3\}$

$p2a = \{1:4, 0:5\}$

① $2 \times 4 = 8, 2 + 1 = 3$

$k_dict = \{3:8\}$

② $2 \times 5 = 10, 2 + 0 = 2$
 $= \{3:8, 2:10\}$

① $0 \times 4 = 0 \rightarrow \text{ข้าม}$

① $3 \times 4 = 12, 0 + 1 = 1$
 $= \{3:8, 2:10, 1:12\}$

② $3 \times 5 = 15, 0 + 0 = 0$
 $= \{3:8, 2:10, 1:12, 0:15\}$

เป็นค่าที่รวมค่าแล้ว เพราะยังไม่ถึง ตัว key ที่ใหม่

② (5, 2, 3, 1, 4), 7

dict_ = {}

list_ = []

for i in t: 5
for j in t: 5 2 3 1 4

dict_[i+j] += list(i, j)

{0: [5, 5], 7: [5, 2], 8: [5, 3], 6: [5, 1], 9: [5, 4], ...}

for l in dict_[target_value]:

a += dict_[value]

return a

→ 90100 อินโดนีเซีย

- ## ชนิดข้อมูล

จำนวนดังกล่าวมีตัวอย่างดังนี้

המשפט 2.1

Hand-drawn diagram showing a map of Thailand with arrows pointing to different regions and their corresponding numbers. The regions are labeled in Thai: 31 (North), 32 (Northeast), 5 (Central), 4 (South), and 2 (East).

- ชนิดข้อมูล

Process by use in

จุดตัด/อง $n=2 \rightarrow$ แล้ว จะมี 7 แท่ง 1 ซัก

จ.รู้ได้ไงว่าตรงเป็น ๕๕

dict[3] = '66'

4 = 67

$$5 = 76$$

6 = '77'

$$\log_2 n = 1$$

$n = \frac{1}{2}; 2 : 6, 7 \rightarrow \text{pairs} \checkmark$

$$\log_2 n = 4$$

$n=4, 2^2: 66, 67, 76, 77$
 $0: 2 \rightarrow 0, 1: 2 \rightarrow 1, 2: 2 \rightarrow 0, 3: 2 \rightarrow 1$

12V5 74 $2^3 = 8$

$n = 7$; $2:666, 667, 676, 677, 766, 767, 776, 777$
 $0 \quad 1 \quad 2 \quad 3 \quad 4:4 \quad 5:4 \quad 6:4 \quad 7:4$

11
12
13
14

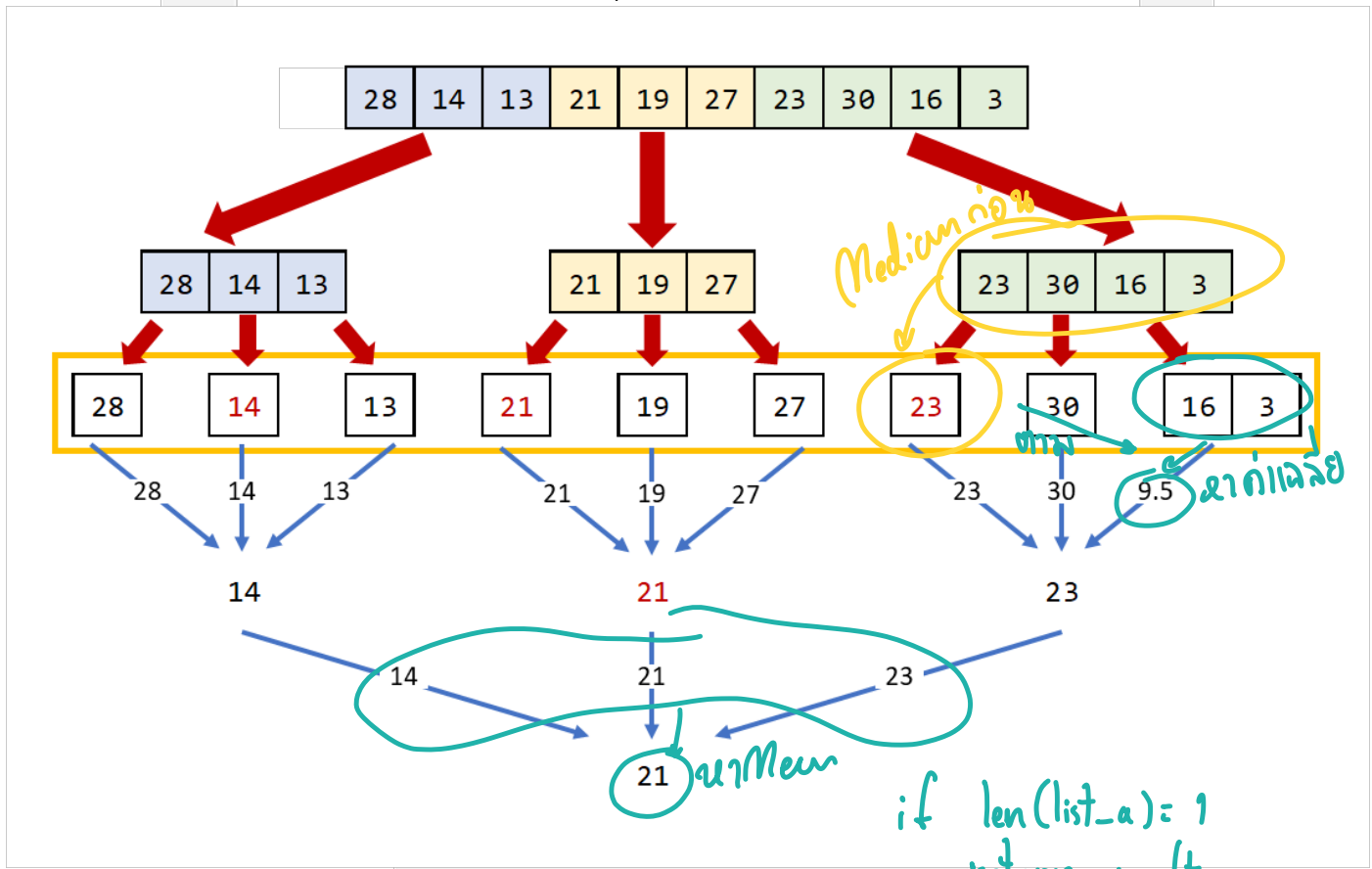
ของตัวที่ n ระบุ
 $a = \text{เลขหน้า} - 1$

$n=18 \rightarrow$ ความเป็นอิสระของตัวแปร

$a = \text{arraySize} - 1$
 $1 \rightarrow 3 \rightarrow 15$
 $l(0)+1 \quad l(1)+1 \quad (2)3+1 \quad l(7)+1$
 $19 - 2^{3-1} = 2^{3-2} = 2$
 $1011 \text{ array list } [666, 667]$
 $\text{min } 8$

4) 4 คะแนน (HW12_2_6XXXXXXX.py) ให้เขียนฟังก์ชัน `median_of_median(list_a)` เพื่อคืนค่ากลางของ element ใน `list_a` ซึ่งเป็น List ของจำนวนจริง (`list_a` ไม่เป็น List ว่าง) โดยคำนวณค่ากลางดังกล่าวด้วยวิธี Median of Median

การคำนวณ Median of Median เป็นการคำนวณค่ากลางแบบ recursive โดยมีเงื่อนไขคือหาก `list_a` มีความยาวตั้งแต่ 3 ขึ้นไป จะต้องแบ่ง `list_a` เป็น List ย่อย 3 List เท่าๆ กัน หากมีเศษจากการแบ่ง ให้นำไว้ List สุดท้ายเสมอ เช่น ถ้า `list_a` ยาว 40 จะต้องได้ความยาวทั้งสาม List ย่อยเป็น 13 13 และ 14 และแบ่งย่อยลงไปจนเหลือ List ที่มีความยาว 1 หรือ 2 จากนั้นให้คำนวณค่ามัธยฐานในแต่ละ List ย่อยก่อน แล้วจึงนำค่าที่ได้มาหาค่ามัธยฐานอีกที โดยกำหนดให้ค่ามัธยฐานของ List ที่มีความยาว 2 element คือค่าเฉลี่ยของทั้งสอง element นั้น ๆ ดังรูปด้านล่างค่าเฉลี่ยของ 16 และ 3 คือ 9.5 ทั้งนี้ไม่อนุญาตให้ใช้ฟังก์ชัน `sorted()` ในการแก้ปัญหา



Input	Output
[28, 14, 13, 21, 19, 27, 23, 30, 16, 3]	21.0

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

if len(list_a) == 1
return result
else:
if len(list_a) % 2 == 0:
แบ่งออกเป็น list ละ 3

Recursive

1. ~~return~~ 3 list to list

1.1 sind zwei differente un. aug.

2. ~~return~~ list to list

2.1 len = 3 → find max
min

max > 1 → return max

min > 1 → return min

max = min → return av

2.2 len > 3 → return

5) 4 คะแนน (HW12_3_6XXXXXXX.py) ให้เขียนฟังก์ชัน `subset_sum(set_a)` เพื่อคืนค่า list ของ ผลบวก สมาชิกของแต่ละเซตย่อยทั้งหมดของ `set_a` ซึ่งเซตย่อยดังกล่าวนั้นรวมถึงเซตว่าง (\emptyset) และ `set_a` เองด้วย โดยกำหนดให้ผลบวกของเซตว่าง คือ 0 ทั้งนี้ไม่อนุญาตให้ `import` โมดูลเพิ่มเติมในการแก้ปัญหา

ในทางคณิตศาสตร์ถ้า S เป็นเซต $\{1, 2, 3\}$ แล้วเซตย่อยของ S ได้แก่:

- $\{\}$ (อาจเขียนแทนด้วย \emptyset ได้แก่เซตว่าง)
- $\{1\}$
- $\{2\}$
- $\{3\}$
- $\{1, 2\} \rightarrow 1+2=3$
- $\{1, 3\} \rightarrow 1+3=4$
- $\{2, 3\} \rightarrow 2+3=5$
- $\{1, 2, 3\} \rightarrow 1+2+3=6$

`list-a = []`
`set-b = {}`
`for i in set-a:`
`for j in set-a:`

Input

{1, 2, 3}

Output

[0, 1, 2, 3, 3, 4, 5, 6]

การวิเคราะห์ปัญหา

• Input:

• Output:

(แสดงค่า)
(คืนค่า)

จำนวนข้อมูล	ชนิดข้อมูล
จำนวนข้อมูล	ชนิดข้อมูล
จำนวนข้อมูล	ชนิดข้อมูล

การส่งงาน

1. ลักษณะ/ลำดับข้อความของการรับค่า/แสดงผล จะต้องเป็นไปตามที่ระบุในตัวอย่างการ run
2. ไฟล์งานที่ส่ง จะต้องมีการแทรก comment ที่ต้นไฟล์ตามข้อกำหนดใน canvas รายวิชา
3. ไฟล์งานโปรแกรมที่ส่ง จะต้องมีการแทรก pseudocode เป็น comment ในแต่ละขั้นตอน
4. Upload ไฟล์ source code ตามที่ระบุในแต่ละข้อ ไปยังระบบตรวจให้คะแนนอัตโนมัติ <https://cmu.to/gdr223>