



กระบวนวิชา 204113

Lab	Feb 19
HW	Feb 22

## การบ้านปฏิบัติการ 6

### Sorting and Searching (20 คะแนน)

#### ข้อกำหนด

- i. การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อความสะดวกในการ import จาก Script อื่นๆ
  - ii. นักศึกษาสามารถสร้างฟังก์ชันย่อยต่างๆ เพิ่มเติมได้ตามความเหมาะสม
- 1) 5 คะแนน (Lab06\_1\_5XXXXXXX.py) ให้เขียนฟังก์ชัน `sort_date(list_x, show_step=False)` เพื่อเรียงลำดับสมาชิกของ `list_x` ด้วยวิธีเรียงลำดับแบบ Insertion Sort ซึ่งแต่ละสมาชิกจะเป็น String แทนวันที่ (อาจซ้ำกันได้) ในรูปแบบ "d/m/yyyy" เช่น "9/11/2016" กำหนดให้ฟังก์ชันทำงานแบบ Destructive โดยผลลัพธ์จะต้องเรียงลำดับวันที่จากน้อยไปมาก โดยมี Optional Parameter `show_step` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration

**Hint:** พิจารณาสร้างฟังก์ชัน Boolean `less_than(date1, date2)` เพื่อใช้เปรียบเทียบ String แทนวันที่

#### Function Call 1

```
list_x = ["11/1/2100", "5/12/1999", "19/1/2003", "11/9/2001"]
sort_date(list_x, show_step=True)
print("---")
print(list_x)
```

#### Output 1

```
1: ['5/12/1999', '11/1/2100', '19/1/2003', '11/9/2001']
2: ['5/12/1999', '19/1/2003', '11/1/2100', '11/9/2001']
3: ['5/12/1999', '11/9/2001', '19/1/2003', '11/1/2100']
---
['5/12/1999', '11/9/2001', '19/1/2003', '11/1/2100']
```

#### Function Call 2

```
list_x = ["11/1/2100", "5/12/1999", "19/1/2003", "11/9/2001"]
sort_date(list_x)
print("---")
print(list_x)
```

#### Output 2

```
---
['5/12/1999', '11/9/2001', '19/1/2003', '11/1/2100']
```

2) 5 คะแนน (Lab06\_2\_5XXXXXXX.py) กำหนดให้ *list\_x* เป็น List ที่มีสมาชิกเป็น tuple ของ String ในรูปแบบ (วันที่, เหตุการณ์) และเรียงลำดับไว้แล้วตามวันที่จากน้อยไปมาก

ให้เขียนฟังก์ชัน `search_event(list_x, key, show_step=False)` เพื่อค้นหา tuple ที่มีวันที่ตรงกับ *key* ในรูปแบบ "d/m/yyyy" ด้วยวิธีการค้นหาแบบ Binary Search ทั้งนี้ให้ถือว่าไม่มี Event ที่มีวันที่ซ้ำกัน โดยฟังก์ชันจะคืนค่า tuple ที่หาพบ หรือ None หากหาไม่พบ (ไม่พิจารณา 0 นำหน้าในเลขหนึ่งหลัก e.g. 01/01/10 = 1/1/10) และฟังก์ชันจะมี Optional Parameter *show\_step* เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration

#### Function Call 1

```
list_x = [("11/1/1900", "Event A"), ("5/12/2001", "Event B"),
          ("5/12/2002", "Event C"), ("21/8/2008", "Event D"),
          ("9/3/2013", "Event E"), ("11/3/2017", "Event F"),
          ("7/5/2019", "Event G"), ("29/2/2032", "Event H"),
          ("9/11/2042", "Event I")]
event = search_event(list_x, "29/02/2032")
print("---")
print(event)
```

#### Output 1

```
---
('29/2/2032', 'Event H')
```

#### Function Call 2

```
list_x = [("11/1/1900", "Event A"), ("5/12/2001", "Event B"),
          ("5/12/2002", "Event C"), ("21/8/2008", "Event D"),
          ("9/3/2013", "Event E"), ("11/3/2017", "Event F"),
          ("7/5/2019", "Event G"), ("29/2/2032", "Event H"),
          ("9/11/2042", "Event I")]
event = search_event(list_x, "23/08/2008", show_step=True)
print("---")
print(event)
```

#### Output 2

```
[4]: 9/3/2013
[1]: 5/12/2001
[2]: 5/12/2002
[3]: 21/8/2008
---
None
```

- 3) 5 คะแนน (HW06\_1\_5XXXXXXX.py) ให้เขียนฟังก์ชัน `radix_int(list_x, show_step=False)` เพื่อทำการเรียงลำดับสมาชิกจำนวนเต็มที่ไม่เป็นลบใน `list_x` จากน้อยไปมาก โดยใช้ Radix Sort Algorithm ทั้งนี้ฟังก์ชันจะทำงานแบบ destructive และไม่คืนค่า โดยมี Optional Parameter `show_step` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration

Function Call	Output
<pre>list_x = \ [19, 48, 175, 290, 873, 7, 43, 69] radix_int(list_x, True) print('-----') print(list_x)</pre>	<pre>[290, 873, 43, 175, 7, 48, 19, 69] [7, 19, 43, 48, 69, 873, 175, 290] [7, 19, 43, 48, 69, 175, 290, 873] ----- [7, 19, 43, 48, 69, 175, 290, 873]</pre>

Function Call	Output
<pre>list_x = \ [19, 48, 175, 290, 873, 7, 43, 69] radix_int(list_x) print('-----') print(list_x)</pre>	<pre>----- [7, 19, 43, 48, 69, 175, 290, 873]</pre>

- 4) 5 คะแนน (HW06\_2\_5XXXXXXX.py) ให้เขียนฟังก์ชัน `merge_sort(list_x, show_step=False)` เพื่อทำการเรียงลำดับสมาชิกใน Integer List `list_x` ตามลำดับจากน้อยไปมากโดยใช้ Merge Sort Algorithm ทั้งนี้ฟังก์ชันจะทำงานแบบ Destructive โดยมี Optional Parameter `show_step` เพื่อแสดง/ไม่แสดงขั้นตอนในแต่ละ Iteration

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

### การส่งงาน

1. ลักษณะ/ลำดับข้อความของการรับค่า/แสดงผล จะต้องเป็นไปตามที่ระบุในตัวอย่างการ run
2. ไฟล์งานที่ส่ง จะต้องมีการแทรก comment ที่ต้นไฟล์ตามข้อกำหนดใน website รายวิชา
3. ไฟล์งานโปรแกรมที่ส่ง จะต้องมีการแทรก pseudocode เป็น comment ในแต่ละขั้นตอน
4. Upload ไฟล์ source code ตามที่ระบุในแต่ละข้อ ไปยัง website ที่ใช้ส่งการบ้าน  
[http://hw.cs.science.cmu.ac.th/CS\\_HW/p204113.html](http://hw.cs.science.cmu.ac.th/CS_HW/p204113.html) ตาม section ที่นักศึกษาเรียน