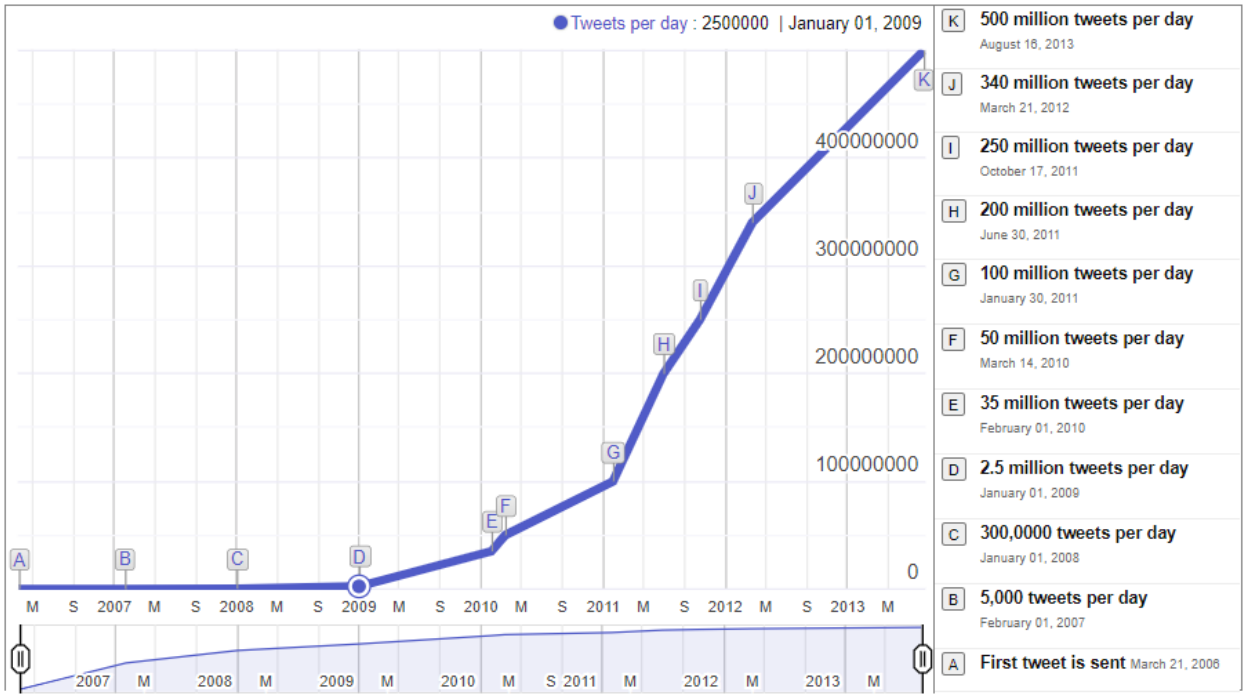
Assignment 12: Twitter Data Analytics



https://www.internetlivestats.com/twitter-statistics

Twitter เป็นบริการเครือข่ายสังคมที่มีผู้ใช้เขียนข้อความเผยแพร่มากมาย (ดังแสดงในรูปข้างบนนี้)

เว็บ https://developer.twitter.com ให้รายละเอียดในการเขียนโปรแกรมดึงข้อมูลหลายอย่างจากระบบมาวิเคราะห์ตามจินตนาการ สำหรับโจทย์ข้อนี้คงไม่ต้องไปดึงข้อมูลเอง แต่มีแฟ้มข้อมูลที่ดึงมาแล้ว และได้เปลี่ยนชื่อผู้ทวีต ข้อความที่ทวีต ไปเป็นรหัสผู้ใช้และรหัสข้อความทั้งหมด (เพื่อปกปิดอัตลักษณ์) แล้วบันทึกลงแฟ้มชื่อ `tweet_info.csv` ในรูปแบบง่าย ๆ ดังนี้

```
2020-10-18 15:51:05,user_7,text_30,0
2020-10-18 15:51:06,user_10,text_31,1
2020-10-18 15:51:07,user_8,text_32,1
2020-10-18 15:51:07,user_1,text_33,1
2020-10-18 15:51:13,user_10,text_34,1
2020-10-18 15:51:13,user_6,text_35,1
2020-10-18 15:51:14,user_1,text_36,1
```

แต่ละบรรทัดประกอบด้วย วันเวลา รหัสผู้ใช้ รหัสข้อความ และสถานะระบุเป็นการทวีตข้อความใหม่ (0) หรือรีทวีตข้อความอื่น (1) (ข้อมูลทั้งสี่คั่นด้วยเครื่องหมายจุลภาค) ฟังก์ชัน `read_tweets` ข้างล่างนี้อ่านข้อมูลในแฟ้มเก็บในลิสต์ของทูเปิล `[ (date_time, user_id, text_id, is_re_tweet), ... ]`

```
def read_tweets(filename):
    f = open(filename)
    tweets = [tuple(line.strip().split(','))
              for line in f.readlines()]
    f.close()
    return tweets

tweets = read_tweets('tweet_info.csv')
```

สำหรับการบ้านนี้ เราจะไม่สนใจข้อมูลส่วนขวาสุดที่ระบุว่าเป็นรีทวีตหรือไม่ คือ ไม่ว่าจะทวีตหรือรีทวีต ขอเรียกทั้งสองว่าเป็นการทวีต

งานที่ต้องทำ

ให้เขียน 4 ฟังก์ชันข้างล่างนี้

- **top_K_tweeters(tweets, K)**
  - รับ `tweets` เป็นลิสต์ของทูเปิล `[ (date_time, user_id, text_id, is_rt), ... ]`
    `K` เป็นจำนวนเต็มบวกระหว่าง 1 ถึง 50
  - คืน ลิสต์ของทูเปิล `[ (user_id, number_of_tweets), ... ]` ที่เก็บรหัสผู้ใช้และจำนวนการทวีตของผู้ใช้ที่มีจำนวนการทวีตมากที่สุด จำนวน `K` ราย (ถ้ามีข้อมูลไม่ถึง `K` ก็ให้มีเท่าที่มี) ลิสต์ผลลัพธ์นี้เรียงตามจำนวนทวีตจากมากไปน้อย ถ้าจำนวนทวีตเท่ากันให้เรียงตามรหัสผู้ใช้จากน้อยไปมาก
  - หมายเหตุ: ในกรณีที่ผู้ใช้คนหนึ่งทวีตข้อความเดิมหลายครั้ง ให้นับจำนวนครั้งตามที่ได้ทวีต

- **top_K_tweeters_in_S_seconds(tweets, K, S)**
  - รับ `tweets` เป็นลิสต์ของทูเปิล `[ (date_time, user_id, text_id, is_rt), ... ]`
    `K` เป็นจำนวนเต็มบวกระหว่าง 1 ถึง 50 และ `S` เป็นจำนวนเต็มบวก
  - คืน ลิสต์ของทูเปิล `[ (user_id, number_of_tweets), ... ]` ที่เก็บรหัสผู้ใช้และจำนวนการทวีตของผู้ใช้ที่ทวีตในช่วงเวลา `S` วินาทีเป็นจำนวนมากสุด จำนวน `K` ราย (ถ้ามีข้อมูลไม่ถึง `K` ก็ให้มีเท่าที่มี) ลิสต์ผลลัพธ์นี้เรียงตามจำนวนทวีตจากมากไปน้อย ถ้าจำนวนทวีตเท่ากันให้เรียงตามรหัสผู้ใช้จากน้อยไปมาก
  - หมายเหตุ: - ช่วงเวลา `S` วินาที ให้นับตั้งแต่เวลาที่สนใจ จนถึงเวลานั้นไปอีก (`S-1`) วินาที เช่น ถ้า `S = 3` และเริ่มสนใจที่ 2020-10-20 10:00:07 ก็คือ ตั้งแต่ 2020-10-20 10:00:07 ถึง 2020-10-20 10:00:09
    - ในกรณีที่ผู้ใช้คนหนึ่งทวีตข้อความเดิมหลายครั้ง ให้ถือว่าเป็นคนละทวีต

- **top_K_common_tweet_pairs(tweets, K)**
  - รับ `tweets` เป็นลิสต์ของทูเปิล `[ (date_time, user_id, text_id, is_rt), ... ]`
    `K` เป็นจำนวนเต็มบวกระหว่าง 1 ถึง 50
  - คืน ลิสต์ของทูเปิล `[ ((user_id1, user_id2), number_of_common_tweets), ... ]` ที่เก็บรหัสผู้ใช้สองรายและจำนวนการทวีตของผู้ใช้สองราย ที่ทวีตข้อความที่มีรหัสข้อความเดียวกัน เป็นจำนวนมากสุด จำนวน `K` ทูเปิล (ถ้ามีข้อมูลไม่ถึง `K` ก็ให้มีเท่าที่มี) ให้เก็บคู่รหัสผู้ใช้เป็นทูเปิลโดยรหัสผู้ใช้ทางซ้ายต้องน้อยกว่าตัวขวา ให้ลิสต์ผลลัพธ์นี้เรียงตามจำนวนทวีตจากมากไปน้อย ถ้าจำนวนทวีตเท่ากันให้เรียงตามทูเปิลรหัสผู้ใช้ จากน้อยไปมาก
  - หมายเหตุ: ในกรณีที่ผู้ใช้สองรายทวีตข้อความหนึ่งที่เหมือนกันหลายครั้ง ให้ถือว่าสองรายนี้ได้ทวีตข้อความนั้น เพียงหนึ่งทวีต

- **top_K_common_tweet_triples(tweets, K)**
  - รับ `tweets` เป็นลิสต์ของทูเปิล `[ (date_time, user_id, text_id, is_rt), ... ]`
    `K` เป็นจำนวนเต็มบวกระหว่าง 1 ถึง 50
  - คืน ลิสต์ของทูเปิล `[ ((user_id1, user_id2, user_id3), number_of_common_tweets), ... ]` ที่เก็บรหัสผู้ใช้สามรายและจำนวนการทวีตของผู้ใช้สามราย ที่ทวีตข้อความที่มีรหัสข้อความเดียวกัน เป็นจำนวนมากสุด จำนวน `K` ทูเปิล (ถ้ามีข้อมูลไม่ถึง `K` ก็ให้มีเท่าที่มี) ให้เก็บรหัสผู้ใช้ทั้งสามในทูเปิล เรียงจากน้อยไปมาก ให้ลิสต์ผลลัพธ์นี้เรียงตามจำนวนทวีตจากมากไปน้อย ถ้าจำนวนทวีตเท่ากันให้เรียงตามทูเปิลรหัสผู้ใช้ จากน้อยไปมาก
  - หมายเหตุ: ในกรณีที่ผู้ใช้สามรายทวีตข้อความหนึ่งที่เหมือนกันหลายครั้ง ให้ถือว่าสามรายนี้ได้ทวีตข้อความนั้น เพียงหนึ่งทวีต

- หมายเหตุ: **top_K** ของทุกฟังก์ชัน จะคืนลิสต์ที่มีขนาด `K` เท่านั้น (หรือน้อยกว่าในกรณีข้อมูลมีไม่ถึง) เราจะไม่สนใจตัวที่ `K+1` ที่มีจำนวนทวีตเท่ากับของตัวที่ `K` จะตัดข้อมูลให้เหลือ `K` ตัวเลยเป็นผลลัพธ์

ตัวอย่าง

ถ้าใช้แฟ้มข้อมูล tweet_info_mini.csv ข้างล่างนี้ เป็นข้อมูลทดสอบ

```
2020-10-18 15:00:00,user_0,text_0,0
2020-10-18 15:00:01,user_0,text_1,0
2020-10-18 15:00:01,user_0,text_2,0
2020-10-18 15:00:01,user_0,text_3,0
2020-10-18 15:00:01,user_0,text_4,0
2020-10-18 15:00:02,user_0,text_0,1
2020-10-18 15:00:02,user_0,text_4,1
2020-10-18 15:00:02,user_0,text_5,0
2020-10-18 15:00:02,user_0,text_6,0
2020-10-18 15:00:02,user_0,text_7,0
2020-10-18 15:00:03,user_0,text_8,0
2020-10-18 15:01:00,user_0,text_9,0
2020-10-18 15:00:10,user_1,text_0,1
2020-10-18 15:00:12,user_1,text_1,1
2020-10-18 15:00:22,user_1,text_2,1
2020-10-18 16:00:05,user_1,text_3,1
2020-10-19 13:00:04,user_1,text_7,1
2020-10-19 15:01:09,user_1,text_4,1
2020-10-19 15:10:10,user_1,text_5,1
2020-10-19 16:00:04,user_1,text_6,1
2020-10-18 16:00:05,user_2,text_3,1
2020-10-18 16:00:09,user_2,text_4,1
2020-10-18 16:00:10,user_2,text_0,1
2020-10-18 16:00:10,user_2,text_5,1
2020-10-18 16:00:12,user_2,text_2,1
2020-10-18 16:00:11,user_2,text_1,1
```

```
2020-10-23 15:00:10,user_10,text_5,1
2020-10-23 15:00:11,user_11,text_5,1
2020-10-23 15:00:12,user_12,text_5,1
2020-10-23 15:00:13,user_13,text_5,1
2020-10-23 15:00:11,user_14,text_5,1
2020-10-23 15:00:11,user_15,text_5,1
2020-10-23 15:00:12,user_16,text_5,1
2020-10-19 15:00:04,user_21,text_61,0
2020-10-19 15:00:04,user_21,text_71,0
2020-10-19 15:00:05,user_21,text_31,1
2020-10-19 15:00:09,user_21,text_41,1
2020-10-19 15:00:10,user_21,text_51,1
2020-10-19 15:00:22,user_21,text_21,1
2020-10-20 15:00:10,user_9,text_0,1
2020-10-20 15:00:10,user_9,text_51,1
2020-10-20 15:00:11,user_9,text_11,0
2020-10-20 15:00:11,user_9,text_21,1
2020-10-20 15:00:11,user_9,text_31,1
2020-10-20 15:00:12,user_9,text_41,1
2020-10-18 15:00:05,user_23,text_31,0
2020-10-18 15:00:09,user_23,text_41,0
2020-10-18 15:00:10,user_23,text_51,0
2020-10-18 15:00:22,user_23,text_21,0
```

จะได้ผลลัพธ์ของการเรียกใช้ฟังก์ชัน ข้างล่างนี้

	ผลลัพธ์

`tweets = read_tweets('tweet_info_mini.csv')`
`top_K_tweeters(tweets, 5)`
	`[('user_0', 12),`
	` ('user_1', 8),`
	` ('user_2', 6),`
	` ('user_21', 6),`
	` ('user_9', 6)]`

`tweets = read_tweets('tweet_info_mini.csv')`
`top_K_tweeters_in_S_seconds(tweets, 5, 3)`
	`[('user_0', 10),`
	` ('user_9', 6),`
	` ('user_2', 4),`
	` ('user_21', 3),`
	` ('user_1', 2)]`

`tweets = read_tweets('tweet_info_mini.csv')`
`top_K_common_tweet_pairs(tweets, 5)`
	`[(('user_0', 'user_1'), 8),`
	` (('user_0', 'user_2'), 6),`
	` (('user_1', 'user_2'), 6),`
	` (('user_21', 'user_23'), 4),`
	` (('user_21', 'user_9'), 4)]`

`tweets = read_tweets('tweet_info_mini.csv')`
`top_K_common_tweet_triples(tweets, 5)`
	`[(('user_0', 'user_1', 'user_2'), 6),`
	` (('user_21', 'user_23', 'user_9'), 4),`
	` (('user_0', 'user_1', 'user_10'), 1),`
	` (('user_0', 'user_1', 'user_11'), 1),`
	` (('user_0', 'user_1', 'user_12'), 1)]`

ข้อแนะนำ

ให้ตัวแปร **date_time** เก็บสตริงที่แทนวันเวลาในรูปแบบ `yyyy-mm-dd hh:mm:ss` เช่น `'2020-10-18 15:00:22'`

เราสามารถหาจำนวนวินาทีตั้งแต่วันเวลา `'1970-01-01 00:00:00'` จนถึง **date_time** ด้วยคำสั่งข้างล่างนี้

```
epoch = datetime.datetime.strptime(date_time, '%Y-%m-%d %H:%M:%S').timestamp()
```

อย่าลืมว่า ต้อง `import datetime` ก่อนด้วย (หมายเหตุ: ในวงการคอมพิวเตอร์เรียกจำนวนวินาทีนี้ว่า epoch)

ในกรณีที่เขียนฟังก์ชันเสร็จ ควรทดสอบกับ tweet_info_mini.csv ก่อน หากได้ผลถูกต้อง จึงดาวน์โหลดแฟ้ม tweet_info.zip แล้ว unzip ได้ tweet_info.csv ที่มีข้อมูล 196,632 รายการ ควรหาผลลัพธ์ของทั้ง 4 ฟังก์ชันได้ ในเวลารวมประมาณไม่เกิน 5 วินาที

อย่าลืม

- ห้ามเขียนคำสั่งในฟังก์ชันที่ไปใช้ตัวแปรที่อยู่นอกฟังก์ชัน (หรือที่เรียกว่าตัวแปร `global`)
- อนุญาตให้เพิ่มคำสั่งในบริเวณพื้นสีเขียวเท่านั้น จะเขียนฟังก์ชันเพิ่มเติมก็ได้
- ตั้งชื่อแฟ้ม ให้ถูกต้องตามที่เขียนใน `CourseVille`
- เพิ่มเติมข้อมูลใน `comment` ในรูปแบบเดียวกับการบ้านก่อนนี้ให้ตรงตามความจริง
- ก่อนส่งโปรแกรม ควรสั่งทำงานโปรแกรมที่ส่งอีกครั้ง ว่าทำงานได้ตามที่ต้องการ
- ไม่อนุญาตให้ `import` คลังคำสั่งใด ๆ เพิ่มเติม (นอกจาก `datetime` ที่เขียนให้แล้ว)

download code นี้ได้

```
# Prog-12: Tweeter Data Analytics
# Fill in your ID & Name
# ...
# Declare that you do this by yourself

import datetime

def to_epoch(date_time):
    d = datetime.datetime.strptime(date_time, '%Y-%m-%d %H:%M:%S')
    return d.timestamp()

def read_tweets(filename):
    f = open(filename)
    tweets = [tuple(line.strip().split(','))
              for line in f.readlines()]
    f.close()
    return tweets

def prt(x):
    print('\n'.join(str(e) for e in x))
    print('-----------------')

def test(filename, K, S):
    tweets = read_tweets(filename)
    print(filename, 'K=', K, 'S=', S)
    prt( top_K_tweeters(tweets, K) )
    prt( top_K_tweeters_in_S_seconds(tweets, K, S) )
    prt( top_K_common_tweet_pairs(tweets, K) )
    prt( top_K_common_tweet_triples(tweets, K) )

def main():
    test('tweet_info_mini.csv', 5, 3)
    test('tweet_info.csv', 10, 24*60*60)
#---------------------------------

def top_K_tweeters(tweets, K):

    return []
#---------------------------------
def top_K_tweeters_in_S_seconds(tweets, K, S):

    return []
#---------------------------------
def top_K_common_tweet_pairs(tweets, K):

    return []
#---------------------------------
def top_K_common_tweet_triples(tweets, K):

    return []

#---------------------------------
main()
```

ไม่เพิ่ม ลบ หรือ เปลี่ยนแปลง
บริเวณคำสั่ง สีแดง เด็ดขาด

ไม่ใช้ตัวแปรใด ๆ ที่อยู่นอกฟังก์ชัน
สามารถเพิ่มฟังก์ชันเสริมอื่น ๆ ได้