執行環境：mac terminal (using vscode)

執行步驟：python Main.py 名稱(直接打名稱即可，不用去空白)

程式碼說明：

* 執行時是用argv，先把名稱先組起來 -- @a
* 再搜尋對應網址的html –- @b
* 判斷是否有沒有下一頁，找是否有無“pagination-list” –- @c
* 如果有，則先取出“下一頁的網址” –- @d
* 如果沒有，也把原本網址存在跟“下一頁的網址”一樣的變數內 –- @e
* 之後用逐個網址去取出html –- @f
* 因為搜尋名字可能會搜尋出相似的作者但不是正確的作者名，所以必須先取出所有co-worker，確認是否有要的名字在裡面，才能繼續下去 –- @h
* 取出所有的年份之後，再利用dictionary去存進data\_Problem\_1中 –- @i
* 取出所有的作者之後，再利用dictionary去存進data\_Problem\_2中 –- @j
* 輸出problem 2，利用sort去排順序，並把搜尋作者過濾掉 –- @k
* 輸出problem 1柱狀圖，利用sort去排順序 –- @l

import matplotlib.pyplot as plt

import sys

import re

import urllib.request

##### get particular html #####

original\_Author = sys.argv[1] ## @a

author = sys.argv[1]

if len(sys.argv) is not 2:

for inputAuthor in sys.argv[2:]:

author = author + "+" + inputAuthor

original\_Author = original\_Author + " " + inputAuthor

url = "https://arxiv.org/search/?query=" + author + "&searchtype=author&abstracts=show&order=-announced\_date\_first&size=50"

content = urllib.request.urlopen(url) ## @b

html\_str = content.read().decode("utf-8") # get all html

##### if have next page #####

is\_Next\_Page = False

target\_Url = []

try: ## @c

is\_Next\_Page = True

nextPage\_Pattarn = "pagination-list[\s\S]\*?</ul>"

nextPage\_Result = re.findall(nextPage\_Pattarn, html\_str)

except:

is\_Next\_Page = False

if is\_Next\_Page is True: ## @d

## get url

tmp\_Result = nextPage\_Result[0].split('pagination-list">')[1].split("</a>")[0:-1] #[0:-1] get rid of the last useless data

for tmp in tmp\_Result:

tmp\_Url = tmp.split('<a href=')[1].split("class")[0].strip()[1:-1]

target\_Url.append("https://arxiv.org" + tmp\_Url.replace("amp;", ''))

else:

target\_Url.append(url) ##@e

data\_Problem\_1 = {}

data\_Problem\_2 = {}

print("[ Author: " + author + " ]")

for tmp\_Url in target\_Url: ## @f

## get next page url

if is\_Next\_Page is True:

content = urllib.request.urlopen(tmp\_Url)

html\_str = content.read().decode("utf-8") # get all html

pattarn = 'Authors:</span>[\s\S]\*?</li>' # get Name of author

result = re.findall(pattarn, html\_str) ## @h

for r1 in result:

name = r1.split('</p>')[0]

name = name.split('</a>')[:-1]

## check if the source is right author

tmp\_Name\_List = [] ## @h get name list

for n in name:

tmp\_Name = n.split('">')[1].strip()

tmp\_Name\_List.append(tmp\_Name)

## @h if not then continue

if not(original\_Author in tmp\_Name\_List): continue # if not the right author

## get problem 1 data

## @i

pattarn1 = "originally announced</span>[\s\S]\*?</p>" # get year

result1 = re.findall(pattarn1, r1)

for r2 in result1:

## @i 取出年份

year = r2.split('</span>')[1].split(".")[0].strip().split(" ")[1].strip()

## @i 將資料加進dictionary

if year in data\_Problem\_1:

data\_Problem\_1[year] = data\_Problem\_1[year] + 1

else:

data\_Problem\_1[year] = 1

## get problem 2 data

## @j 因為前面要確認名字，因此沿用前面抓出的name list即可，將資料加進dictionary

for nn in tmp\_Name\_List:

if nn in data\_Problem\_2:

data\_Problem\_2[nn] = data\_Problem\_2[nn] + 1

else:

data\_Problem\_2[nn] = 1

## Print Problem 2

## @k

for data in sorted(data\_Problem\_2):

if not(data == original\_Author): ## @k過濾掉搜尋的作者

print("[" + data + "]: " + str(data\_Problem\_2[data]) + " times")

## Print Problem 1

## @l x軸為key，y軸為value

plt.bar(sorted(data\_Problem\_1.keys()), data\_Problem\_1.values())

plt.show()