

執行環境：皆在 windows 10 的 spyder 裡執行 (Python3.7)

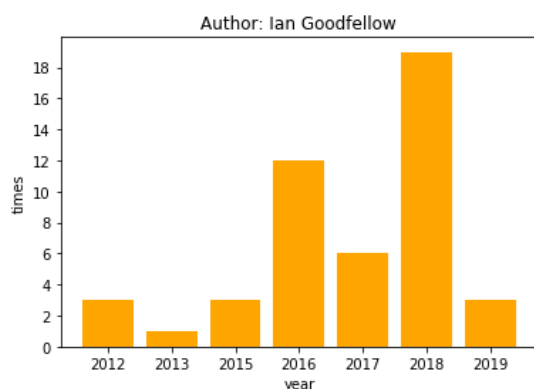
Question 1

執行 q1.py 後，需輸入作者的名字，而後會出現 Please wait the result ...，等待結果爬完後，即會出現 bar chart。

範例 Output 為下圖：

```
In [20]: runfile('D:/Programming Language/HW2/q1.py', wdir='D:/Programming Language/HW2')
```

```
Input Author:Ian+Goodfellow  
Please wait the result ...
```



程式碼說明：

在使用者輸入作者名字後，便會開始以作者姓名開始進行爬蟲，而網址預設一頁的顯示結果數量為 200(減少換頁次數，加速輸出速度)，若遇到有下一頁的情況，則會繼續前往下一頁抓取資料(是否有下一頁以 next_page 這個參數來判斷，假如 next_page 為 None 則表示已無下一頁)。而搜尋到的每篇論文都會檢查 author 欄位裡面是否有和 input 相同的 author(以 author_exist 這個參數來判斷，

`None` 表示此篇論文的 `author` 和 `input` 的 `author` 不相同，因此不列入計算)，若 `author_exist` 不為 `None`，才獲取 `originally announced` 的年份後，將其記錄在 `Record` 這個 `dictionary` 裡面(`key` 為年分，`value` 為次數)，若該年分已存在此 `dictionary` 裡，則將他的 `value+1`，若不存在，將其年份加入並把 `value` 設為 1。把各個 `paper` 的年份都記錄完後，按照年分進行 `Sort`(讓 `bar chart` 依照年分繪出)，而後利用 `matplotlib` 將 `Record` 的資料用 `bar chart` 畫出。

Question 2

執行 `q2.py` 後，需輸入作者的名字，而後會出現 `Please wait the result ...`，等待結果爬完後，即會印出 `co-author` 的結果。

範例 `Output` 為下圖：

```
In [3]: runfile('D:/Programming Language/HW2/q2.py', wdir='D:/Programming Language/HW2')
```

```
Input Author:Alex+Shee
Please wait the result ...
```

```
Chris Pal: 1 times
Misha Benjamin: 1 times
Negar Rostamzadeh: 1 times
Paul Gagnon: 1 times
Yoshua Bengio: 1 times
```

程式碼說明：

在使用者輸入作者名字後，便會開始以作者姓名開始進行爬蟲，而網

址預設一頁的顯示結果數量為 200(減少換頁次數,加速輸出速度),若遇到有下一頁的情況,則會繼續前往下一頁抓取資料(是否有下一頁以 `next_page` 這個參數來判斷,假如 `next_page` 為 `None` 則表示已無下一頁),而搜尋到的每篇論文都會檢查 `author` 欄位裡面是否有和 `input` 相同的 `author`(以 `author_exist` 這個參數來判斷,`None` 表示此篇論文的 `author` 和 `input` 的 `author` 不相同,因此不列入計算),若 `author_exist` 不為 `None`,才將該篇論文的 `co-author` 資訊記錄在 `Record` 這個 `dictionary` 裡面(`key` 為 `co-author` 姓名,`value` 為次數),若該姓名已存在此 `dictionary` 裡,則將他的 `value+1`,若不存在,將其姓名加入並把 `value` 設為 1。把各個 `paper` 的 `co-author` 姓名都記錄完後,按照姓名進行 `Sort`,再將 `co-author` 姓名(`key`)以及相對應的次數(`value`)印出來(不包含一開始 `input` 的作者)。