Report for pagerank, hits and simrank

- Find a way (e.g., add/delete some links) to increase hub, authority, and PageRank of Node 1 in first 3 graphs respectively.

➢ PageRank：

對於每一個 dataset，增加連接到 node 1 的其他 node 的數量.
用戶停留在 node 1 的概率要比停留在其他的概率高則 PR 越高

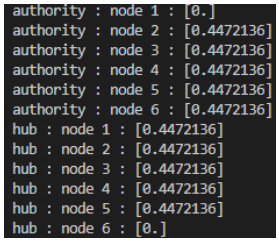
➢ Hit:

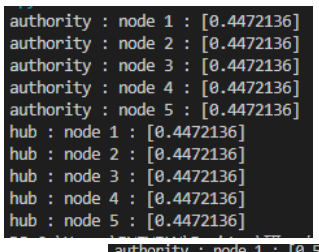
對於 hub, 增加 node 1 連接到其他 nodes 的數量. 因為一個高品質的 hub 頁面會指向很多高品質的 authority 頁面。

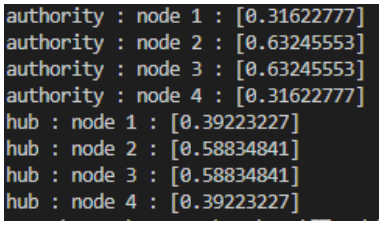
對於 authority, 增加連接到 node 1 的其他 node 的數量, 因為一個高品質的 authority 頁面會被很多高品質的 hub 頁面所指向。

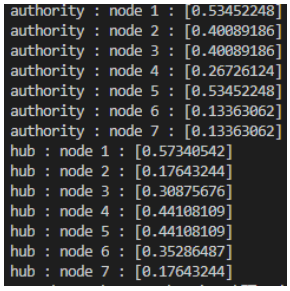
- Please describe and analysis your results for each algorithm in graph.
- Computation performance analysis

➢ Hits：建立一個 nodes 間指向關係的矩陣 M，authority 是 M 的轉置與轉換矩陣的乘積，hub 根據 authority 與 M 的乘積來得到值。重複反覆運算 authority 和 hub 的結果，當前一時刻的結果與當前結果相差很小時，反覆運算停止。









a）以上分別是 graph 1,2,3,4 的結果，graph 1 中只有 node 1 沒有指向他的網頁，重要度極低，authority 為 0; node 6 沒有指向其他的網頁，不是一個合格的 hub，所以 hub 為 0。Graph 2 是一個迴圈，所有 node 屬性都一樣，所以 authority 和 hub 都相同。Graph 3 中 node 2 和 3 的指向比較多，所以相對的 authority 和 hub 都比其他的更高。 Graph 4 中 node 1 和 5 指向自己的 link 多，authority 高，node 1 擁有最多指向其他網頁的連接，hub 最高。其他的 graph 以此類推。

b）隨著 node 數目越來越多，互相指向的 link 越來越多，節點關係越複雜，計算時間會得變長。Graph 1-4 的計算時間都在 0.001~0.002 之間，Graph 5 400+ nodes 需要 1.092s，graph 6 1200+ nodes 需要 6.07s，IBM data 需要 21.54s。演算法的時空複雜度分別為 O（n^2）和 O（n）。

- PageRank: 建立一個 nodes 間指向關係的矩陣，預先給每個網頁一個 PR 值 1/N(網頁的數量)作為初始值矩陣 V0，然後將這些連結以平均的概率表示成一個概率轉移向量 T，T*V0 得到用戶訪問到各個網頁的概率矩陣 V1。用戶從一個網頁轉至另一個網頁的過程中，會以一定的概率不點擊當前網頁中的連結，而是訪問一個自己重新輸入的新位址，所以設定 damping_factor = 0.15，為從新網址離開的概率，1-damping_factor 為用戶繼續點擊當前網頁中的連結的概率。V1*damping_factor*T 和 V0*（1-damping_factor）之和，持續計算到第 n 次跳轉之後的概率分佈矩陣。同樣的，當當前一時刻的結果與當前結果相差很小時，反覆運算停止。

```
pageNumber 1 の pageRank :  0.025
pageNumber 2 の pageRank :  0.02875
pageNumber 3 の pageRank :  0.02931
pageNumber 4 の pageRank :  0.0294
pageNumber 5 の pageRank :  0.02941
pageNumber 6 の pageRank :  0.02941
```

```
pageNumber 1 の pageRank :  0.03529
pageNumber 2 の pageRank :  0.03529
pageNumber 3 の pageRank :  0.03529
pageNumber 4 の pageRank :  0.03529
pageNumber 5 の pageRank :  0.03529
```

```
pageNumber 1 の pageRank :  0.04104
pageNumber 2 の pageRank :  0.04719
pageNumber 3 の pageRank :  0.04719
pageNumber 4 の pageRank :  0.04104
```

```
pageNumber 1 の pageRank :  0.02983
pageNumber 2 の pageRank :  0.02534
pageNumber 3 の pageRank :  0.02456
pageNumber 4 の pageRank :  0.02339
pageNumber 5 の pageRank :  0.02853
pageNumber 6 の pageRank :  0.0225
pageNumber 7 の pageRank :  0.02232
```

a）以上分別是 graph 1,2,3,4 的結果，pagerank 意味著如果一個網頁被很多其他網頁連結到的話說明這個網頁比較重要，

PageRank 值會相對較高，如果一個 PageRank 值很高的網頁連結到一個其他的網頁，那麼被連結到的網頁的 PageRank 值會相應地因此而提高。Graph 中的 node 1 沒有被其他頁面連接到，所以他的 PR 最低，其他網頁被前一個網頁連接到，隨著前置網頁的增加，網頁的 PR 值可能會相對增加。同樣的，graph 2 是一個迴圈所以每個 node 的 PR 值都一樣。在 graph4 中，因為 node 1，5 出現在連結裡的次數比較多，所以他們的 PR 相對更高。

b）同樣的，隨著 node 數目越來越多，互相指向的 link 越來越多，節點關係越複雜，計算時間會得變長。Graph 1-4 的計算時間都在 0.001~0.002 之間，Graph 5 400+ nodes 需要 1.10s，graph 6 1200+ nodes 需要 5.26s。演算法的時空複雜度分別為 O（n^2）和 O（n）。

- Simrank：建立一個 graph 表現 nodes 之間的關係，這裡除了自己生成矩陣也可以用 import networkx 來生成 graph。若 node 和 node 之間沒有 in-neighbor 則 S 為 0，若 node 相同，則 S 為 1。

a）以 graph 4 為例，node3 和 node6 的打分大，可能是因為他們共同被 node1 和 node5 指向，相似度更高；node7 和 node2 的打分大，可能是因為他們共同被 node1 指向且自身又指向 node1。

	1	2	3	4	5	6	7
1	1.	0.133	0.267	0.067	0.16	0.133	0.
2	0.133	1.0	0.	0.1	0.08	0.2	0.4
3	0.267	0	1.	0.	0.16	0.4	0.
4	0.067	0.1	0.	1.	0.08	0.1	0.2
5	0.16	0.08	0.16	0.08	1.	0.08	0.
6	0.133	0.2	0.4	0.1	0.08	1.	0.4
7	0.	0.4	0.	0.2	0.	0.4	1.

a）b）Graph 1-4 的計算時間都在 0.001~0.002 之間，Graph 5 400+ nodes 需要 0.09s，graph 6 1200+ nodes 需要 9.75s。演算法的時空複雜度分別為 O（n^3）和 O（n）。

- Discussion (what you learned from this project and your comments about this project)

  這幾個演算法都可以說是基於 gragh 的 web 排名演算法，所以首先需要把圖以矩陣的方式表現出來。通過結果我認為 PageRank 的穩定性比 hits 更高。Hits 演算法只是單純計算 authority 和 hub，如果在網頁裡增加連接到高品質網頁的 links，他的 hits 得分會升高更多，所以這樣會使網頁排名更容易被操控。Simrank 演算法相對更難實現一點，演算法能得到兩個 website 之間的關係，比 hits 和 pagerank 這兩個只能得到一個網頁權重的會更加準確。瞭解連結分析讓我們在推廣網站的時候更好地製造一些更能被搜尋引擎找到的網頁。

- More limitations about link analysis algorithms?

  給網頁進行打分的時候，例如用 PageRank 打分，舊的頁面等級會比新頁面高。受到網頁更新的影響，同一個頁面在不同的時間得到的打分也會有所不同。

- Can link analysis algorithms really find the “important” pages from Web?

  link analysis 只能找到部分重要的網頁，這些網頁一定要是知名活躍的。依靠網頁導向尋找大部分只能找到一些相似相依的網頁。

- What are practical issues when implement these algorithms in a real Web?

  現實問題比如，很多網站都不會在自己的頁面上寫入其他高品質網站的 link，並且很多網頁也不會連到一些不熱門的網頁，因為這些網站很可能是相互競爭的關係。此時非熱門網站所能做的就是每天更新網頁內容提交給搜尋引擎，更新自己網頁的關鍵字與時事熱點與時俱進，link analysis 就不怎麼有用了。

- Any new idea about the link analysis algorithm?

對於 PageRank，可以考慮為 graph 裡 node 之間的邊設置權重，對於 node 之間的邊連接越多，權重越大、

- What is the effect of “C” parameter in SimRank?

C 是一個阻尼係數，兩個節點間相隔的節點越多，相似度衰減的越厲害。