Project3

Q36074251 董濟鈐

本次作業要實作三種演算法：HITS、PageRank、SimRank，並對三種方法進行討論比較。Dataset 的部分需要 project1 的 transaction，再利用 code 將其轉換成此次作業所需的格式。

1. Data

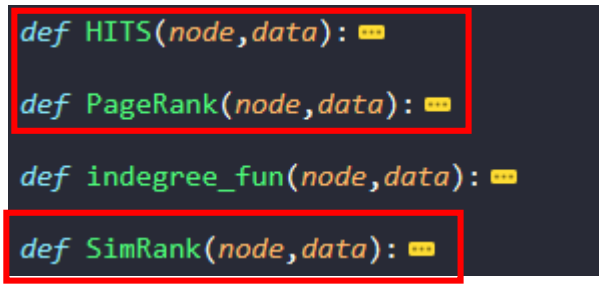
除了老師提供的 6 組 dataset 外，要再自行產生 2 組 data。檔名及內容說明如下：

(1) graph_7.txt：為 Project1 的 IBM transaction 利用 convert.py 轉換而成，共 9 個 items，9 組 transactions，bi-directed。

(2) graph_8.txt：為 graph7 使用的 transaction 的 association rules。

2. Implement of three algorithms

三種演算法皆寫在 main.py 中，利用 for 迴圈讓 8 的檔案都有用每種演算法實作，並將結果印出來方便觀察。



←Three algorithm in main.py

- **Details**
  - **HITS**

    每個 node 的 hub 是 children 的 authority 值的總和，authority 則是來自 parents 的 hub 值加總，且兩個數值皆有經過標準化。每次 iteration 後都會與上次 iteration 的結果做比較，若相差的值小於我們所設定的 threshold，則停止並輸出結果。

  - **PageRank**

    設定的 damping factor 為 1.5，將 pagerank 的利用 code 實作後，對每個 node 的結果進行標準化後，與前次 iteration 的結果作比較，若相差的值小於預設的 threshold，則結束 iteration 並輸出結果。

- **SimRank**

  用來計算網路圖中兩點間的相似度，與前兩個演算法的執行方式相同，每次 iteration 更新 node 相似度，若與前次的結果相差不多則停止。

- **Parameter 設定**

```
C:\Users\user\Desktop\data mining\hw3>python main.py 100 0.0001 0.8
```

每個 algorithm 共用相同的參數，以便於比較。

sys.argv[1]：iteration 的最大次數 (100)

sys.argv[2]：threshold，連續兩次的 iteration 結果相差若小於此數值則停止 (0.0001)

sys.argv[3]：decay factor C，用於 SimRank 演算法 (0.8)

- **Result**

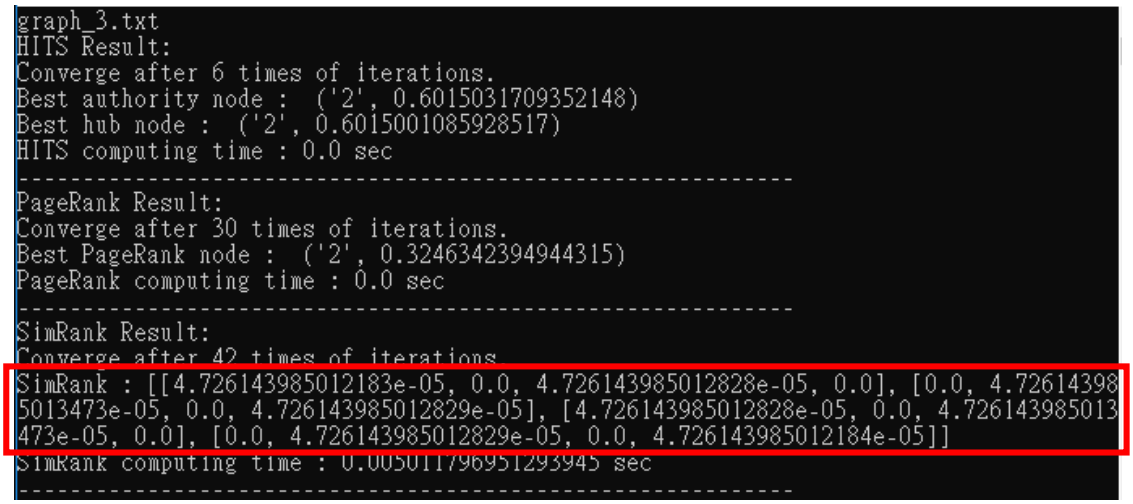
HITS 的最佳結果：

	authority		hub	
HITS	Best node	value	Best node	value
graph_1	2	0.447	1	0.447
graph_2	1	0.447	1	0.447
graph_3	2	0.602	2	0.602
graph_4	5	0.501	1	0.646
graph_5	61	0.491	274	0.192
graph_6	1151	0.275	171	0.156
graph_7	0	0.316	0	0.316
graph_8	8	1.0	0	0.707

PageRan 的最佳結果：

PageRank	Best node	value
graph_1	5	0.234
graph_2	1	0.200
graph_3	2	0.325
graph_4	1	0.280
graph_5	61	0.014
graph_6	1052	0.004
graph_7	0	0.100
graph_8	8	0.487

SimRank 結果用 list 表示與每點的相似度，下圖以 graph3 為例：



三種演算的 iteration 次數與執行時間：

	HITS_iter	HITS_time	PG_ iter	PG_ time	SR_ iter	SR_ time
graph_1	2	0.0	17	0.0	46	0.014
graph_2	2	0.0	14	0.0	43	0.011
graph_3	6	0.0	30	0.0	42	0.006
graph_4	11	0.0	39	0.0	46	0.030
graph_5	11	1.33	46	413.34	Run out of 3 hours	
graph_6	58	88.63	49	8563.48		
graph_7	2	0.0	33	0.004	Only first 5 graphs	
graph_8	2	0.0	26	0.001		

- **Discussion**

(1) 由上述結果的表可以發現，Performance：HITS>PageRank>SimRank。

(2) SimRank 的 complexity 很高($O(n^3)$)，以至於在處理很多個 nodes 時會花費許多時間。

(3) 若 threshold 設定的值減少，所需的時間便有可能會增加。

(4) 從所有 graph 的結果中發現 SimRrank 計算出的兩點相似度幾乎都不高，猜想可能是 C 沒有設定好抑或是他需要更多次的 iteration 才能得到更好的結果。

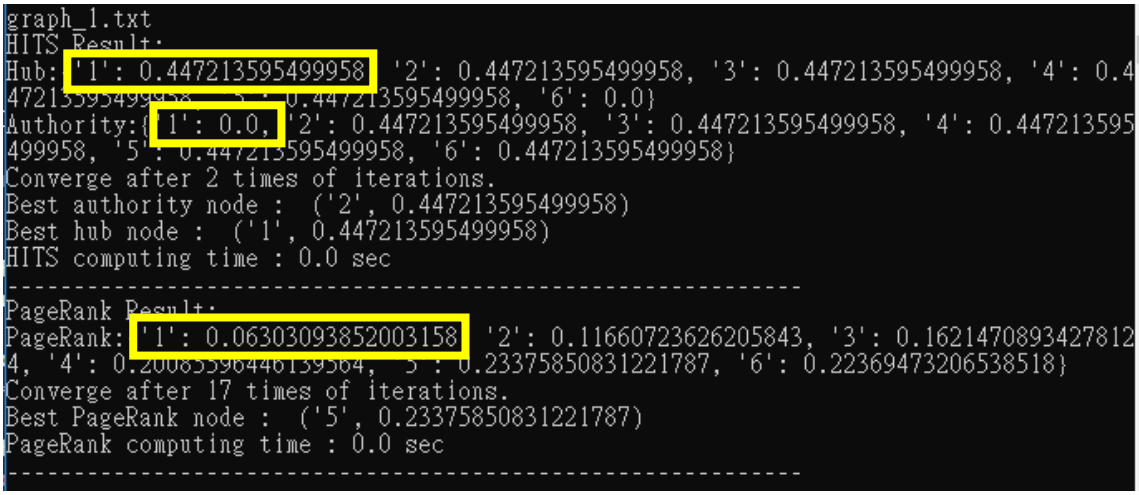
3. **Find a way to increase hub, authority and PageRank of Node 1 in first 3 graphs respectively.**

(1) Increase hub：Hub的計算方式為所連結到的點的authority相加，增加node 1連到其他nodes，便可增加node 1的hub。

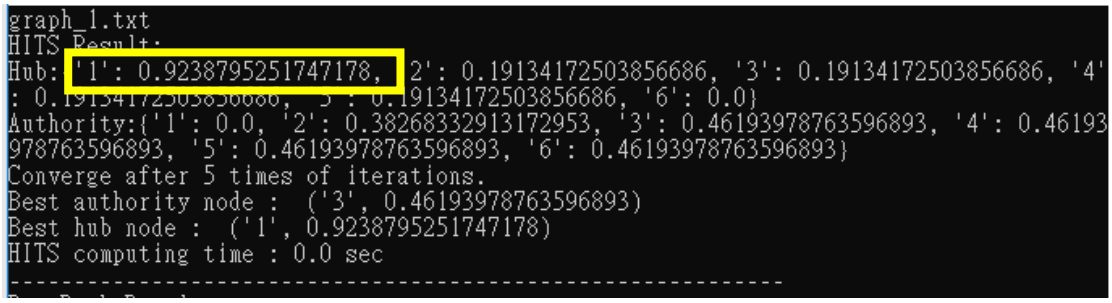
(2) Increase authority、PageRank：authority計算方法為parents的hub值加總，，PageRank的計算方式也是與連結到本身的node有關，因此增加其他nodes連到node 1的邊，可以同時增加authority與PageRank。

(3) Result (皆以graph1為例)

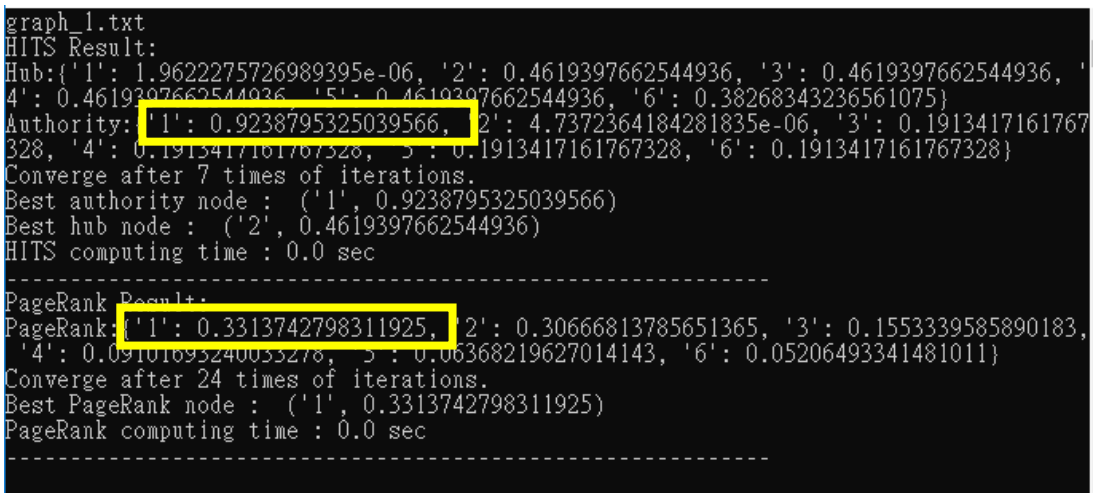
原本的結果：



Increase hub：



Increase authority and PageRank：



三個graph的結果比較：

	original hub	new hub	original authority	new authority	original PageRank	new PageRank
graph1	0.447	0.924	0	0.924	0.063	0.331
graph2	0.447	0.910	0.447	0.910	0.200	0.341
graph3	0.372	0.759	0.372	0.759	0.175	0.301

4. Questions

(1) **More limitations about link analysis algorithms**

大部分的演算法中，圖中的 nodes 與 structure 無法對應到 collection 中最相關的 page。

(2) **Can link analysis algorithms really find the "important" pages from Web?**

如上題所述，演算法沒辦法很好的找到 important pages。就比如我們造訪一個網頁，頁面中的廣告由該公司決定出現的 priority，因此便無法有效地找到 important pages。

(3) **What are practical issues when implement these algorithms in a real Web?**

生活中最常見的便是使用 search engine 了吧，利用這些演算法幫我們所需的相關資訊。

(4) **What is the effect of "C" parameter in SimRank?**

前面所述結果所使用的 C 皆為 0.8，改變 C 的值來觀察結果的變化，以 graph3 為例：

由上圖結果可以發現當 C 值越小，達到我們想要結果的 iteration 次數也會跟著減少，因此我想 C 是一個可以決定是否快速收斂的指標吧。