Project3: Link Analysis

E54036219 張富嘉

I. Software usage

這次 Link Analysis 報告中我用 python 實作了 HITS、PageRank 和 SimRank 演算法，並透過此算法去分析 8 張大小形狀不一的圖，以下為程式的使用方法：

```
./Link-Analysis.py -f <datafile> -m method
```

<datafile>是 8 張圖的檔案路徑，分別放置於 hw3dataset 資料夾中，method 是使用的方法，分別為”hits”、”pagerank”和”simrank”。程式執行後會根據所指定的 method 計算結果，並產生一個 output file，紀錄計算的結果。

II. Implementation Detail

HITS

Hits 算法的目的在於計算 Authority 和 Hubness 值，算法如下：

1. 賦予每個節點的 Authority 值為 1
2. 計算各節點 Hubness 值：$H_t(v) = \sum_{w \in ch[v]} A_{t-1}(w)$
3. 計算各節點 Authority 值：$A_t(v) = \sum_{w \in pa[v]} H_{t-1}(w)$
4. Normalize Hubness and Authority
5. 計算ε($\varepsilon = ||A_t - A_{t-1}||$)值，若ε > threshold則回到(2)，否則結束

PageRank

PageRank 算法如下：

1. 初始化轉移矩陣M：若節點 j 有 k 個 out-links，每個連結指向節點 i，則$M_{ij} = 1/k$
2. 初始化 PageRank：若有 N 個節點，則美個節點的初始機率相同，為$1/N$
3. 進行一次迭代：$[\text{PageRank}]^T = \text{DM}\ [\text{PageRank}]^T + (1 - \text{D})\text{e}$，其中 D 為 damping factor，e 為$1/N$
4. 計算ε($\varepsilon = ||\text{PageRank}_t - \text{PageRank}_{t-1}||$)值，若ε > threshold則回到(3)，否則結束

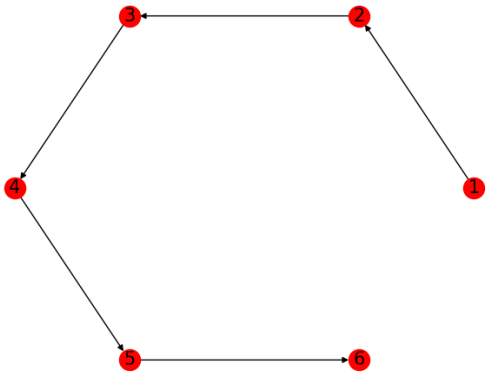
SimRank

SimRank 算法如下：

1. 初始化 SimRank Matrix：Identity Matrix: S，S_{ab}表示 a,b 的相似度值
2. 初始化轉移矩陣W：若存在一條連結從 a 指向 b，則$W_{ab} = \frac{1}{|I(b)|}$，否則為 0
3. 開始迭代：$\text{S} = \text{C}(W^T\text{SW}) + (1 - \text{C})\text{I}$，C 為 decay factor
4. 將 S 的對角線改為 1，因為跟自己的相關性為 1
5. 計算ε($\varepsilon = ||\text{S}_t - \text{S}_{t-1}||$)值，若ε > threshold則回到(3)，否則結束

III. Result Analysis and Discussion

Graph_1：

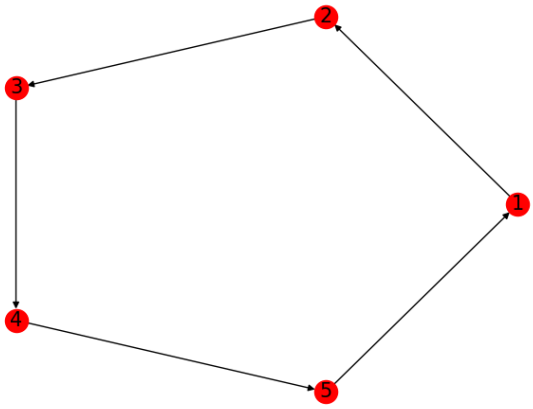


下表為此圖的 Authority、Hubness 和 PageRank 比較：

	Authority	Hubness	PageRank
節點 2	0	0	0
節點 3	0	1	0
節點 4	1	1	0.000002
節點 5	1	1	0.000048
節點 6	1	1	0.000571

由於 graph_1 從節點 1 開始指向後面的節點，而沒有往回指，所以可以預期流向最後都會集中在最後面的幾個節點，而從以上結果也應證了這個預期。
而 graph_1 經過 SimRank 計算後的結果為：所有節點之間的相似度皆為 0。

Graph_2：

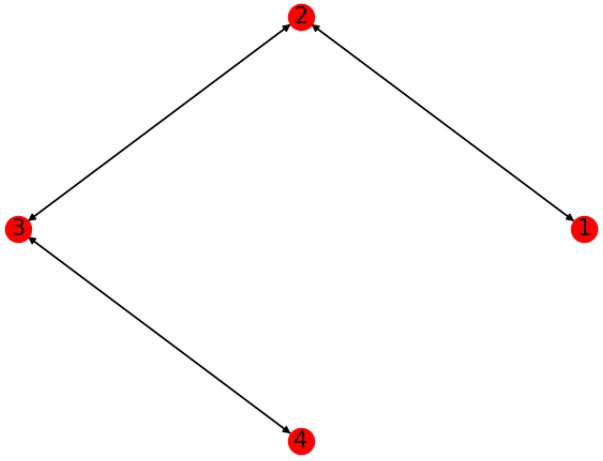


相對於 graph_1，graph_2 是一個 cycle，也就是流向最後並不會停留在某一個特定的節點，而是會平均的分散在各個節點。Authority、Hubness 和 PageRank 如下表所示：

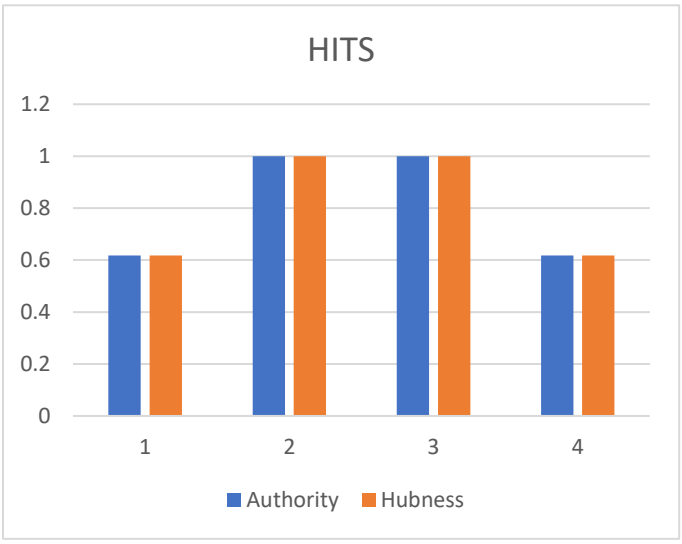
	Authority	Hubness	PageRank
節點 2	1	1	0.2
節點 3	1	1	0.2
節點 4	1	1	0.2
節點 5	1	1	0.2
節點 6	1	1	0.2

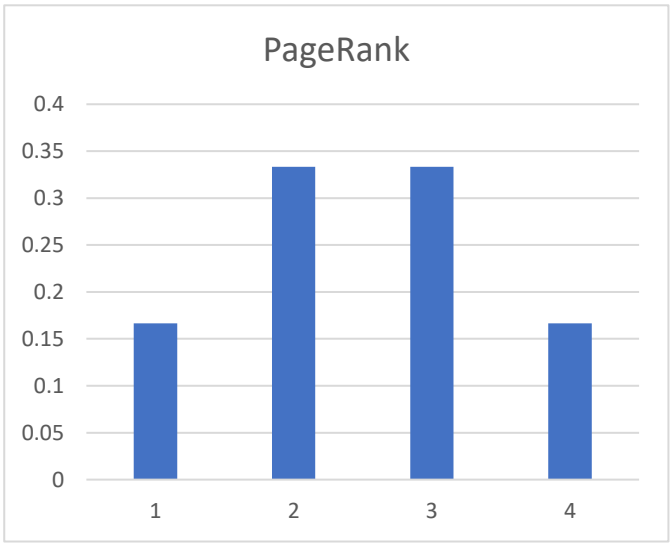
然而 SimRank 計算的結果和 graph_1 一樣，所有節點之間相似度皆為 0。

Graph_3：



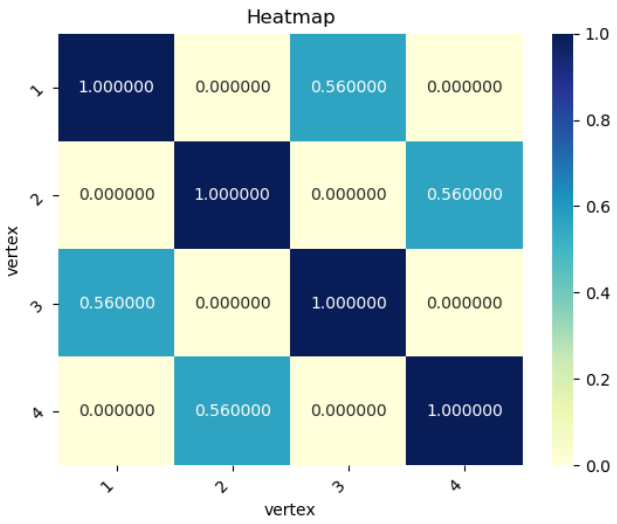
下圖左為 graph_3 經過 HITS 計算後，各個節點的 Hubness 和 Authority 直方圖，下右圖為 graph_3 經過 PageRank 的計算之後，各個節點的 PageRank 直方圖：



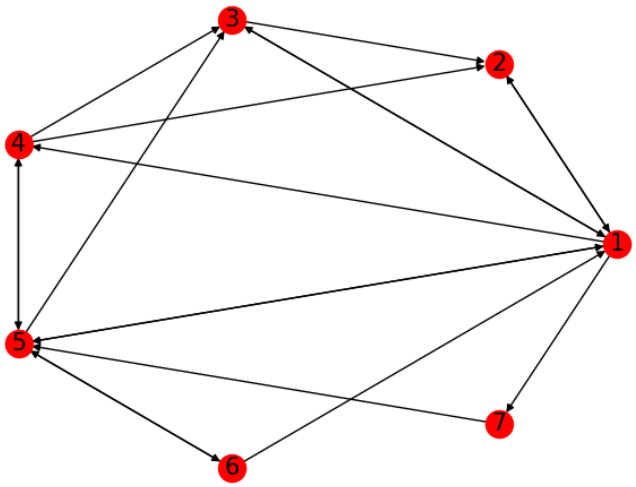


從 HITS 圖中可以看到節點 2、3 的 Authority 和 Hubness 值高於節點 1、4，5 這是因為節點 2、3 被 2 個連結指向(節點 1、4 只有一個)，所以會接收比節點 1、4 還多的 Authority 和 Hubness。而 PageRank 也是相同的情況。

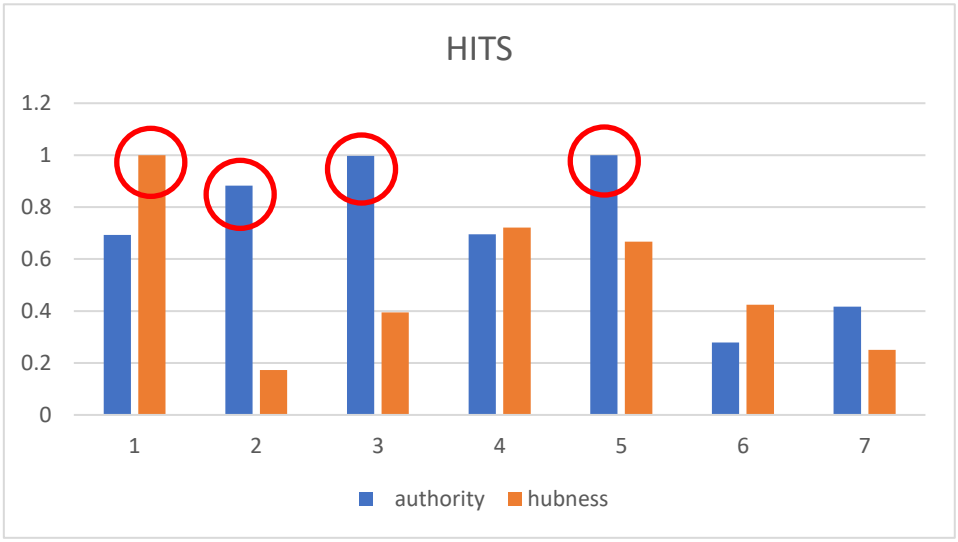
右圖為 graph_3 經過 SimRank 計算之後，各節點之間相似度的 Heat map。
從此圖顯示除了和自己的相似度為 1 之外，節點 1、3 有相似度為 0.56，以及節點 2、4 有相似度 0.56，其餘相似度皆為 0。



Graph_4：

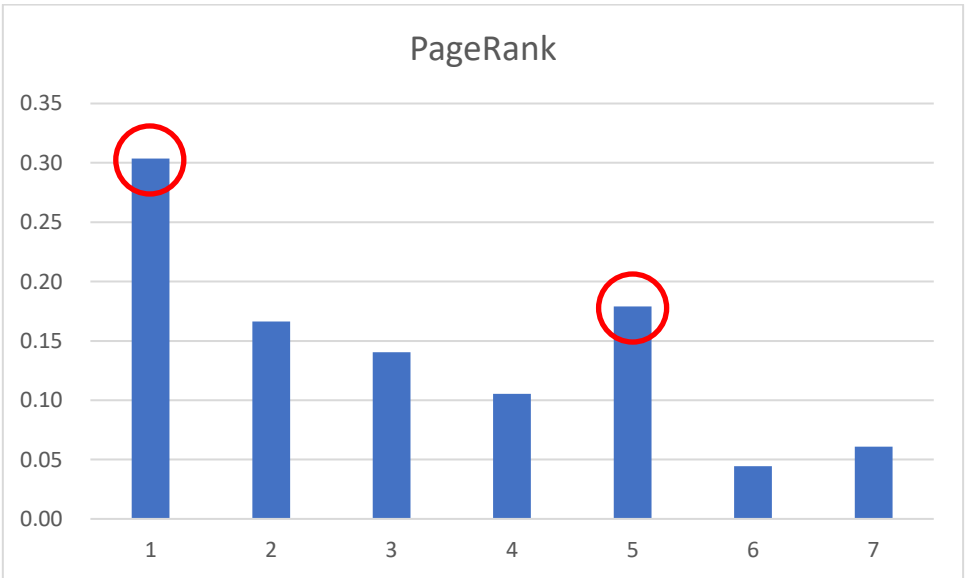


下圖為 graph_4 經過 HITS 計算之後，各個節點的 Hubness 和 Authority 直方圖：



從此圖可以看到，節點 1 的 Hubness 值較高，而因為節點 1 的連結有連到節點 2、3、5，所以連帶使得節點 2、3、5 的 Authority 值升高，此為 HITS 算法的特性。

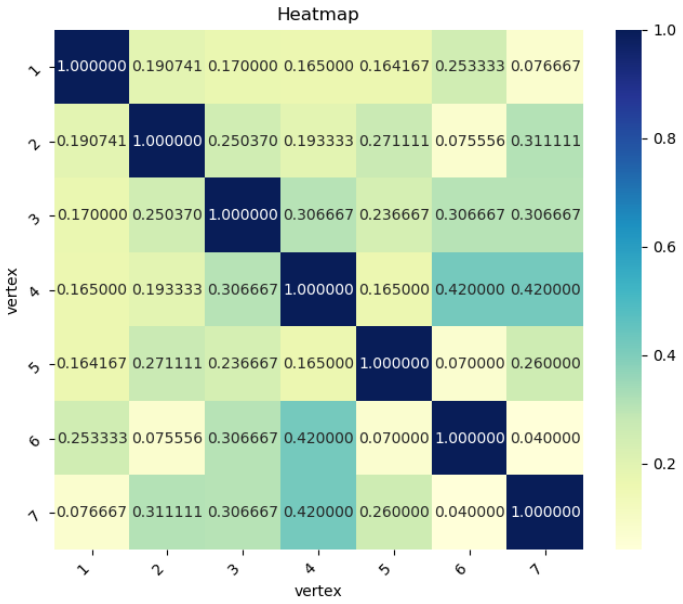
下圖為 graph_4 經過 PageRank 的計算之後，各個節點的 PageRank 直方圖：



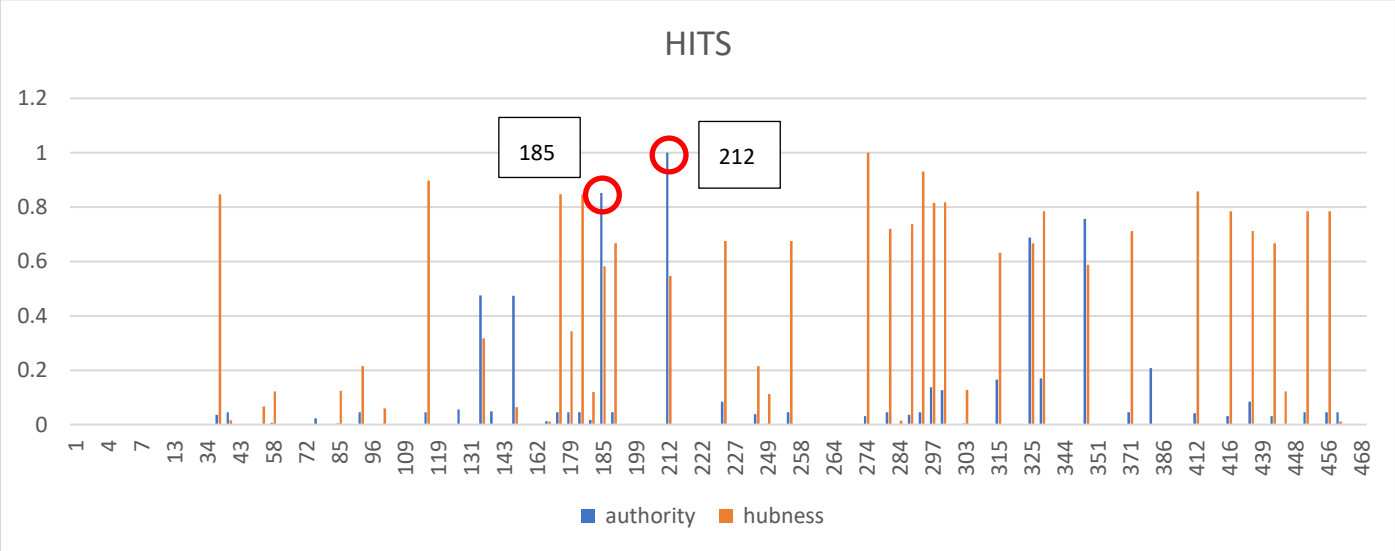
從此直方圖可以看到節點 1 和 5 分別有較高的 PageRank，這是因為有相對較多的連結連到這兩個節點(4 條)，而節點 1 又明顯高過於節點 5，猜測為節點 5 往外的連結只有 3 條，而節點 1 則有 5

條，所以節點 5 流向節點 1 的機率會大於節點 1 流向節點 5 的機率，因而造成節點 1 累積了較高的 PageRank。

右圖為 graph_4 經過 SimRank 計算之後，各節點之間相似度的 Heat map。
從此圖可以清楚看到各節點和自己的相似度值為 1，另外，節點 4 和節點 6、7 的相似度較高，其餘節點之間的 PageRank 皆在 0.3 以下，顯示節點間的相似度並不高。

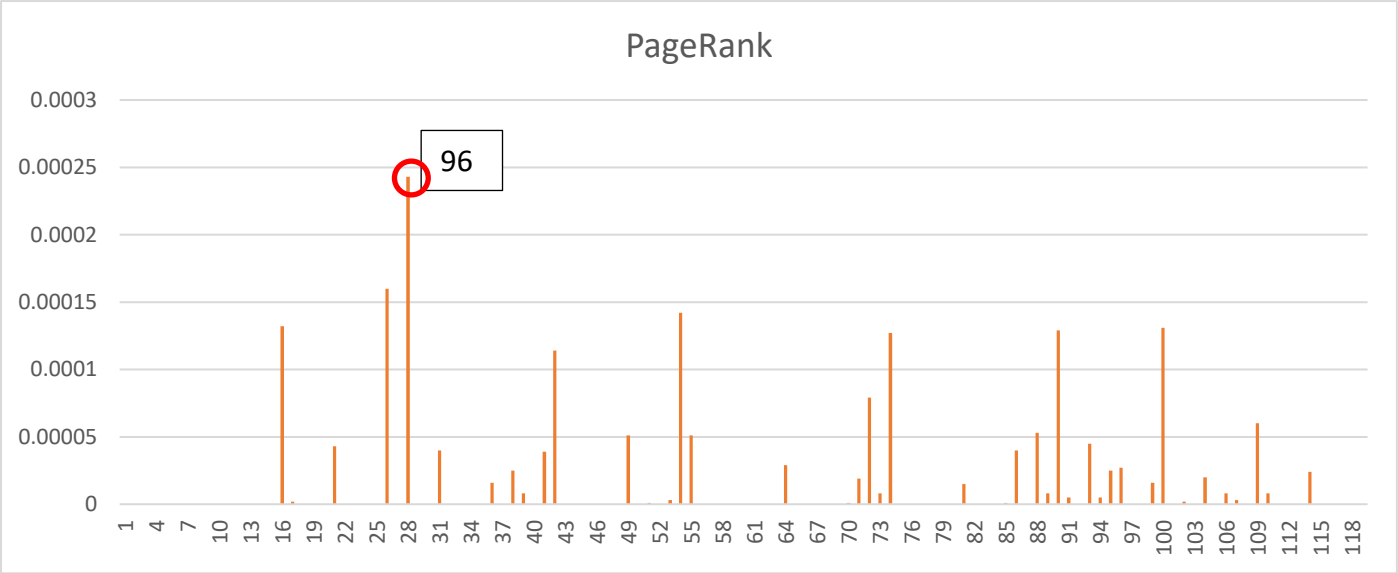


Graph_5：
下圖為 graph_5 經過 HITS 計算之後，各個節點的 Hubness 和 Authority 直方圖：



從此圖可以看到，節點 185 和節點 212 有較高的 Authority 值，這個結果由之前 graph_4 的例子是可以預期的，透過比對節點 212 的連結可以發現，幾乎所有指向他的節點皆有較大的 Hubness 值，節點 185 也是相同的情況。

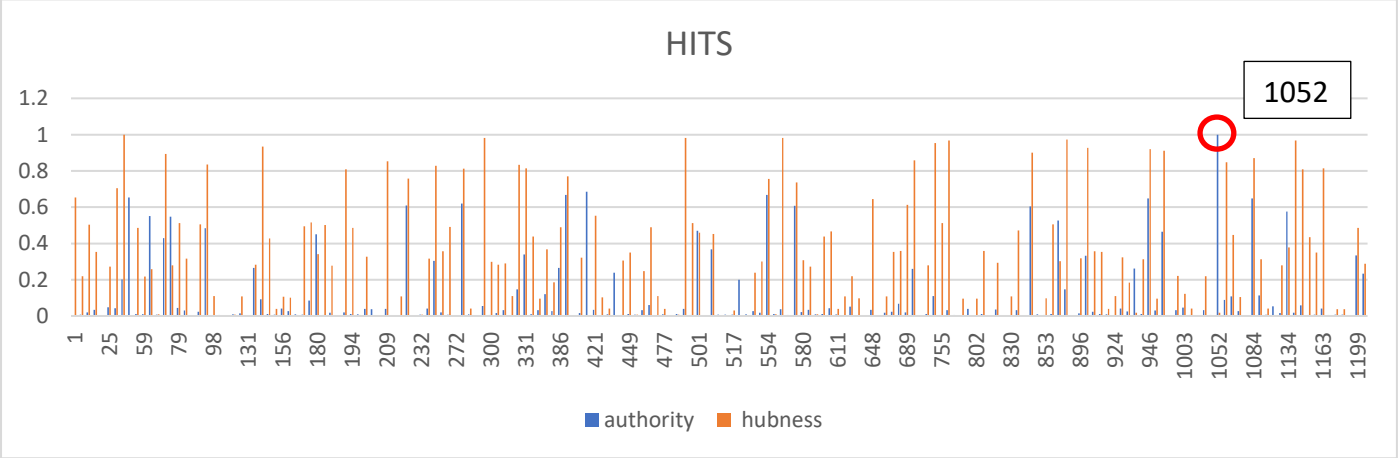
下圖為 graph_5 經過 PageRank 的計算之後，各個節點的 PageRank 直方圖：



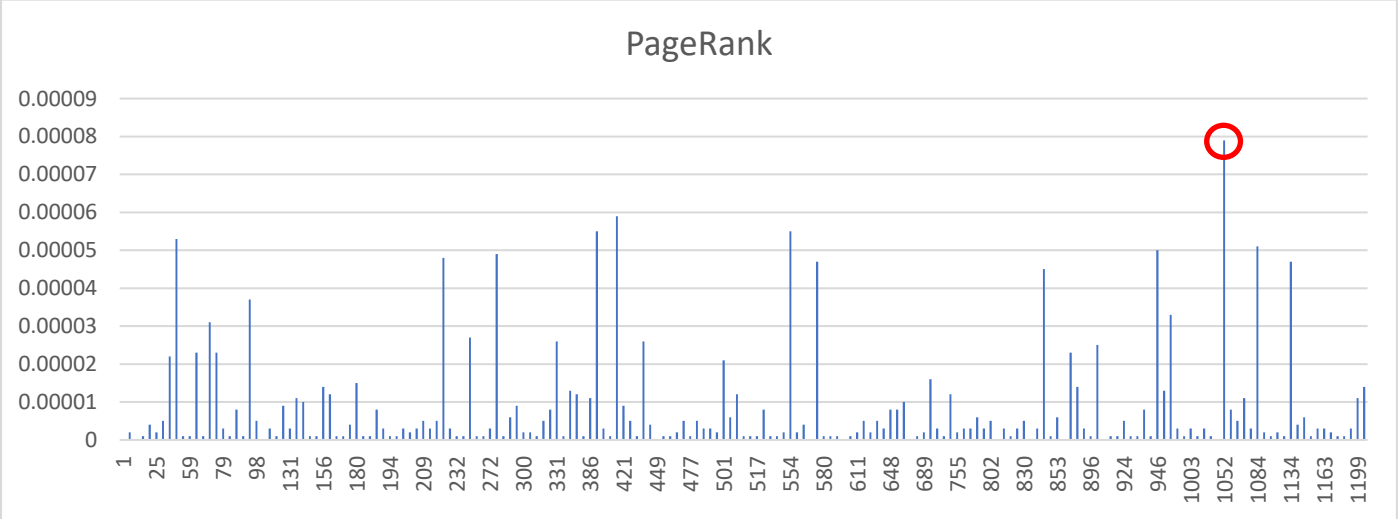
從此圖可以發現所有節點的 PageRank 都很小，而節點 96 有最高的 PageRank(0.000243)，經過對照 graph_5 可以發現指向節點 96 的連結雖然沒有到很多，但是節點 96 指出去的連結很少(3 個)，所以可能因此造成機率逐漸累積到此節點上。

Graph_6：

下圖為 graph_6 經過 HITS 計算之後，各個節點的 Hubness 和 Authority 直方圖：



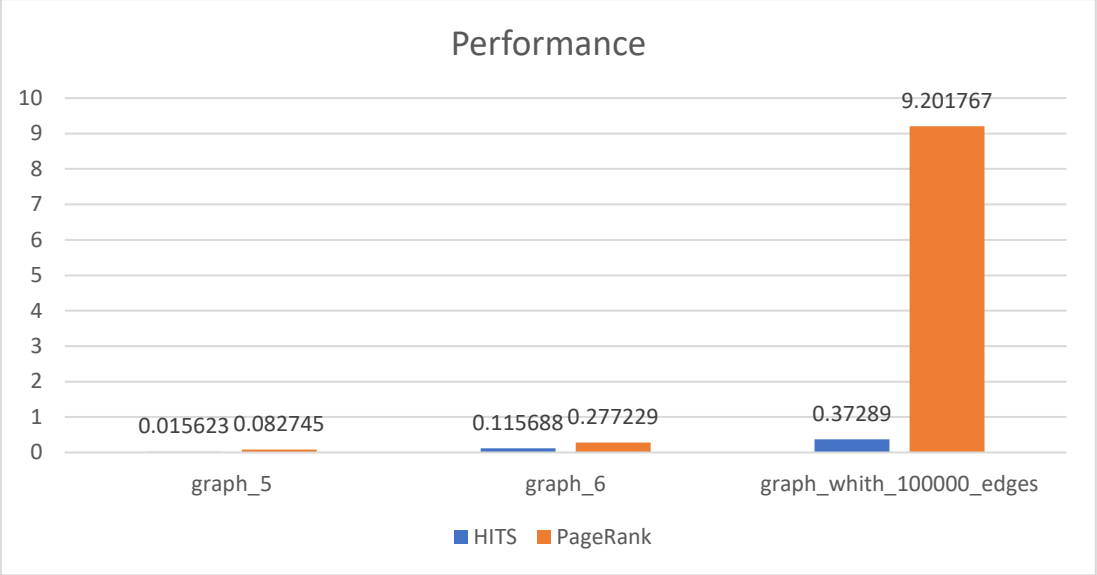
直接觀察 Authority 最高的節點 1052，可以發現和先前的結果一樣，都是被高 Hubness 所指向的緣故。



上圖為 graph_5 經過 PageRank 的計算之後，各個節點的 PageRank 直方圖，PageRank 最高值仍然是節點 1052。

IV. Performance Analysis

下圖為 HITS 和 PageRank 演算法分別計算 graph_5、graph_6 和 graph from project1 所花費的時間(單位：秒)，可以發現 HITS 在效能上要比 PageRank 快上很多，尤其是在比較大的圖上，HITS 也只用了 0.37 秒，而 PageRank 則花了 9 秒多。



另外，由於 SimRank 所計算的是每個節點間的相似度關係，計算量比較大，所以 performance 也相對比較慢。

V. Discussion

1. 如何增加 graph_1、graph_2 和 graph_3 中 Node1 的 Hubness、Authority 和 PageRank？回答：

   對於 graph_1，只要將節點 5 連回節點 1 即可(形成 cycle，Hubness、Authority 和 PageRank 會平均分散)。對於 graoh_2，只需將節點 1 連出的連接刪除即可，如此一來機率就會累積在節點 1 上面，進而提高 PageRank。對於 graph_3，可以增加一條連接節點 1、3 的雙向連結，這樣 Authority 和 Hubness 會提升到 0.854702, 0.854543，而 PageRank 會提升到 0.249949。