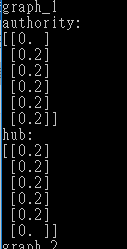
(以下均以圖1為例)

HITS:

1. 先執行read\_data() 讀取資料，得到Adjacency matrix(A)
2. 每node的authority,hub初值令為1，authority,hub分別存放每個Node的authority值 與hub值
3. Authority的新值可由A^T \*hub 更新，hub值可由 A\*Authority 更新
4. 計算完authority\_new,hub\_new 後，計算authority,hub與authority\_new,hub\_new差 的平方合取根號誤差是否大於0.000001

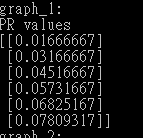
若有則更新authority,hub。直到跑完10000次或是誤差小於0.000001

1. authority,hub即為所求



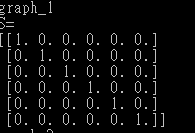
PageRank:

1. 先執行read\_data() 讀取資料，得到Adjacency matrix(A)
2. 每node的pagerank初值令為1/Node的數量，PR 存放每個node的pagerank值，d令為0.1
3. PK的新值可由A^T \* PR\*(1-d)+(1/ Node的數量)\*d 更新，並計算PK舊值與新值的差的平方取根號，是否大於0.000001，若有則跟新PK值。直到跑完10000次或是誤差小於0.000001
4. PK即為所求



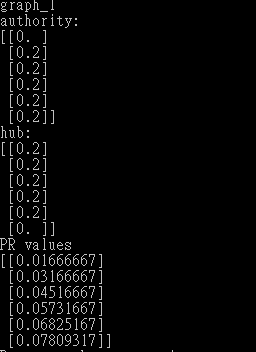
SimRank:

1. 先執行read\_data() 讀取資料，得到Adjacency matrix(A)
2. 將A修改成 每行合為1的矩陣Q，C令為0.8，S令為與A大小相同的單位矩陣
3. S新值可由Q^T\*S\*Q\*C更新，並計算S與S的新值 中每個元素是否誤差大於0.000001，若有則更新S。直到跑完1000次或是誤差小於0.000001
4. S即為所求

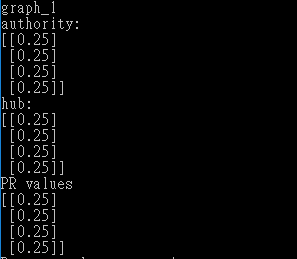


Find a way (e.g., add/delete some links) to increase hub, authority, and PageRank of Node 1 in first 3 graphs respectively.

Grapg\_1: 去掉4,5 5,6 增加4,1

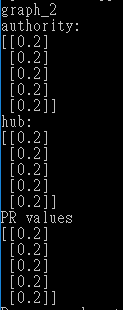


變成

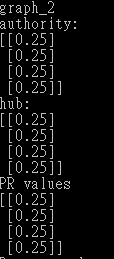


Node 1的hub, authority, and PageRank 均有增加

Graph\_2:刪去4,5 5,1 增加4,1

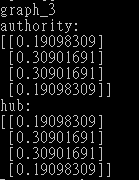
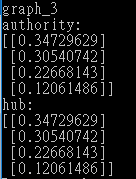


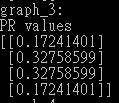
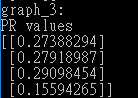
變成



Node 1的hub, authority, and PageRank 均有增加

Graph\_3:增加1,1

變成

Node 1的hub, authority, and PageRank 均有增加