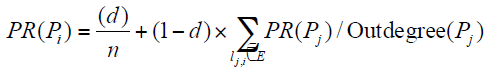
Project3\_report

* Implementation detail
  + HITS：

1. 讀取csv的資料
2. 初始一個node數\*node數的矩陣，值為零
3. 把兩node間有邊的再矩陣相對位置標為1，此為aut矩陣
4. 把aut矩陣轉置，此為hub矩陣
5. 初始每個node的aut值，都為1
6. 寫一個recursive function，把node的aut乘以aut matrix即為hub，再把hub乘以hub matrix即為aut🡺值到來回一次的值不變

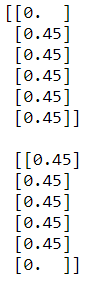
* PageRank：

1. 讀取csv的資料
2. 初始一個node數\*node數的矩陣，值為零
3. 把兩node間有邊的再矩陣相對位置標為1，此為out矩陣
4. 把out矩陣每一列除以那列的加總，再轉置，此為parent矩陣
5. Matrix🡺
6. 初始一開始每個node的rank🡪1除以node數
7. 寫一個recursive function，node rank乘以matrix後跟上個node rank如果對大相差的值小於threshold則停止

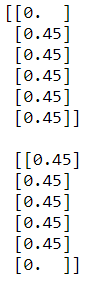
* SimRank：

1. 讀取csv的資料
2. 初始一個node數\*node數的矩陣，值為零
3. 把兩node間有邊的再矩陣相對位置標為1，此為simrank矩陣
4. 把simrank矩陣轉置，此為parent矩陣
5. 創建一個單位矩陣query\_sim，此為要記錄output相似值的矩陣
6. 寫一個function simrank，可以設定要跑幾次(time)，每次會搜尋任兩個node的parents然後找到是否能更新query\_sim

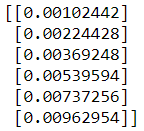
* Increase hub, authority,and PageRank of Node 1：
  + Graph\_1：
    - 增加(6,1)、(4,1)
      * Hub：0🡺0..01
      * Aut：0.45🡺0.85
      * PageRank：0.001🡺0.199
  + Graph\_2：
    - 增加(3,1)、(1,4)
      * Hub：0.45🡺0..59
      * Aut：0.45🡺0.59
      * PageRank：0.2🡺0.275
  + Graph\_3：
    - 增加(3,1)、(1,4)
      * Hub：0.37🡺0..55
      * Aut：0.37🡺0.49
      * PageRank：0.175🡺0.225
* Result analysis and discussion：
* Graph\_1：
  + Hits
    - Hub：因為是從1開始連到6，除了1每個node都有1個node連到它，所以除了node 1 是零其他都是0.45



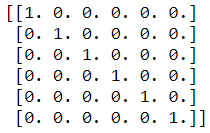
* + - Aut ：相反的，只有node 6沒有連出去的node，所以是零



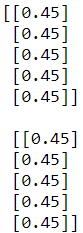
* + PageRank：因為是一條線的連下去，所以水流越加越大



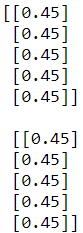
* + SimRank：他們沒有共同指向的node，所以彼此相似度為0



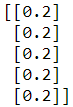
* Graph\_2：
  + Hits
    - Hub：因為是一個環狀，每個都有1個node連它



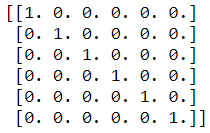
* + - Aut ：因為是一個環狀，每個都有連出去1個node



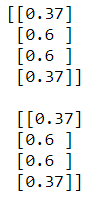
* + PageRank：因為是一個環狀，所以大家平分了值



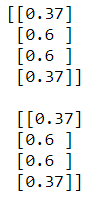
* + SimRank：他們沒有共同指向的node，所以彼此相似度為0



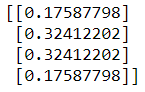
* Graph\_3：
  + Hits
    - Hub：因為有兩個node連到2.3，相較1.4會有更大的hub



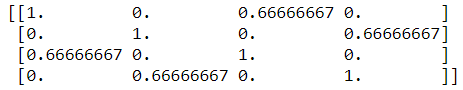
* + - Aut ：因為2.3連出去兩個node，相較1.4會有更大的aut



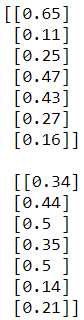
* + PageRank：因為2.3是連結最多的node，又2.3的連結數、1.4的連結數一樣，所以2=3>1=4



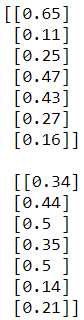
* + SimRank：1.3有共同的parent (2)，所以有相似度。  
    2.4共同parent(3)



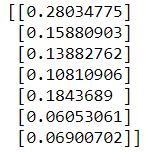
* Graph\_4：
  + Hits
    - Hub：沒有規律性，主要看多少Node連到自己



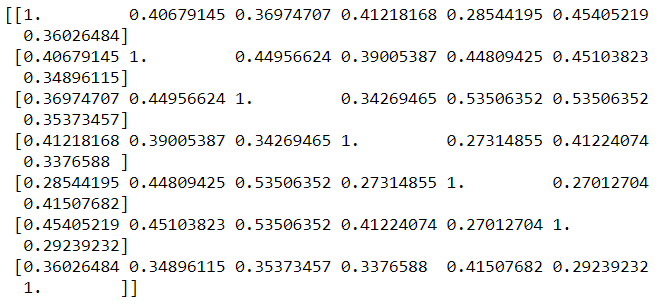
* + - Aut ：沒有規律性，主要看自己連出去多少nodes



* + PageRank：



* + SimRank：



* Graph\_5：
  + Hits
    - Hub：
      * graph\_5\_hub.txt
* Aut ：
  + graph\_5\_ aut.txt
  + PageRank：
    - * graph\_5\_pagerank.txt
  + SimRank：
    - * graph\_6\_SimRank.csv
* Graph\_6：
  + Hits
    - Hub：
      * graph\_6\_hub.csv
* Aut ：
  + graph\_6\_ aut.csv
  + PageRank：
    - * graph\_6\_pagerank.csv
* Discussion ：

我覺得這次的作業也蠻有趣的，以前使用google等瀏覽器都不知道網頁是怎麼排名的。自己寫過後才真正了解每個algorithm的算法。寫完這次的作業也讓我對python的矩陣乘法、轉置更加熟練了