INFORMATION RETRIEVAL(CSE 508)

GROUP MEMBER 1: SHASHANK RUSTAGI(MT21081)

GROUP MEMBER 2: DRISHYA UNIYAL(MT21119)

NOTE: CODE IS PRESENT IN THE ATTACHED IPYNB FILE

LIBRARIES WHICH WE USED IN OUR FILE

```
import numpy as np #NUMERICAL PYTHON
import pandas as pd #FOR DATAFRAMES
import math, operator, warnings #SOME IMPORTANT SYSTEM LIBRARIES
import matplotlib.pyplot as plt #FOR PLOTTING
import networkx as nx  #FOR REPRESENTATION OF A GRAPH
from prettytable import PrettyTable #FOR THE CREATION OF THE RELATIONAL TABLES
from tqdm import tqdm #FOR CREATING THE PROGRESS BARS IN THE CODE
warnings.filterwarnings('ignore')
filename = 'C:/Users/HP/Downloads/soc-sign-bitcoinalpha.csv'
```

ANS 1A

DATASET: BITCOIN ALPHA
Type of data: weighted, signed, directed and temporal
Number of nodes in the dataset → 3783
Number of Edges in the dataset → 24186

PARAMETER	VALUE
TOTAL NUMBER OF EDGES IN THE NETWORK	24186
TOTAL NUMBER OF NODES IN THE NETWORK	3783
NODE WITH MAXIMUM IN-DEGREE IN THE NETWORK	1
NODE WITH MAXIMUM OUT-DEGREE IN THE NETWORK	1
AVERAGE IN-DEGREE IN THE NETWORK	6.39 (rounded off to two decimal places)
AVERAGE OUT-DEGREE IN THE NETWORK	6.39 (rounded off to two decimal places)
DENSITY OF THE NETWORK	0.002 (rounded off to three decimal places)

Network Representation in the form of Adjacency Matrix

n*n adjacency matrix:
Set row and column in the matrix for each of the n pages in the base.
Entry[i][j] is 1 if page i links to page j, else = 0

ADJACENCY MATRIX

	1	2	3	4	5	6	7	8	9	10	...	7595	7596	7597	7598	7599	7600	7601	7602	7603	7604
1	0	1	0	1	0	0	0	0	1	1	...	0	0	1	0	0	0	0	0	1	0
2	1	0	0	1	1	0	1	1	1	1	...	0	0	0	0	0	0	0	0	1	0
3	0	1	0	0	1	1	1	1	0	1	...	1	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	1	1	...	0	1	0	0	0	0	0	0	1	0
5	0	1	1	0	0	1	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7600	0	0	0	0	0	0	1	0	1	0	...	0	0	0	1	1	0	1	1	0	1
7601	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	1	0	1
7602	0	0	0	0	0	0	1	0	1	0	...	0	0	0	0	0	0	0	0	0	1
7603	1	1	0	0	0	0	0	1	0	0	...	0	0	0	1	0	0	0	0	0	1
7604	0	0	1	0	0	1	1	0	0	0	...	1	0	0	1	0	0	1	1	0	0

.

ANS 1B

Dataset Chosen: Bitcoin ALPHA

We have chosen a network dataset named BITCOIN ALPHA WEB OF TRUST NETWORK. This network describes trust between two bitcoin users. If an edge exists from node A to node B, then A trusts B. The dataset comprises the source node, destination node, edge weight, and time stamp columns.

- The average indegree and out-degree remain the same because every outdegree constituted by a single node will be counterbalanced by in degrees of the nodes having incoming edges.
- The density of the network tells us how dense the network is; if the density of the network is 1, then the graph is a complete graph. If it is a 0, then the network has no edges. The density of the network = total number of edges in the network/(n)(n+1), where n is the total count of nodes in the network(in the case of a directed graph)
- In the edge list, if we represent all the sets of edges present in the network in a list, then it is called edge list representation. Fraction of the output image of how the edge list looks can be seen in the below figure.

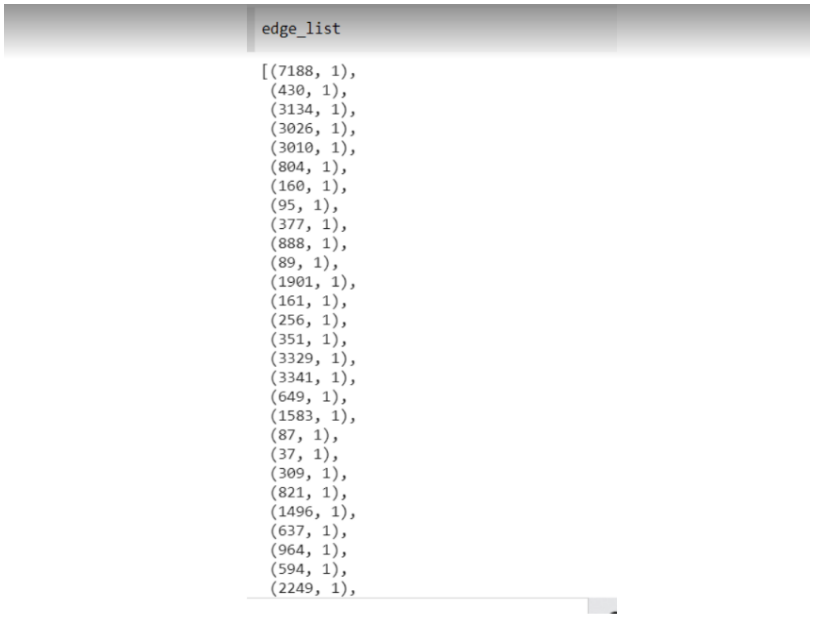
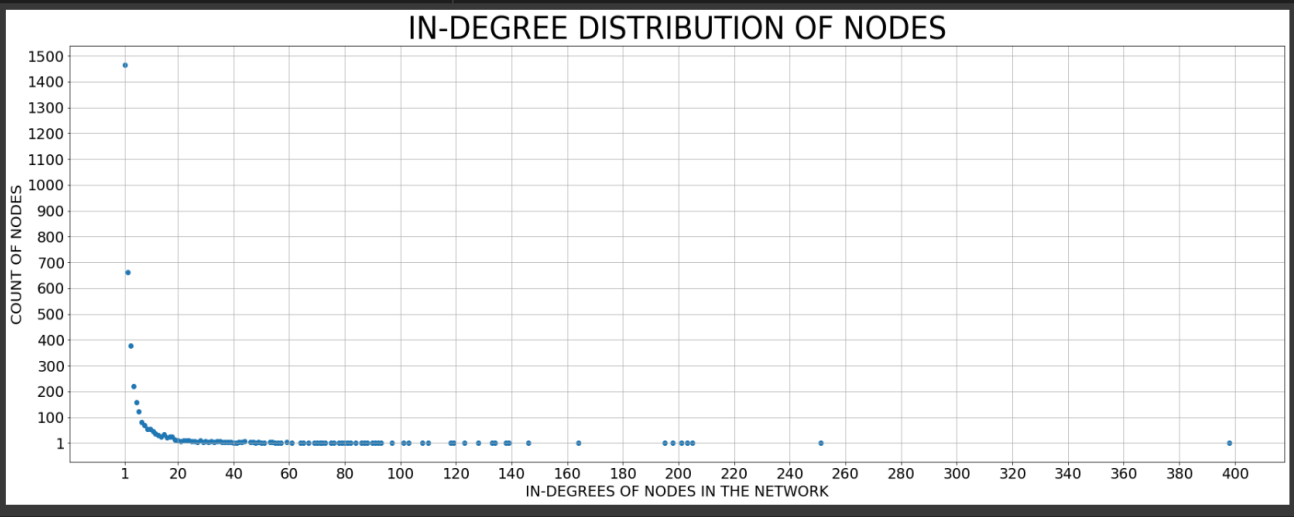


Fig 3. Edge List representation of network

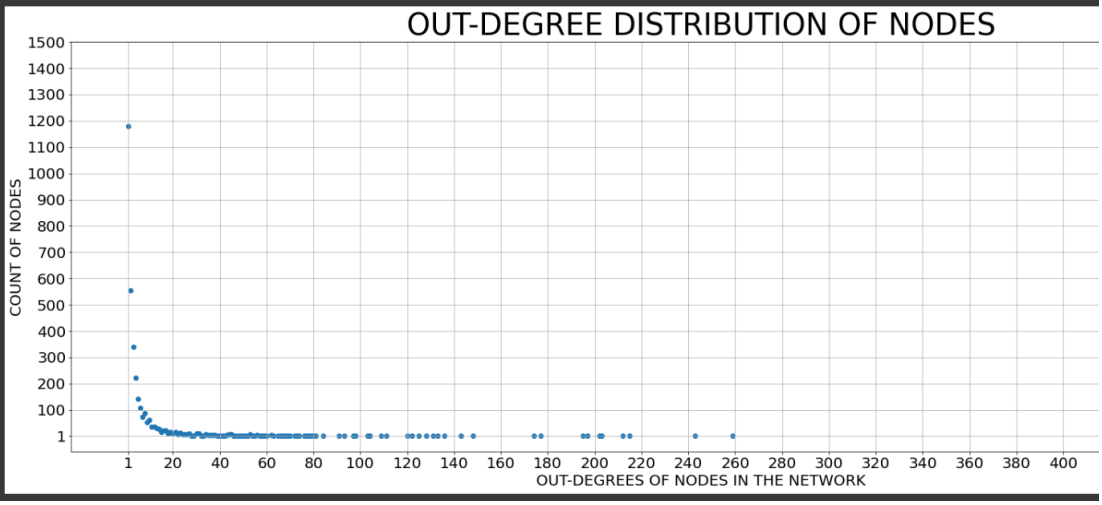
```
indegrees_vs_nodecount = degree_cal(data['target'], 2)

showGraph(indegrees_vs_nodecount, calAxisVals(401, 20), calAxisVals(1501, 100), "IN-DEGREES OF NODES IN THE NETWORK", "IN-DEGREE DISTRIBUTION OF NODES")
```




```
outdegrees_vs_nodecount = degree_cal(data['source'],2)

showGraph(outdegrees_vs_nodecount, calAxisVals(401, 20), calAxisVals(1501, 100), "OUT-DEGREES OF NODES IN THE NETWORK", "OUT-DEGREE DISTRIBUTION OF NODES")
```



ANS 1C

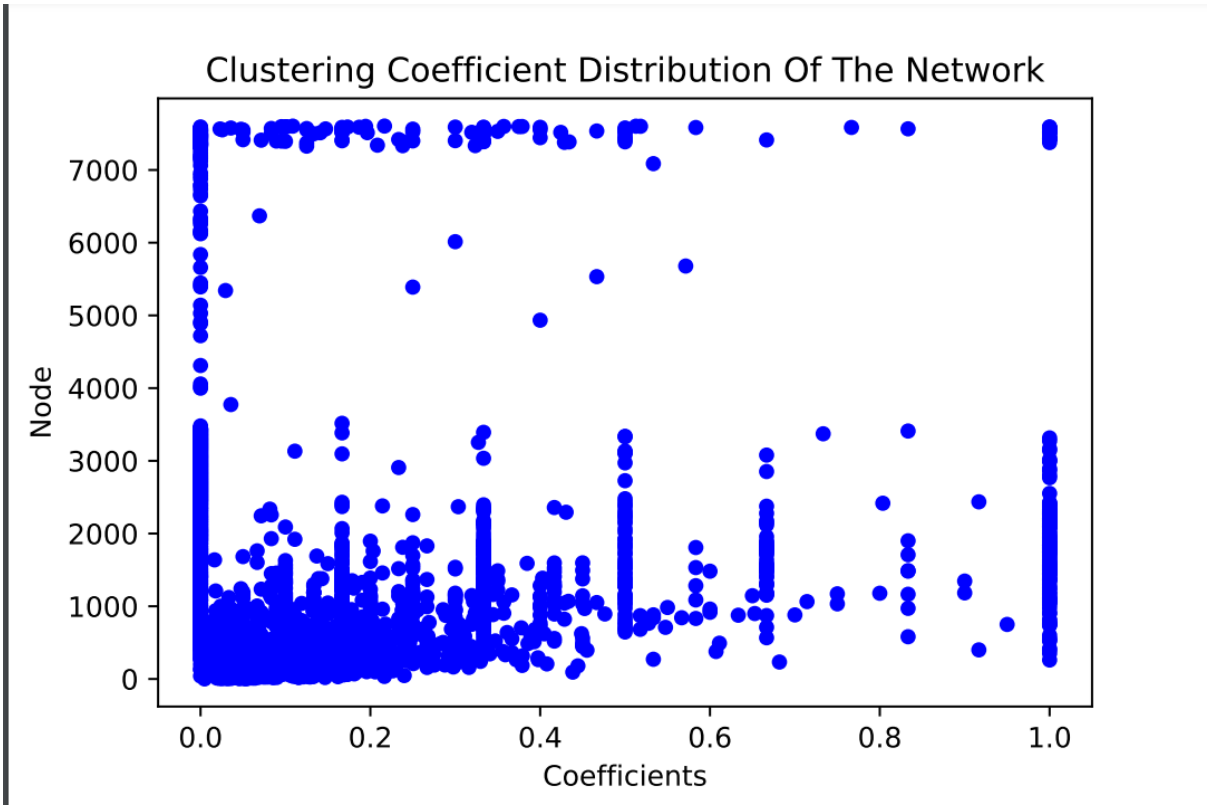
1. A clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together.
2. Calculate neighbors, possible pairs that can be formed using the nodes.
3. Calculate the number of actual pairs adjacent to each other.
4. Divide the both for local clustering coefficient.

Results:

NODE	CLUSTERING-COEFFICIENT
261	1.0
352	1.0
399	1.0
418	1.0
524	1.0
538	1.0
556	1.0
600	1.0
736	1.0
746	1.0

```
NODES WITH CLUSTERING COEFFICIENT 0:: 2317
NODES WITH CLUSTERING COEFFICIENT 1:: 173
CLUSTERING COEFFICIENT OF THE NETWORK:: 0.12635262321014096
```

Clustering Coefficient and Number of Nodes Graph



ANS 2A

PAGE RANK

1. Use the library networkx for this calculating PageRank.
2. Get the page ranks in sorted order.

Page rank ranks the pages in the order of relevance. More incoming edges implies higher page rank.


```
{7188: 4.9739967085536536e-05,
 1: 0.016993099228405292,
 430: 0.00028751507169557397,
 3134: 0.00011917626603597539,
 3026: 7.937879011849084e-05,
 3010: 7.937879011849084e-05,
 804: 0.00034797608896262324,
 160: 0.0004624121679665046,
 95: 0.002948042055839976,
 377: 0.0003107517366525091,
 888: 0.00018761408622671417,
 89: 0.0018127896997493635,
 1901: 7.937879011849084e-05,
 161: 0.001282547951816451,
 256: 0.0010931452350225065,
 351: 0.0004277054038172364,
 3329: 7.937879011849084e-05,
 3341: 7.937879011849084e-05,
 649: 0.0004168676578369132,
```

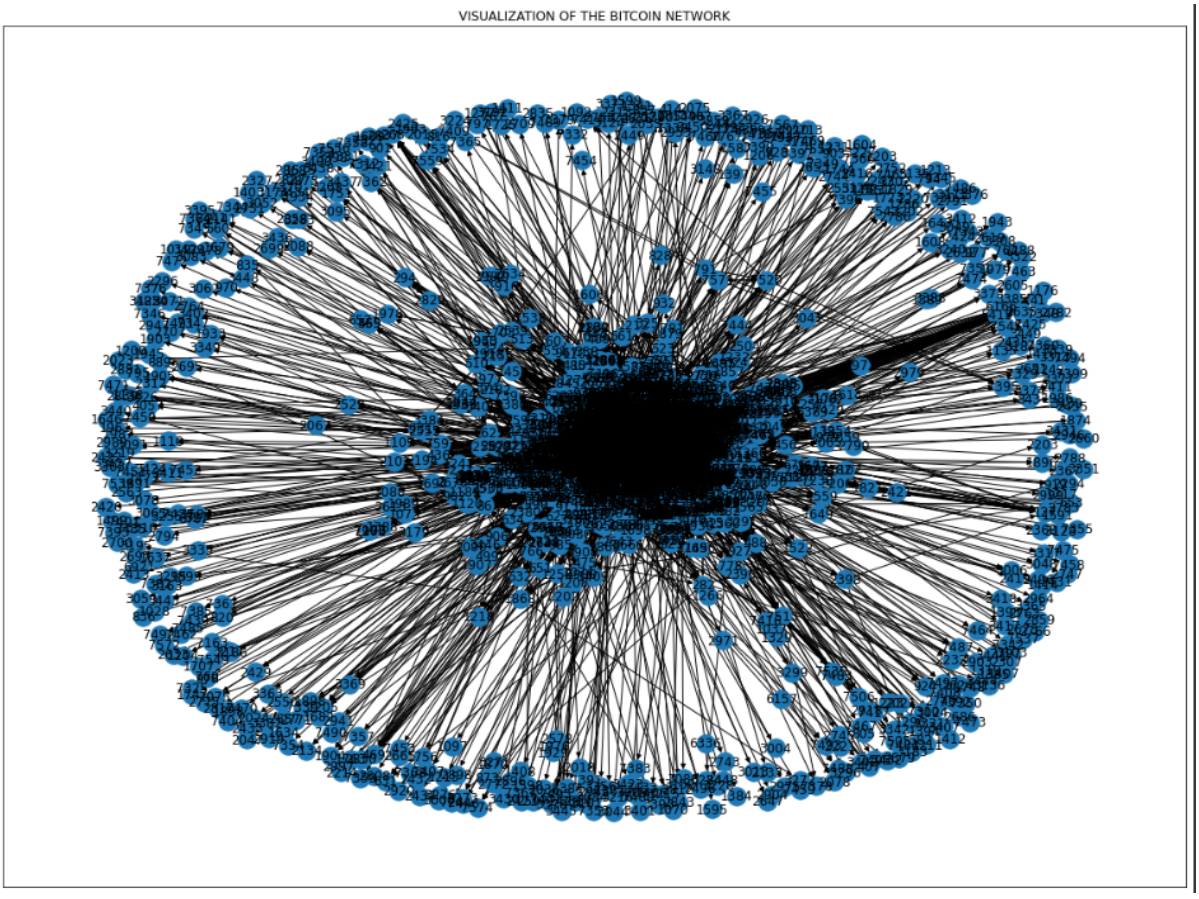

Authority and Hub score for each node.

1. Let the number of iterations be k.
2. Hub score = 1 and authority score =1 are assigned to each node.
3. Repeat k times.
4. Update the scores.
5. Normalise the scores by dividing hub scores by the square root of the sum of the squares of all hub scores and, similarly, authority scores.

```
Hub Scores:  {7188: 0.01773641450887142, 1: 0.7498940032246939, 430: 0.042126293773090726, 3134: 0.041142266477210074, 3026:
0.01773641450887142, 3010: 0.01773641450887142, 804: 0.06419318429621863, 160: 0.03198207321517327, 95: 0.6371391725770787, 3
77: 0.01828263814485761, 888: 0.025072789889361253, 89: 0.17132813954857867, 1901: 0.01773641450887142, 161: 0.17745019275678
883, 256: 0.12300443477746259, 351: 0.10387437614561869, 3329: 0.01773641450887142, 3341: 0.01773641450887142, 649: 0.0837628
6562446185, 1583: 0.01773641450887142, 87: 0.27877048927529646, 37: 0.2679767466692828, 309: 0.1290052434400096, 821: 0.02600
782993014586 3, 1496: 0.020615308198089044, 637: 0.055371359637170096, 964: 0.032695884400922214, 594: 0.02848785360005503, 22
49: 0.01773641450887142, 554: 0.032803802189718 67, 20: 0.3474900838988803, 2227: 0.01773641450887142, 1315: 0.01795518710171
7587, 519: 0.07499589349358508, 1316: 0.01773641450887142, 2149: 0.025309414400931148, 1724: 0.02733734040506788, 18: 0.17023
942048179824, 57: 0.11036080608895127, 118: 0.22842246071092756, 3254: 0.09248775542646258, 1177: 0.01852503728375827, 112:
0.16973668337234435, 11: 1.0, 586: 0.043363695321042 54, 35: 0.1208536974369069, 15: 0.5709340428814462, 1445: 0.0300362304499
14453, 152: 0.13613487609680733, 2: 0.80053681376352, 113: 0.25340180024545306, 44: 0.22148110752277103, 2401: 0.029144911961
85645, 10: 0.7059029101361409, 2378: 0.026886688680864072, 126: 0.11734037645269654, 3245: 0.01773641450887142, 783: 0.034690
753584208416, 493: 0.06304756166190292, 1358: 0.038584800090737316, 1180: 0.072311440325725, 529: 0.02442577633200952, 333:
0.13381885531936122, 1538: 0.03783832270475893, 2282: 0.01773641450887142, 1519: 0.024660759406486522, 2966: 0.01773641450887
142, 474: 0.03974955874418089, 330: 0.125753523508884, 958: 0.05123379473933577, 17: 0.5223467674665369, 1295: 0.043850772961
0968, 38: 0.2534871263970796, 1952: 0.01773641450887142, 223: 0.09083744643568245, 625: 0.02784308690340712, 1392: 0.03131762
7418681805, 3355: 0.01773641450887142, 1881: 0.018199564526444043, 58: 0.6403129569660658, 96: 0.24297376668918338, 1580: 0.0
3589297189743019, 196: 0.13854411490903423, 146: 0.1979422634208842, 416: 0.05557340003116229, 1198: 0.03894087006092552, 331
9: 0.01773641450887142, 1867: 0.02526218296161409, 896: 0.030921604151264114, 617: 0.04717003891152235, 3300: 0.0177364145088
```

In [28]:
```
1 print("Authority Scores: ", authorities)
```

```
Authority Scores:  {7188: 0.0, 1: 0.7629812867700283, 430: 0.015547634604266813, 3134: 0.05043716170504289, 3026: 0.017925912
803921522, 3010: 0.017925912803921522, 804: 0.06502543707903086, 160: 0.03090544571870701, 95: 0.6025928593091977, 377: 0.018
1016679413288, 888: 0.033817638610082676, 89: 0.18594537889323326, 1901: 0.017925912803921522, 161: 0.14535578717919853, 256:
0.14547566331683087, 351: 0.09479333871559069, 3329: 0.017925912803921522, 3341: 0.017925912803921522, 649: 0.102654541192792
27, 1583: 0.017925912803921522, 87: 0.27660316997921464, 37: 0.2341862658295437, 309: 0.14596054783482817, 821: 0.02564801191
952796, 1496: 0.020946342945116863, 637: 0.06584733175897735, 964: 0.05690146343822338, 594: 0.029284062181457737, 2249: 0.01
7925912803921522, 554: 0.037885340687396345, 20: 0.33725678908338563, 2227: 0.017925912803921522, 1315: 0.018115996942283837,
519: 0.09395234449294437, 1316: 0.017925912803921522, 2149: 0.018943277252462865, 1724: 0.026258196985474814, 18: 0.168275748
17940252, 57: 0.1066162000241972, 118: 0.25309262183944986, 3254: 0.08973091051202335, 1177: 0.018110430936136835, 112: 0.149
5442344501994, 11: 1.0, 586: 0.05691132636382892, 35: 0.28654195994446374, 15: 0.50171367686528, 1445: 0.031152966263314867,
152: 0.10763005180865486, 2: 0.8802049287910431, 113: 0.29662837040254914, 44: 0.23354786262056898, 2401: 0.0298541358225275
5, 10: 0.6960401525907898, 2378: 0.02664305213981905, 126: 0.0974379926804272, 3245: 0.017925912803921522, 783: 0.03502583052
746775, 493: 0.06557692159249434, 1358: 0.038116409866249354, 1180: 0.057752181885489286, 529: 0.025578487319147993, 333: 0.1
2628146619641528, 1538: 0.03895084084935476, 2282: 0.017925912803921522, 1519: 0.025988193424654284, 2966: 0.0179259128039215
22, 474: 0.03206369143826861, 330: 0.1238436331297235, 958: 0.055547489714389706, 17: 0.5078592940190219, 1295: 0.04960429574
769319, 38: 0.2057857693826845, 1952: 0.017925912803921522, 223: 0.08557800344741906, 625: 0.026687219698181698, 1392: 0.0321
0001144191892, 3355: 0.017925912803921522, 1881: 0.018401126159848288, 58: 0.291212230259513, 96: 0.21532142489532974, 1580:
0.03553453272678536, 196: 0.16569256376303795, 146: 0.19509089002598, 416: 0.07966279164936055, 1198: 0.03963467267058953, 33
19: 0.017925912803921522, 1867: 0.03403517583867399, 896: 0.041497186520381465, 617: 0.06654315059191059, 3300: 0.01792591280
```



Comparison between Algorithm1 and Algorithm 2

The HITS algorithm is way slower and eats up a lot of time compared to the Pagerank Algorithm.
Pagerank Algorithm produces more relevant results than the HITS algorithm due to mutual reinforcement between the hub scores and authority.