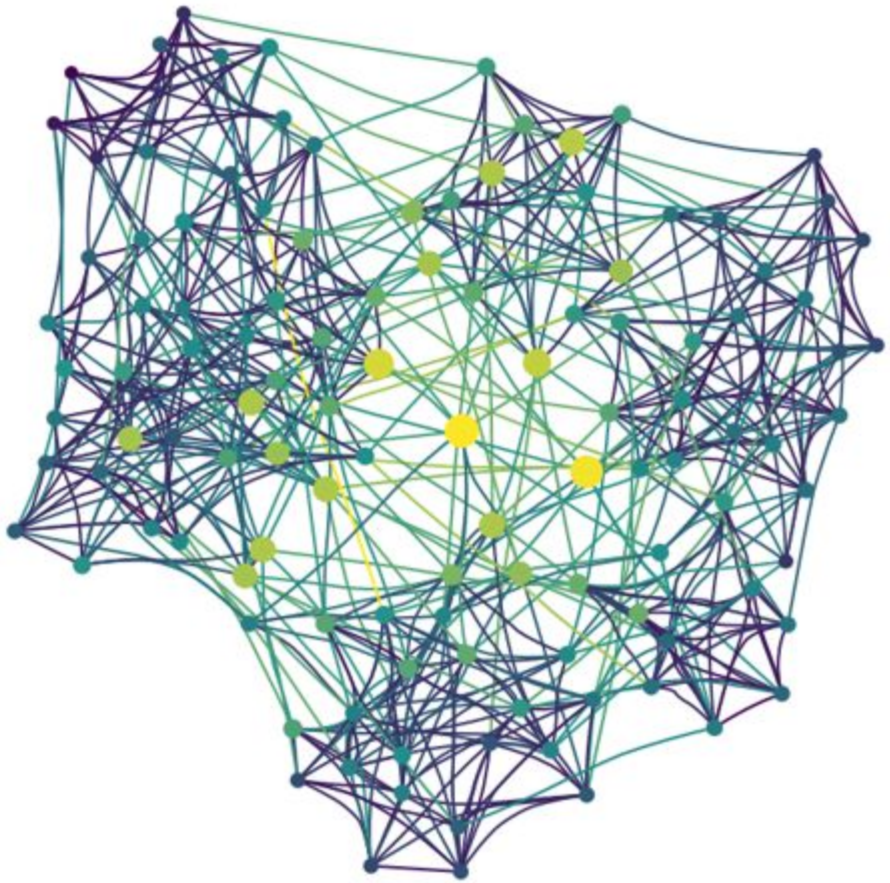
# SPECTRAL CENTRALITY MEASURES IN TEMPORAL NETWORKS

Novi Sad, 2020

Jovana Sofinkić
Hunor Tot-Bagi

# Agenda

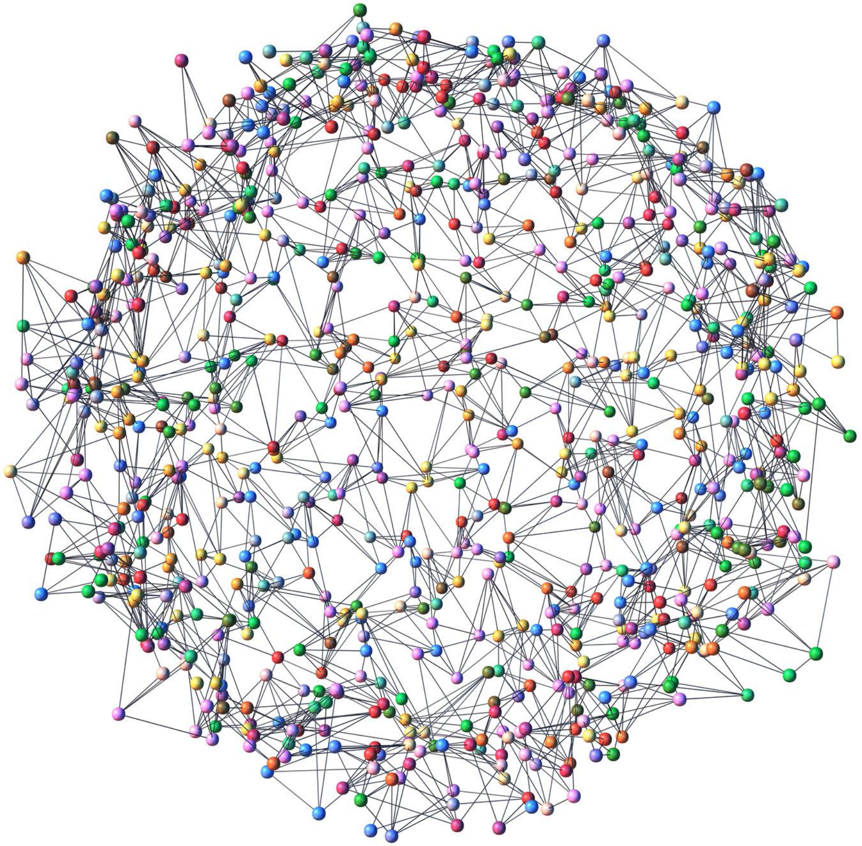
- Motivation
- Algorithms
- Results



# MOTIVATION

- Determine the important nodes in the network
- The usual questions:
  - Who are the influential people in a social network?
  - Which roads are most often used?
  - Which web pages are important
- To give answer to these questions we use several algorithms

# ALGORITMS



- Eigenvalue centrality
- Katz centrality
- Bonacich $\alpha$-centrality
- Bonacich $(\alpha, \beta)$-centrality
- Hubs and authorities
- PageRank

# EIGENVECTOR CENTRALITY

- Degree centrality – are the nodes with highest degree the most important?
- "It is better to have less friends who are powerful than to have a lot of non-powerful friends."
- $x_v = a_{1v}x_1 + a_{2v}x_2 + \cdots + a_{nv}x_n = \sum_{u:u \to v} a_{uv}x_u$ (prestige measure)
- $A^T x = \lambda x$ – in centrality
- $Ax = \lambda x$ – out centrality
- Power Iteration Algorithm – approximates the dominant eigenvalue and the corresponding eigenvector

# KATZ CENTRALITY

- Taking into account the centralities of all the nodes from which the observed node is reachable
- $T = \alpha A + \alpha^2 A^2 + \cdots + \alpha^k A^k + \cdots = (I - \alpha A)^{-1} - I$
- $\left(\frac{1}{\alpha} I - A^T\right) t = d$ (equivalent form)
- Weight $\alpha$ is used to dampen the effects of more distant nodes, $\alpha \in [0,1]$
- $A^r$ - number of walks of length $r$
- $d$ - vector of indegrees

# JACOBI ITERATION

- Iterative method for solving linear systems of the form $\boldsymbol{A}x = b$
- Idea is to decompose the matrix A in the following way: $A = L + D + U$
- $D = diag(A)$
- $L$ - lower triangle of $A$
- $U$ - upper triangle of $A$
- $Dx_{m+1} = -(L + U)x_m + b$

# BONACICH $\alpha$-CENTRALITY

- $x = (I - \alpha A^T)^{-1}s$
- Almost identical to Katz measure
- $s$ – in many cases vector of ones
- $\alpha$ – centrality

# TEMPORAL BONACICH $(\alpha,\beta)$-CENTRALITY

- Very similar to Katz's centrality measure
- Depends on two parameters α and β
- $c(\alpha, \beta) = \alpha(\boldsymbol{I} - \beta\boldsymbol{A})^{-1}\boldsymbol{Ae}$
- β affects how much of the node's influence is due to the node's neighborhood
- $\beta > 0$ - the status of the node increases with connections to influential nodes
- $\beta < 0$ - the status of the node decreases with connections to powerful nodes
- The difference between this measure and Katz's is that we allow $\beta < 0$

# HUBS AND AUTHORITIES

- The node represents a Web page
- Two scores are assigned:
  - the hub score $x_v$
  - the authority score $y_v$

$$\lambda y_v = \sum_{u:u\to v} x_u = \sum_u a_{uv} x_u = (\mathbf{A}^T x)_v$$

$$\mu x_v = \sum_{u:v\to u} y_u = \sum_u a_{vu} y_u = (\mathbf{A} y)_v$$

$$\xrightarrow{\text{Matrix form}}$$

$$\lambda\mu x = \mathbf{A}\mathbf{A}^T x, \quad \lambda\mu y = \mathbf{A}^T\mathbf{A} y$$

- The hub and authority scores are just the elements of the dominant eigenvectors of the matrices $AA^T$ and $A^T A$, respectively

# PAGERANK

- The centrality measure used by Google to rank Web pages
- Simulates the behaviour of a user browsing the Internet
- Two different views:
  - random walk on a graph
  - eigenvector of a network matrix
- A random walk is a stationary process on any undirected graph.
- The centrality of a node - number of times that the walker stops at the node in the random process

# PAGERANK

- Jumps – when user types an URL

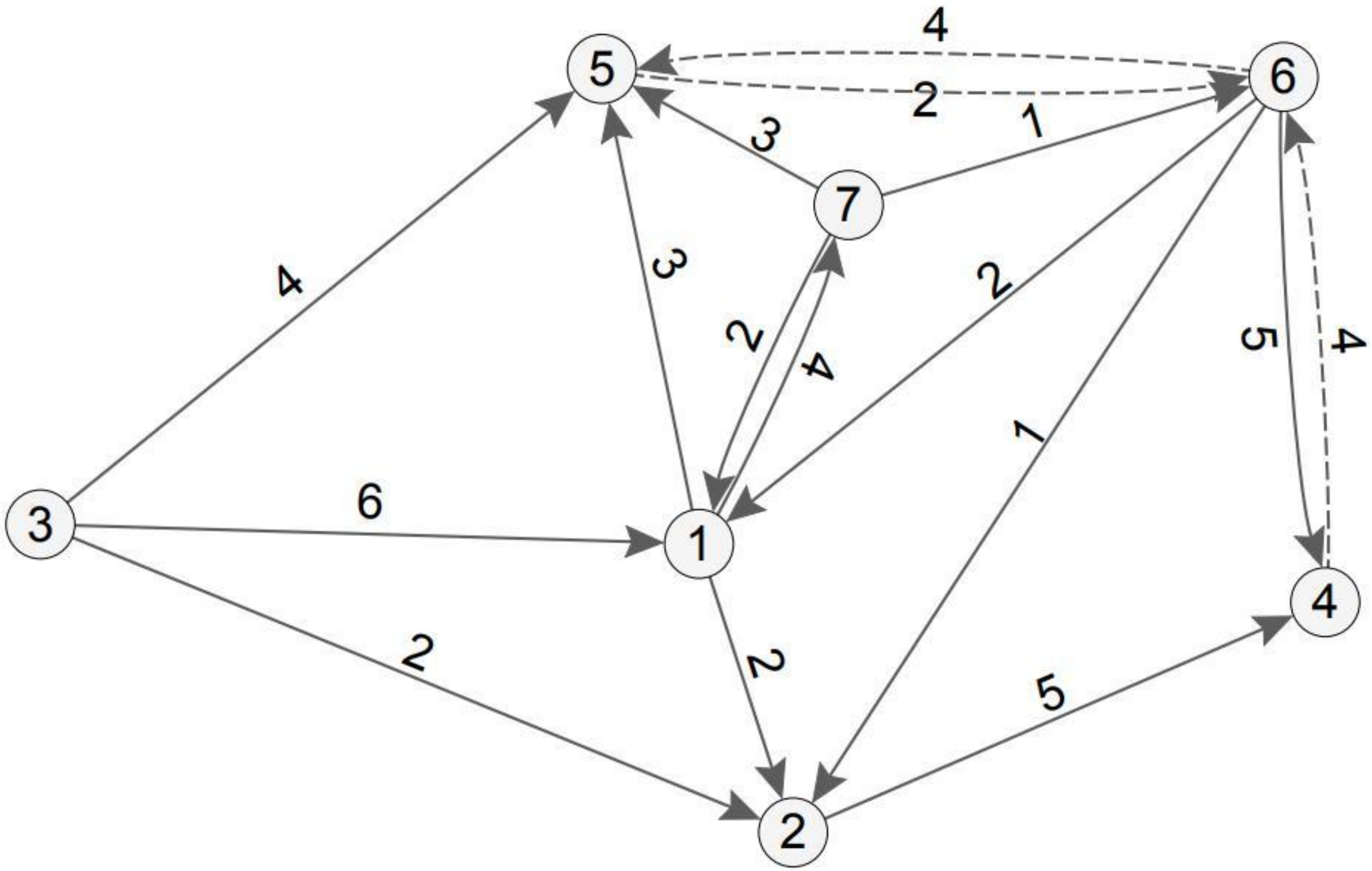
$$p_v = \frac{q}{n} + (1-q) \sum_{u:u \to v} \frac{p_u}{\text{outdeg}(u)}, \quad v = 1, 2, \ldots, n,$$

- $q$ – probability that a jump occurs
- $n$ - the number of nodes
- $p_v$ - the PageRank value of the node $v$

# PAGERANK

- $\boldsymbol{xM} = \boldsymbol{x}$
- $\boldsymbol{M} = \frac{q}{n}\mathbf{1} + (1-q)\boldsymbol{S}$
- $\boldsymbol{S} = \boldsymbol{D}^{-1}\boldsymbol{A}$
- $D$ - the diagonal matrix of outdegrees
- $\boldsymbol{S}_{vu} = \frac{\mathbf{1}}{\boldsymbol{n}}$, for all $u$

# RESULTS & NETWORK




Dotted links not present in the graph

- [1,3) – all weights equal to 1
- [3,5) – weights from figure

Dotted links present in the graph

- [5,7) – weights from figure
- [7,9) – all weights equal to 1

# IN-EIG

| | time 1-3 | time 3-5 | time 5-7 | time 7-9 |
|---|---|---|---|---|
| in-eig | 4,5,2,1,7,6 | 5,4,7,1,2,6 | 4,6,5,1,2,7 | 6,5,4,2,1,7 |

```
NodesInOrder(inEig(A13)[0])
```

[(4.0, 0.38323959281071945),
(5.0, 0.38323959281071945),
(2.0, 0.3428301504546229),
(1.0, 0.28929875962810886),
(7.0, 0.21838549181861674),
(6.0, 0.1648541009921027),
(3.0, 0.0)]

```
NodesInOrder(inEig(A35)[0])
```

[(5.0, 0.1977165029359666),
(4.0, 0.1647637524466388),
(7.0, 0.1178952801886321),
(1.0, 0.0953793431493424),
(2.0, 0.07020563073748974),
(6.0, 0.0364316809314975),
(3.0, 0.0)]

```
NodesInOrder(inEig(A57)[0])
```

[(4.0, 0.09060267067273253),
(6.0, 0.08822586927649721),
(5.0, 0.08401723940252281),
(1.0, 0.035348776020060625),
(2.0, 0.025354545633545576),
(7.0, 0.022558118955545263),
(3.0, 0.0)]

```
NodesInOrder(inEig(A79)[0])
```

[(6.0, 0.25055920117339614),
(5.0, 0.23178055289678445),
(4.0, 0.214409347589365),
(2.0, 0.1956306993127533),
(1.0, 0.15655198008676444),
(7.0, 0.07522857281001999),
(3.0, 0.0)]

# OUT-EIG

| | time 1-3 | time 3-5 | time 5-7 | time 7-9 |
|---|---|---|---|---|
| out-eig | 7,1,3,6 | 3,1,7,6 | 6,3,1,4,7,2,5 | 7,6,1,3,4,5,2 |

```
NodesInOrder(outEig(A13)[0])
```

```
[(7.0, 0.506723582443658),
 (1.0, 0.3825140269734168),
 (3.0, 0.288751382036271),
 (6.0, 0.288751382036271),
 (2.0, 0.0),
 (4.0, 0.0),
 (5.0, 0.0)]
```

```
NodesInOrder(outEig(A35)[0])
```

```
[(3.0, 0.24488220399751592),
 (1.0, 0.13207579228739402),
 (7.0, 0.10685161924706503),
 (6.0, 0.08162740133250532),
 (2.0, 0.0),
 (4.0, 0.0),
 (5.0, 0.0)]
```

```
NodesInOrder(outEig(A57)[0])
```

```
[(6.0, 0.08622967893067747),
 (3.0, 0.08488202453068001),
 (1.0, 0.05569943615737886),
 (4.0, 0.055028057334776594),
 (7.0, 0.044698255990850956),
 (2.0, 0.043895768514011745),
 (5.0, 0.027514028667388297)]
```

```
NodesInOrder(outEig(A79)[0])
```

```
[(7.0, 0.267193318164005),
 (6.0, 0.23465609673068194),
 (1.0, 0.20861825590466235),
 (3.0, 0.1804710611378959),
 (4.0, 0.11276009785212866),
 (5.0, 0.11276009785212866),
 (2.0, 0.05418503559278604)]
```

# KATZ

| | time 1-3 | time 3-5 | time 5-7 | time 7-9 |
|---|---|---|---|---|
| Katz a=0.15 | 2,5,1,4,7,6 | 5,4,7,1,2,6 | 5,6,4,1,7,2 | 5,6,2,1,4,7 |

```
NodesInOrder(katz(A13,0.15)[0])
```

[(5.0, 0.08701521909559679),
(2.0, 0.08602060537673469),
(1.0, 0.0793898501144133 7),
(4.0, 0.06373888070803742),
(7.0, 0.03518481798517937),
(6.0, 0.02855406272285805 3),
(3.0, 0.0)]

```
NodesInOrder(katz(A35,0.15)[0])
```

[(5.0, 0.5215015962375534),
(4.0, 0.4559213070532239),
(1.0, 0.342834378394488),
(7.0, 0.298806005247417),
(2.0, 0.22944660718280882),
(6.0, 0.06809724625058437),
(3.0, 0.0)]

```
NodesInOrder(katz(A57,0.15)[0])
```

[(4.0, 4.52872529768924),
(6.0, 4.37954829572241),
(5.0, 4.3772466717108145),
(1.0, 1.919885695487487 7),
(2.0, 1.349127210056215),
(7.0, 1.2449011485451351),
(3.0, 0.0)]

```
NodesInOrder(katz(A79,0.15)[0])
```

[(5.0, 0.1282818514384184),
(6.0, 0.1062280834891516),
(2.0, 0.0994596050860777),
(1.0, 0.0913091396246193 5),
(4.0, 0.07740583713681091),
(7.0, 0.03697271181379904),
(3.0, 0.0)]

# BONACICH $\alpha = 0.85$

| | time 1-3 | time 3-5 | time 5-7 | time 7-9 |
|---|---|---|---|---|
| Bonacich $\alpha$=0.85 | 1,2,5,4,6,7,3 | 1,4,5,2,7,6,3 | 5,1,4,6,2,7,3 | 5,1,2,6,4,7,3 |

```
NodesInOrder(alpha(A13,0.85)[0])
```

```
[(5.0, 2664175.6544733457),
 (4.0, 2664174.8044733456),
 (2.0, 2383260.5847311434),
 (1.0, 2011127.036358026),
 (7.0, 1518154.6764232314),
 (6.0, 1146021.128050114),
 (3.0, 1.0)]
```

```
NodesInOrder(alpha(A35,0.85)[0])
```

```
[(5.0, 5.477165055839387e+44),
 (4.0, 4.56430421319949e+44),
 (7.0, 3.26595023702488e+44),
 (1.0, 2.6422092445360162e+44),
 (2.0, 1.9448456147568724e+44),
 (6.0, 1.0092341260235758e+44),
 (3.0, 1.0)]
```

```
NodesInOrder(alpha(A57,0.85)[0])
```

```
[(4.0, 1.0561031039951371e+73),
 (6.0, 1.0283971904927055e+73),
 (5.0, 9.793404252479036e+72),
 (1.0, 4.120403866403615e+72),
 (2.0, 2.9554317897573465e+72),
 (7.0, 2.62946665246507e+72),
 (3.0, 1.0)]
```

```
NodesInOrder(alpha(A79,0.85)[0])
```

```
[(6.0, 2.3012704045561674e+25),
 (5.0, 2.1287973770505854e+25),
 (4.0, 1.96925066413978e+25),
 (2.0, 1.7967776366341986e+25),
 (1.0, 1.4378579899897374e+25),
 (7.0, 6.90939387060848e+24),
 (3.0, 1.0)]
```

# BONACICH $\beta = 0.15$

| | time 1-3 | time 3-5 | time 5-7 | time 7-9 |
|---|---|---|---|---|
| | 7,1,3,6,2 | 3,1,6,7,2 | 3,6,1,4,7,2,5 | 6,7,1,3,4,5,2 |

```
NodesInOrder(bonacich(A13,0.15)[0])
```

```
[(7.0, 1.4104424003845288),
 (1.0, 1.2892726381711272),
 (3.0, 1.2710971758218697),
 (6.0, 1.2710971758218697),
 (2.0, 0.34212897991273156),
 (4.0, 0.0),
 (5.0, 0.0)]
```

```
NodesInOrder(bonacich(A35,0.15)[0])
```

```
[(3.0, 1.9514280499813526),
 (1.0, 1.2052598669579724),
 (6.0, 0.9233241858397826),
 (7.0, 0.8852739968468359),
 (2.0, 0.3209978460933312),
 (4.0, 0.0),
 (5.0, 0.0)]
```

```
NodesInOrder(bonacich(A57,0.15)[0])
```

```
[(3.0, 1.427626109592854),
 (6.0, 1.4249361893716217),
 (1.0, 0.9386029386842797),
 (4.0, 0.8892820324670281),
 (7.0, 0.7469676528960576),
 (2.0, 0.7099128325802996),
 (5.0, 0.44464101623351404)]
```

```
NodesInOrder(bonacich(A79,0.15)[0])
```

```
[(6.0, 1.4700311721322585),
 (7.0, 1.289554293615113 2),
 (1.0, 1.1432910493683042),
 (3.0, 1.121351564557 4232),
 (4.0, 0.4949428518216443),
 (5.0, 0.4949428518216443),
 (2.0, 0.34867960757483535)]
```

# HUB

| | time 1-3 | time 3-5 | time 5-7 | time 7-9 |
|---|---|---|---|---|
| hub | 3,6,1,7,2 | 3,7,6,1,2 | 3,6,7,1,2,4,5 | 6,3,7,1,2,4,5 |

```
NodesInOrder(hits(A13)[0])

[(3.0, 0.0781394552750261 9),
 (6.0, 0.0615009881876785 7),
 (1.0, 0.05985780619541359 5),
 (7.0, 0.05985780619541359 5),
 (2.0, 0.00926459696947737 2),
 (4.0, 0.0),
 (5.0, 0.0)]
```

```
NodesInOrder(hits(A35)[0])

[(3.0, 0.00992624759750904),
 (7.0, 0.004464306426780718),
 (6.0, 0.004335129687445384 5),
 (1.0, 0.004109614640470102 4),
 (2.0, 0.001987414184883836),
 (4.0, 0.0),
 (5.0, 0.0)]
```

```
NodesInOrder(hits(A57)[0])

[(3.0, 0.006357327612411748),
 (6.0, 0.005812859783106858),
 (7.0, 0.003082619239216009),
 (1.0, 0.002876146304281888 3),
 (2.0, 0.001879805877198664 2),
 (4.0, 0.000149811255045801 42),
 (5.0, 7.490562752290071e-05)]
```

```
NodesInOrder(hits(A79)[0])

[(6.0, 0.0628208203101243 1),
 (3.0, 0.0554721630832782 1),
 (7.0, 0.0444814239995621 5),
 (1.0, 0.0429201509814759 3),
 (2.0, 0.007348657226846101 6),
 (4.0, 0.005892671334141675),
 (5.0, 0.005892671334141675)]
```

# AUTHORITY

| | time 1-3 | time 3-5 | time 5-7 | time 7-9 |
|---|---|---|---|---|
| authority | 1,2,5,4,6,7 | 1,5,2,4,7,6 | 5,1,4,2,7,6 | 5,1,2,4,6,7 |

```
NodesInOrder(hits(A13)[1])
```

```
[(1.0, 0.07218403342215594),
 (2.0, 0.07218403342215593),
 (5.0, 0.0715894646628137),
 (4.0, 0.02560498387011615),
 (6.0, 0.021658216557721894),
 (7.0, 0.021658216557721894),
 (3.0, 0.0)]
```

```
NodesInOrder(hits(A35)[1])
```

```
[(1.0, 0.008651661490188174),
 (5.0, 0.007336401090878623),
 (2.0, 0.003633830515415609),
 (4.0, 0.003544790312388904),
 (7.0, 0.0018432689486385183),
 (6.0, 0.0005005894365912132),
 (3.0, 0.0)]
```

```
NodesInOrder(hits(A57)[1])
```

```
[(5.0, 0.006580243340381865),
 (1.0, 0.00553007379502031),
 (4.0, 0.0038027262006044074),
 (2.0, 0.002400452555040212),
 (7.0, 0.0011374145980066043),
 (6.0, 0.00037882314845527225),
 (3.0, 0.0)]
```

```
NodesInOrder(hits(A79)[1])
```

```
[(5.0, 0.06656603801507417),
 (1.0, 0.05267639628760718),
 (2.0, 0.05217113325771372),
 (4.0, 0.02270795804629237),
 (6.0, 0.01820884133055525),
 (7.0, 0.013889641727466992),
 (3.0, 0.0)]
```

# PAGE RANK Q=0.15

| | time 1-3 | time 3-5 | time 5-7 | time 7-9 |
|---|---|---|---|---|
| pageRank q=0.15 | 4,2,5,1,7,6,3 | 4,5,7,1,2,6,3 | 6,4,5,7,2,3 | 6,4,5,2,1,7,3 |

```
NodesInOrder(pageRank(A13)[0])
```

```
[(4.0, 0.23673981724129228),
 (5.0, 0.16491165217601284),
 (2.0, 0.16185576702968474),
 (1.0, 0.15107022153276267),
 (7.0, 0.1130037854997067),
 (6.0, 0.10221824000278465),
 (3.0, 0.07020051651775622)]
```

```
NodesInOrder(pageRank(A35)[0])
```

```
[(4.0, 0.2252497396131958),
 (5.0, 0.19494269671728723),
 (1.0, 0.16033432938779257),
 (7.0, 0.1330227636150867),
 (2.0, 0.12270167223815273),
 (6.0, 0.09129683159234732),
 (3.0, 0.07245196683613755)]
```

```
NodesInOrder(pageRank(A57)[0])
```

```
[(6.0, 0.36197319746999307),
 (4.0, 0.20801783563595455),
 (5.0, 0.18285428471335385),
 (1.0, 0.09842135733839003),
 (2.0, 0.06869474713644604),
 (7.0, 0.05861000627729119),
 (3.0, 0.02142857142857143)]
```

```
NodesInOrder(pageRank(A79)[0])
```

```
[(6.0, 0.33280161964851196),
 (4.0, 0.20294550202358846),
 (5.0, 0.14552342294189938),
 (2.0, 0.13034888958695798),
 (1.0, 0.11339490360670149),
 (7.0, 0.05355709076376934),
 (3.0, 0.02142857142857143)]
```

# SOFTWARE

python

- Math
- Operator

NumPy

# THANK YOU FOR YOUR ATTENTION! ☺