lOMoARcPSD|17241975

**Experiment 10**

**Aim:** Implementation of Page rank/HITS algorithm

# **Theory:**

**PageRank:**

**PageRank (PR)** is an algorithm used by Google Search to rank websites in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages.

According to Google, PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

# **HITS:**

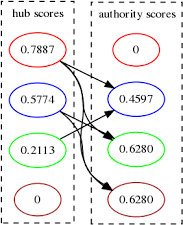
**Hyperlink Induced Topic Search (HITS)** Algorithm is a Link Analysis Algorithm that rates webpages, developed by Jon Kleinberg. This algorithm is used to the web link-structures to discover and rank the webpages relevant for a particular search.

HITS uses hubs and authorities to define a recursive relationship between webpages. Before understanding the HITS Algorithm, we first need to know about Hubs and Authorities.

* Given a query to a Search Engine, the set of highly relevant web pages are called **Roots**. They are potential **Authorities**.
* Pages that are not very relevant but point to pages in the Root are called **Hubs**. Thus, an Authority is a page that many hubs link to whereas a Hub is a page that links to many authorities.

# **HITS algorithm:**

* Let number of iterations be k.
* Each node is assigned a Hub score = 1 and an Authority score = 1.
* Repeat k times:
  + Hub update: Each node’s Hub score = ∑ (Authority score of each node it points to).
  + Authority update: Each node’s Authority score = ∑ (Hub score of each node pointing to it).
  + Normalize the scores by dividing each Hub score by square root of the sum of the squares of all Hub scores and dividing each Authority score by square root of the sum of the squares of all Authority scores. (optional)



# **Program:**

*import* networkx *as* nx

*import* matplotlib.pyplot *as* plt G = nx.DiGraph()

G.add\_edges\_from([('A', 'D'), ('B', 'C'), ('B', 'E'), ('C', 'A'),

('D', 'C'), ('E', 'D'), ('E', 'B'), ('E', 'F'),

('E', 'C'), ('F', 'C'), ('F', 'H'), ('G', 'A'), ('G', 'C'), ('H', 'A')])

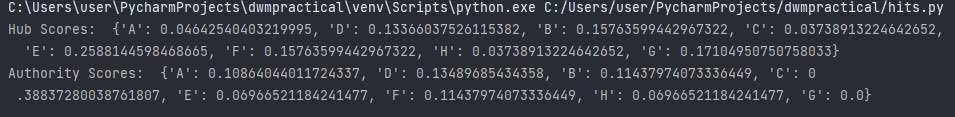
plt.figure(figsize=(10, 10)) nx.draw\_networkx(G, with\_labels=*True*)

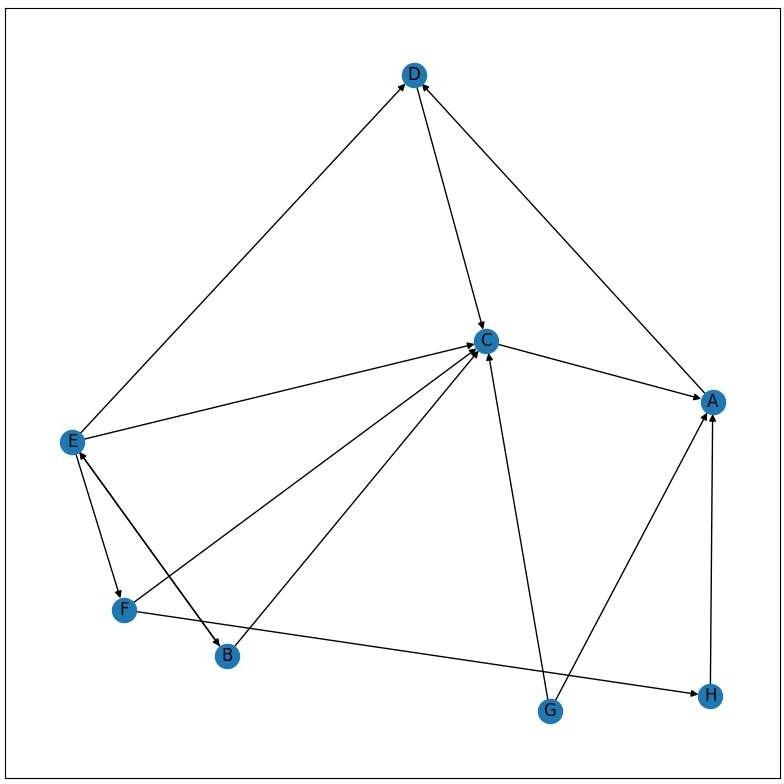
hubs, authorities = nx.hits(G, max\_iter=50, normalized=*True*)

*# The in-built hits function returns two dictionaries keyed by nodes # containing hub scores and authority scores respectively.*

print("Hub Scores: ", hubs) print("Authority Scores: ", authorities) plt.show()

**Output:**





**Conclusion:** Thus, we successfully implemented Page rank/HITS algorithm