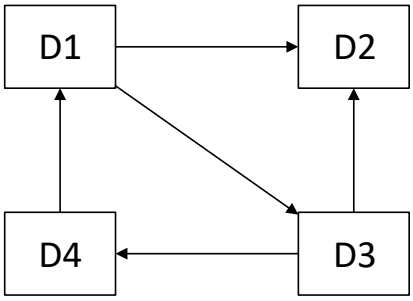
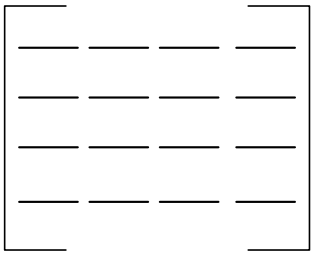
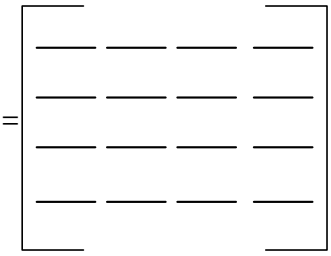
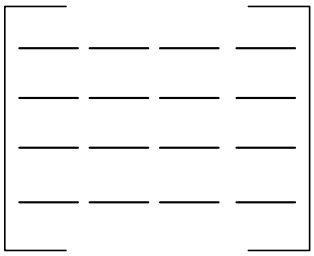
LAB 6: Exercises

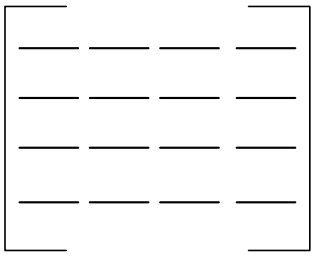
1. **HITS:** Given is the network shown in the image below. Find hubs and authorities vectors for this network. Complete the matrix L and L^T for this network and calculate matrix LL^T. Use online eigenvector calculator to find vectors h and a.



L = 

L^T = 

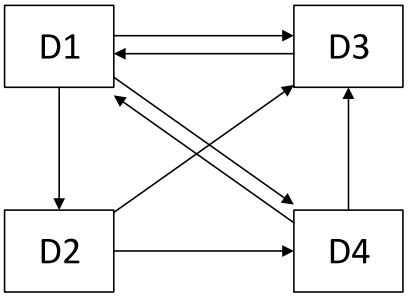
LL^T = 

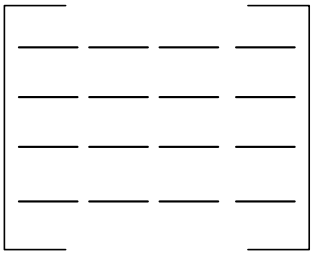
L^TL = 

h = [, , ,], a = [, , ,]
h_{norm} = [, , ,], a_{norm} = [, , ,]

The best hub is page: ……., the best authority is page: …….

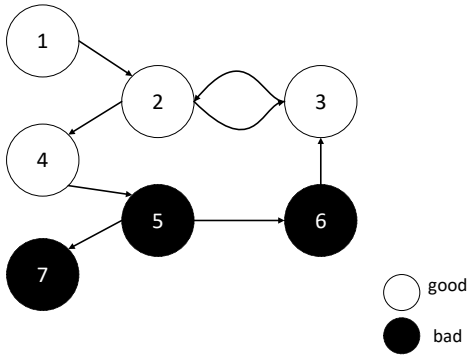
2. **PageRank:** Given is the network shown in the picture below. Find stochastic matrix M, write and solve the equation system for finding PageRank values for this network (use basic PageRank model – without a damping factor).



M = 

Equation system:

3. **TrustRank:** Find initial TrustRank vector d (seed = {2, 4, 5} and write equations for finding TrustRank for pages 2, 3, and 5, q = 0.15.



M =

d = [……., ……., …….,……..,……..,……..,……..]

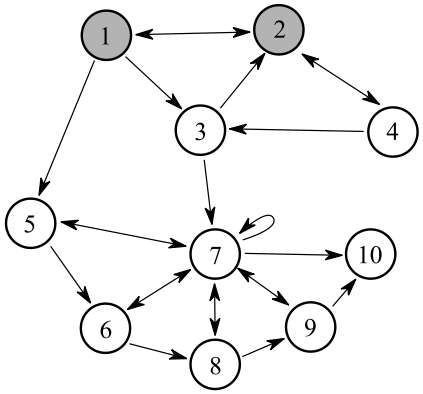
TR(2) =

TR(3) =

TR(5) =

4. **Programming Assignment (deadline +1 week)**

Given is the following web structure:



Download the pr_tr.py python script from the lab directory. The above structure is kept in L matrix (matrix of indices). Complete the TODOs:

TODO 1. Compute stochastic matrix M (function getM).

TODO 2. Compute pagerank vector and return the results (sorted pairs -> [page id : **pagerank**]). Which pages have the greatest pagerank? Why?

TODO 3. Which pages do you think belong to the link farm? Compute trustrank vector. Pages 1 and 2 are marked as “good”. Analyze the results. What has changed?

TODO 4. Repeat TODO3 but remove connections 1->5 and 3->7. Analyze the computed trustrank vector.