

```

1  numNodes = size(A)(1);
2
3  disp("Degree Centrality:\n")
4  degreeCentrality = A*ones(numNodes, 1);
5  [degreeCentralitySorted, sortedInds] = sort(degreeCentrality(:), 'descend');
6  top3Degree = sortedInds(1:3);
7  for i = sortedInds
8      disp(i);
9      disp(names(i));
10     disp(degreeCentralitySorted);
11 endfor
12
13 disp("\n Eigenvector Centrality:\n")
14 [V, D] = eig(A);
15 [~, max_idx] = max(diag(D));
16 eigenvectorCentrality = V(:,max_idx);
17 [eigenvectorCentralitySorted, sortedInds] = sort(eigenvectorCentrality(:),
18 'descend');
19 top3Eigen=sortedInds(1:3);
20 for i = sortedInds
21     disp(i);
22     disp(names(i));
23     disp(eigenvectorCentralitySorted);
24 endfor
25
26 disp("\nPage Rank:\n")
27 alpha = 0.85;
28 s = ones(num_nodes, 1);
29 D = diag(degreeCentrality);
30 pageRank = inv(eye(num_nodes) - alpha * A * inv(D)) * (1-alpha) * s;
31 [pageRankSorted, sortedInds] = sort(pageRank(:), 'descend');
32 top3PageRank = sortedInds(1:3);
33 for i = sortedInds
34     disp(i);
35     disp(names(i));
36     disp(pageRankSorted);
37 endfor
38
39 disp("\nCloseness Centrality:\n")
40 minDist = zeros(num_nodes, num_nodes);
41 visited = zeros(num_nodes, num_nodes);
42 iLenPaths = eye(num_nodes, num_nodes);
43 for i = 1:num_nodes
44     iLenPaths = iLenPaths*A;
45     for r = 1:num_nodes
46         for c = r+1:num_nodes
47             if visited(r,c) == 1
48                 continue;
49             endif
50             if iLenPaths(r, c) > 0
51                 minDist(r,c) = i;
52                 minDist(c,r) = i;
53                 visited(r,c) = 1;
54             endif
55         endfor
56     endfor
57 minDist = minDist * ones(num_nodes, 1);
58 closenessCentrality = (num_nodes-1)./minDist;
59 [closenessCentralitySorted, sortedInds] = sort(closenessCentrality(:), 'descend');
60 top3Close = sortedInds(1:3);
61 for i = sortedInds
62     disp(i);
63     disp(names(i));
64     disp(closenessCentralitySorted);
65 endfor
66

```