Task 2: Hubs, Authorities und PageRank (theoretical)

a) We have defined matrices $\mathbf{M}$ and $\mathbf{A}$ for the iterations. In this sub task we use the original HITS algorithm. Compute the matrices for the example graph.

We obtain matrix $\mathbf{A}$ by setting the component $a_{i,j}$ to 1 if there is a link from node i to node j. The rows in $\mathbf{A}$ contain all outgoing links and the columns in $\mathbf{A}$ contain all incoming links. Hence, we get (empty cells are 0):

$$\mathbf{A} = \begin{bmatrix}
 & & & & & 1 & & & & & & \\
 & & & & & 1 & 1 & & & & & \\
 & & & 1 & & 1 & & & & & & \\
 & & & & & & & 1 & & & & \\
1 & 1 & & & & & & & & 1 & & \\
 & & & & & & & & & & 1 & \\
 & & 1 & & & & & & & & & 1 \\
 & 1 & 1 & & & & & & & & 1 & \\
 & & & & 1 & 1 & & & & 1 & & \\
 & & & & & & & & & & 1 & \\
 & & & & & & & & & 1 & & \\
 & & & & & & & 1 & & & &
\end{bmatrix}$$

Similarly, we obtain the matrix $\mathbf{M}$ by setting the component $m_{i,j}$ to 1 over the number of outgoing links of node j if node j has a link to node i. Note that this leads to a transposed view compared to $\mathbf{A}$. So, rows contain that incoming links and columns the outgoing links. In our case, all nodes have outgoing links, so the special case from the script does not apply. We get:

$$\mathbf{M} = \begin{bmatrix}
 & & & & 1/3 & & & & & & & \\
 & & & & 1/3 & & & 1/3 & & & & \\
 & & & & & & 1/2 & 1/3 & & & & \\
 & & 1/2 & & & & & & & & & \\
 & & & & & & & & 1/3 & & & \\
1 & 1/2 & 1/2 & & & & & & 1/3 & & & \\
 & 1/2 & & & & & & & & & & \\
 & & & 1 & & & & & & & & 1 \\
 & & & & & & & & & & & \\
 & & & & 1/3 & & & & 1/3 & & 1 & \\
 & & & & & 1 & & 1/3 & & 1 & & \\
 & & & & & & 1/2 & & & & &
\end{bmatrix}$$

Task 2: Hubs, Authorities und PageRank (theoretical)

b) Write a small program (e.g., with MATLAB, but also works with Excel) that evaluates the fix-point iteration to obtain all results.

The following code is written for scilab (a free version of Matlab):

```
A = [0 0 0 0 0 1 0 0 0 0 0 0;
     0 0 0 0 0 1 1 0 0 0 0 0;
     0 0 0 1 0 1 0 0 0 0 0 0;
     0 0 0 0 0 0 0 1 0 0 0 0;
     1 1 0 0 0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 0 0 0 0 1 0;
     0 0 1 0 0 0 0 0 0 0 0 1;
     0 1 1 0 0 0 0 0 0 0 1 0;
     0 0 0 0 1 1 0 0 0 1 0 0;
     0 0 0 0 0 0 0 0 0 0 1 0;
     0 0 0 0 0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 0 1 0 0 0 0];
h(1:size(A,1),1) = sqrt(size(A,1))/size(A,1);
ho = zeros(size(A,1), 1);
a = h; ao = ho;
i = 0;
while (i < 100) && (norm(a-ao) > 1.0E-03)
    ao = a; ho = h;
    a = A'*ho; h = A*ao;
    a = a/norm(a); h = h/norm(h);
    i = i+1;
end
[s,auths]=gsort(a);
[s,hubs]=gsort(h);
auths
hubs


M = A'*diag(1./sum(A',1));
alpha = 0.85;
N = size(A,1);
r = ones(size(A,1), 1)./N;
ro = zeros(size(A,1), 1);
i = 0;
while (i < 100) && (norm(r-ro) > 1.0E-03)
    ro = r;
    r=(1-alpha)/N*ones(N,1)+alpha*M*ro
    i = i+1;
end
[s,ranks]=gsort(r);
ranks
```

Task 2: Hubs, Authorities und PageRank (theoretical)

c) For the example graph, determine the best hubs, authorities, and the documents with high PageRanks.

We get the following results for our example graph:

authority:	*6 > 10 > 2 > 5 > 1 > 11 > 4 > 7 > 3 > 12 > 8 > 9*
hub:	*9 > 5 > 2 > 3 > 1 > 8 > 11 > 6 > 7 > 10 > 4 > 12*
PageRank ($\alpha = 0.85$):	*11 > 10 > 6 > 8 > 3 > 2 > 4 > 7 > 12 > 1 > 5 > 9*

d) Apply the SALSA algorithm to the example graph. Does the order change compared to the original HITS algorithm?

We first need to compute the matrices $\mathbf{A}_S$ and $\mathbf{H}_S$ (we use here the subscript to distinguish from the adjacency matrix $\mathbf{A}$). This is the tricky part, especially as we want to build it with the help of the adjacency matrix $\mathbf{A}$ from subtask a). Let $\mathbf{W}_r$ be the matrix generated from $\mathbf{A}$ by dividing each entry from $\mathbf{A}$ by its row sum. Similarly, let $\mathbf{W}_c$ be the matrix generated from $\mathbf{A}$ by dividing each entry from $\mathbf{A}$ by its column sum. The matrix $\mathbf{A}_S$ is defined as:

$$A_S(i,j) = \sum_{q:q\rightarrow p_i \wedge q\rightarrow p_j} \frac{1}{L_{in}(p_i)} \cdot \frac{1}{L_{out}(q)}$$

As the columns in $\mathbf{A}$ contain all incoming links, matrix $\mathbf{W}_c$ contains the $\frac{1}{L_{in}(p_i)}$ values and $\mathbf{W}_r$ holds the $\frac{1}{L_{out}(q)}$ values. We obtain $\mathbf{A}_s = \mathbf{W}_c^\top \mathbf{W}_r$ and, similarly, $\mathbf{H}_s = \mathbf{W}_r \mathbf{W}_c^\top$ (→ transform the matrix multiplication into its sum notation). The scilab code is as follows:

```
Wr=diag(1./(sum(A,2)+1e-10))*A;
Wc=A*diag(1./(sum(A,1)+1e-10));
As=Wc'*Wr;
Hs=Wr*Wc';
h = ones(size(A,1), 1)./size(A,1);
ho = zeros(size(A,1), 1);
a = h; ao = ho;
i = 0;
while (i < 100) && (norm(a-ao)+norm(h-ho) > 1.0E-03)
      ao = a; ho = h;
      a = As'*ao; h = Hs'*ho;
      i = i+1;
end
[s,auths]=gsort(a);
[s,hubs]=gsort(h);
auths
hubs
```

Task 2: Hubs, Authorities und PageRank (theoretical)

d) Apply the SALSA algorithm to the example graph. Does the order change compared to the original HITS algorithm? [continuation]

We get the following results for our example graph:

authority (SALSA): 6 > 10 > 11 > 3 > 2 > 8 > 4 > 7 > 5 > 12 > 1 > 9
hub (SALSA): 8 > 5 > 9 > 7 > 2 > 3 > 4 > 12 > 6 > 10 > 11 > 1

For direct comparison, we had the following results from subtask c)

authority (HITS): 6 > 10 > 2 > 5 > 1 > 11 > 4 > 7 > 3 > 12 > 8 > 9
hub (HITS): 9 > 5 > 2 > 3 > 1 > 8 > 11 > 6 > 7 > 10 > 4 > 12

Discussions: SALSA works a bit differently then HITS. We see this with the authority value of node 11. With HITS, 11 has a smaller authority as it is not linked by nodes 6, 8, and 10 which are not among the best hubs. SALSA, however, assigns node 11 a high authority as it is co-linked by 8 with node 2 and 3 obtaining high shares of their authority values (and keeping a lot of its own authority as nodes 6 and 10 only link to 11). Similarly, node 8 has become a good hub as it links to the same node as the other good hubs 5 (both link to node 2) and 7 (both link to node 3).

Obviously, with such a small example it is difficult to assess which algorithm works better. We would need a more extensive test data set for that.