HW 7
Rob Schwartz

The graph shown in figure 5.9 has *n* nodes, each of which is directed to one single other node, with two exceptions. The first node directs to both itself and another node, and the final node is a dead end (i.e. does not direct to any other node). This graph is therefore essentially a straight line from the first node to the final node, with the aforementioned exception of the first node's self-direction.

With this structure in mind, we can create a link matrix for the graph:

$$L = \begin{array}{l} 1\ 1\ 0\ 0\ 0 \ldots\ldots\ldots\ 0 \\ 0\ 0\ 1\ 0\ 0 \ldots\ldots\ldots\ 0 \\ 0\ 0\ 0\ 1\ 0 \ldots\ldots\ldots\ 0 \\ 0\ 0\ 0\ 0\ 1 \ldots\ldots\ldots\ 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0 \ldots\ldots\ldots\ldots\ldots\ 0\ 0\ 1 \\ 0 \ldots\ldots\ldots\ldots\ldots\ 0\ 0\ 0 \quad \text{<- - nth row} \end{array}$$

The graph's transposed link matrix is therefore:

$$Lt = \begin{array}{l} 1\ 0\ 0\ 0 \ldots\ldots\ldots\ldots\ 0 \\ 1\ 0\ 0\ 0 \ldots\ldots\ldots\ldots\ 0 \\ 0\ 1\ 0\ 0 \ldots\ldots\ldots\ldots\ 0 \\ 0\ 0\ 1\ 0 \ldots\ldots\ldots\ldots\ 0 \\ 0\ 0\ 0\ 1 \ldots\ldots\ldots\ldots\ 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0 \ldots\ldots\ldots\ldots\ldots\ 1\ 0\ 0 \\ 0 \ldots\ldots\ldots\ldots\ldots\ 0\ 1\ 0 \end{array}$$

Due to the formatting concerns on Word, I'm going to describe the following vectors in horizontal format. We begin with ***h*** (the hubs vector) as a vector of all ones with size *n* (i.e. [1,1,1,1,1,1,1,1,...]). Multiplying ***h*** by *Lt* yields us the same ***h*** vector. Scaling that ***h*** vector yields us our first iteration of the ***a***, or authorities, vector. But, because ***a*** is already scaled to 1, it will remain the same as ***h***: [1,1,1,1,1,1,1,1,...].

We now multiply ***a*** by *L*. This operation yields us: [2,1,1,1,1,1,...,1,0]. Scaled, this becomes: [1, 1/2, 1/2, 1/2, 1/2, 1/2, ..., 1/2, 0]. This is our new ***h*** vector. Multiplying this vector by *Lt* yields our new ***a***: [

If we continue this operation (***h*** = *L**a***, ***a*** = *Lt**h***, etc.), we find the following pattern:

0th iteration: ***h*** = [1, 1, 1, ..., 1, 1] ***a*** = [1, 1, 1, ..., 1, 1]
1st iteration: ***h*** = [1, 1/2, 1/2, ..., 1/2, 0] ***a*** = [1, 1/2, 1/2, ..., 1/2, 1/2]
2nd iteration: ***h*** = [1, 1/4, 1/4, ..., 1/4, 0] ***a*** = [1, 1/4, 1/4, ..., 1/4, 1/4]
3rd iteration: ***h*** = [1, 1/8, 1/8, ..., 1/8, 0] ***a*** = [1, 1/8, 1/8, ..., 1/8, 1/8]

I believe the pattern is becoming clear. The first node will always be both the best hub and the best authority. The last node will never be a hub. All other nodes (*n*-2) have hub and authority values equal to 1/2^(iteration). Running this algorithm indefinitely will yield:

h = [1, 0, 0, 0, 0, ..., 0, 0]
a = [1, 0, 0, 0, 0, ..., 0, 0]

Both of these vectors will have size *n*.