1. Consider Kleinberg’s HITS algorithm for computing hubs and authorities. Let $G = (V, E)$ be the directed web graph of hyperlinks and let $A$ be its adjacency matrix. Let $h$ and $a$ be two vectors over the vertices, both initialized to the all 1’s vector. Suppose they are updated as follows:

$$a \leftarrow A^T A a; \quad h \leftarrow A A^T h.$$ 

Consider running this algorithm on the following graph: $V = \{u, v\}$ and $E = (u, v)$ (remember the edges are directed). What are the final values of the vectors $h$ and $a$?

2.a. Consider the following naive alternative to the Page Rank algorithm. Let $G = (V, E)$ be the directed web graph of hyperlinks. Let each vertex $v$ have a non-negative weight $w(v)$. Let $n(v)$ be the number of out-edges from $v$. Simultaneously, update all the weights as follows:

$$w(v) \leftarrow \sum_{(u,v) \in E} \frac{w(u)}{n(u)}.$$ 

Keep iterating this update until all values change by less than some preset small $\epsilon > 0$.

Note that we can also write this update as $w \leftarrow (DA)^T w$, where $A$ is the adjacency matrix of the graph, $w$ is the vector of vertex weights, and $D$ is the following diagonal matrix: $D(v, v) = 1/n(v)$ if $n(v) \geq 1$, $D(v, v) = 1$ if $n(v) = 0$, and $D(u, v) = 0$ for $u \neq v$.

Consider running this algorithm on the same graph as in Question 1. Suppose we begin with $w(u) = w(v) = 1$. What is the final value of these variables?

Comment: As noted in class, this naive algorithm is not effective when there are vertices (or even components) with no outedges.

b. Now consider the actual Page Rank algorithm.

We let $\|w\|_1$ denote $\sum_{v \in V} w(v)$. We let $p$ denote a fraction, say $\frac{1}{3}$ (for simplicity of calculation). Finally, we let $n$ denote the number of vertices in $G$.

$$w_0 \leftarrow 1 \quad \text{(the vector of all ones)}$$

$$i \leftarrow 0$$

repeat:

$$w_{i+1} \leftarrow \frac{p}{n} \cdot 1 + (1 - p) \cdot (DA)^T w_i$$

error $\leftarrow \|w_{i+1} - w_i\|_1$

$i \leftarrow i + 1$

until (error $\leq \epsilon$)

Suppose this algorithm is run on the 2-vertex graph from part (a). What is the final value of $w$? (Take $\epsilon = 1/10$.) What is the limit value of $w$ as $\epsilon \to 0$?

3. The limit value, the vector $\text{PR}(p)$ computed by the algorithm in problem 2b is given by the equation:

$$\text{PR}(p) = p1 + (1 - p)(DA)^T \text{PR}(p).$$
Consider the Personalized Page Rank (PPR) defined by the following equation:

$$\text{PPR}(p, u) = pe_u + (1 - p)(DA)^T\text{PPR}(p, u),$$

where $e_u$ is the vector with all zero entries except for the entry for $u$ which is equal to 1.

Show that $\text{PR}(p) = \sum_u \text{PPR}(p, u)$.

4. Suppose that the Page Rank restart rule at a node with no outedges is modified to be a restart with probability 1. In effect this modifies the adjacency matrix to be:

$$\tilde{A}(i, j) = \begin{cases} A(i, j) & \text{if outdegree(} \text{vertex } i \text{) } \geq 1 \\ 1 & \text{if outdegree(} \text{vertex } i \text{) } = 0 \end{cases}$$

Suppose we use $\tilde{A}$ in place of $A$ in the Page Rank algorithm. Can this change the computed Page Rank values?

5. Consider the Robust Page Rank paper by Andersen et al.

i. What is the goal of this paper? One sentence please.

ii. How does this paper propose to achieve this goal? One sentence again.

iii. For each approach considered, the paper computes an $f(\text{page})$ value for each page, which on average is different for spam and non-spam pages. Given this, explain how you would achieve a particular false positive rate (the fraction of non-spam pages that are reported as spam).

Hint. If all pages are reported as non-spam, then the false positive rate is 0%.