Assignment V

Exercise 1 *Write a* MATLAB*-function which makes a page ranking of webpages.*

To determine which websites are important GOOGLE uses roughly the following algorithm. First a connection matrix is made, where $A_{ij} = 1$ if page j contains a link to page i; otherwise $A_{ij} = 0$. Next, a Markov process is considered, where you follow with probability p an arbitrary link on the current page, whereas with probability $1 - p$ you jump to a completely arbitrary page (usually $p = 0.85$). Let A be an $(n \times n)$ matrix, where the j^{th} column contains c_j 1-s (i.e. page j contains c_j links). Then this Markov process is described by the matrix

$$B_{ij} = \begin{cases} \frac{p}{c_j} A_{ij} + \frac{1-p}{n} & \text{if } c_j \neq 0, \\ \frac{1}{n} & \text{if } c_j = 0. \end{cases}$$

Such a matrix (with $B_{ij} > 0$ for all i, j) has a positive dominant eigenvalue, i.e., the eigenvalue with largest absolute value is unique, real, and positive. Moreover, the corresponding eigenvector v has strictly positive elements ($v_j > 0$). This eigenvector represents the limiting long term distribution of this Markov process. The element v_j of this eigenvector is the page rank of page j.

The input variables are an $(n \times n)$ sparse connection matrix A, and optionally a list of associated websites (a cell array with n strings). The output is a vector with page rankings. It is **important** that the input matrix can be very large (GOOGLE has $n > 10^{10}$). Therefore, your function can *not* use matrix operations with full matrices. In particular, the matrix B can not even be formed.

1. Use the power method to determine the page rankings.
2. The function should also show (with `bar`) a picture of the result.
3. If the optional lists of websites is given as an input variable, the function should produce a list of the top sites (and their page rank).
4. On the course website you find the file `wwwdata.mat` containing the sparse $(n \times n)$ matrices `sp`n for $n = 100, 500, 2000, 10000, 50000, 200000$, with associated lists of websites `url`n. Depending on the speed and memory of your computer you can test your function on these matrices.