Project
Deadline 11 December 23:59 hrs

This project consists of problems and programming exercises on *Google Matrix* and the *PageRank* algorithm, which is one the main reasons that makes Google a very popular search engine. The algorithm ranks webpages based on its links to other webpages.

Submit ONE .pdf and ONE .m or .ipynb file via MS Teams.

Deadline extension requests will not be entertained.

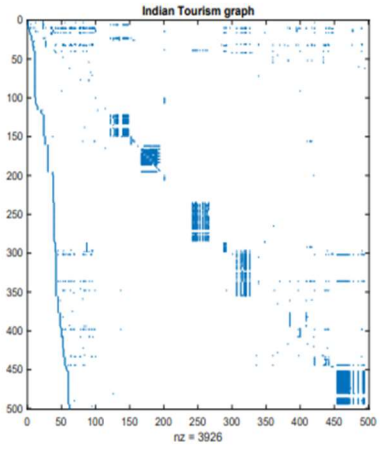
Let U be the set of webpages that can be reached by following a chain of hyperlinks starting at some root page, and let n be the number of pages in U. Let $\mathbf{G}$ be the $n \times n$ connectivity matrix of a portion of the Web, i.e., $g_{ij} = 1$ if there is a hyperlink to page i from page j and $g_{ij} = 0$ otherwise. The matrix $\mathbf{G}$ can be huge, but it is often very sparse. Its jth column shows the links on the jth page. The number of nonzeros in $\mathbf{G}$ is the total number of hyperlinks in U. The quantity out-degree of the jth page is the column sum of $\mathbf{G}$, i.e., $c_j = \sum_i g_{ij}$.

Let p be the probability that the random walk follows a link. Then $1-p$ is the probability that some arbitrary page is chosen and $(1-p)/n$ is the probability that a particular random page is chosen. Let us define the matrix $\mathbf{A}$ with its (i,j)th entry as

$$a_{ij} = \begin{cases} pg_{ij}/c_j + (1-p)/n, & \text{if } c_j \neq 0, \\ 1/n & \text{if } c_j \neq 0. \end{cases}$$

The jth column is the probability of jumping from the jth page to the other pages on the Web. If the jth page is a dead end with no out-links, then we assign a uniform probability of $1/n$ to all the elements in its column. The matrix $\mathbf{A}$ is the transition probability matrix of the Markov chain and the Perron vector or the eigenvector $\mathbf{x}$ associated to the leading eigenvalue of $\mathbf{A}$, i.e., the vector $\mathbf{x}$ that satisfies $\mathbf{Ax} = \mathbf{x}$ with a scaling factor $\mathbf{1}^T\mathbf{x} = 1$ is the state vector of the Markov chain and is the so-called Google's PageRank.

Download the two datasets `IndianTourism.mat` and `Harvard.mat`, which contains information about a subset 500 webpages and the corresponding hyperlink graph obtained by accessing Indian tourism's and Harvard University's webpages, respectively. The variable `U` contains the list of URLs and `G` contains the connectivity matrix. The Matlab or Python command `spy(G)` or `matplotlib.pyplot.spy(G)` produces a spy plot of the sparse connectivity matrix as shown below



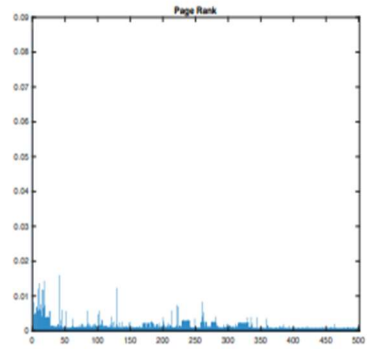
Develop a function based on the power method to compute Google's PageRank vector as

```
x = PageRank(U,G,p)
```

Choose $p = 0.85$, but vary p to understand its impact.

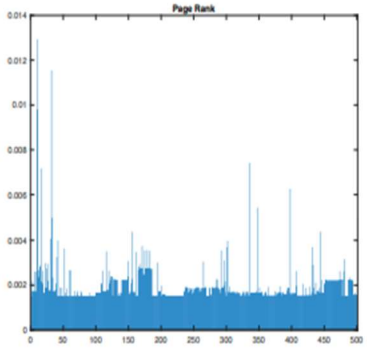
Report the outputs for the two graphs as a bar graph of ranks and print the top most highly ranked URLs in the PageRank order as shown in the next page. Comment on your observations about the PageRank and the graph structure.

Harvard graph



	page-rank	in	out	url	
1	0.0865	195	26	http://www.harvard.edu	47
42	0.0161	42	0	http://search.harvard.edu:8765/custom/query.html	73
18	0.0144	45	46	http://www.gse.harvard.edu	51
10	0.0137	21	18	http://www.hbs.edu	43
130	0.0124	24	12	http://www.med.harvard.edu	51
9	0.0121	21	27	http://www.ksg.harvard.edu	51

Indian tourism graph



	page-rank	in	out	url	
10	0.0129	69	2	http://www.nic.in	42
32	0.0115	36	0	http://india.gov.in	44
11	0.0098	59	61	http://cmf.gov.in	42
336	0.0074	58	2	http://pgportal.gov.in	47
16	0.0072	40	0	http://drupal.org)	43
398	0.0063	31	1	http://www.makeinindia.com/home	56
348	0.0054	26	3	http://www.digitalindia.gov.in	55