

# Implementing the pagerank algorithm

## CS F469 IR Assignment 2

### Design Documentation

#### Team Members:

1. [Subhadip Jana 2015A7PS0039H](#)
2. [Shardul Parab 2015A7PS093H](#)
3. [Himanshu Gupta 2015A3PS0339H](#)

#### **Packages:**

#### Various packages used in this assignment:

##### 1. Scipy

- a. SciPy is an open source Python library used for scientific computing and technical computing.
- b. SciPy contains modules for optimization, linear algebra, integration, interpolation.
- c. Scipy.sparse can be to represent sparse matrix, some examples being Compressed sparse row matrix, Row-based linked list sparse matrix etc. Compressed row matrix was used to represent the various matrices in the pagerank algorithm.

##### 2. Numpy

- a. NumPy is the fundamental package for scientific computing with Python. Numpy can be used to efficiently represent N - dimensional array objects, and perform matrix operations on them.

##### 3. iGraph

- a. igraph is a library collection for creating and manipulating graphs and analyzing networks. We used this package to visualise the pagerank algorithm, including per iteration visualisation.

#### 4. Imageio

- a. Imageio is a Python library that provides an easy interface to read and write a wide range of image data, including animated images.
- b. We used imageio to stitch .png images into an animated gif.

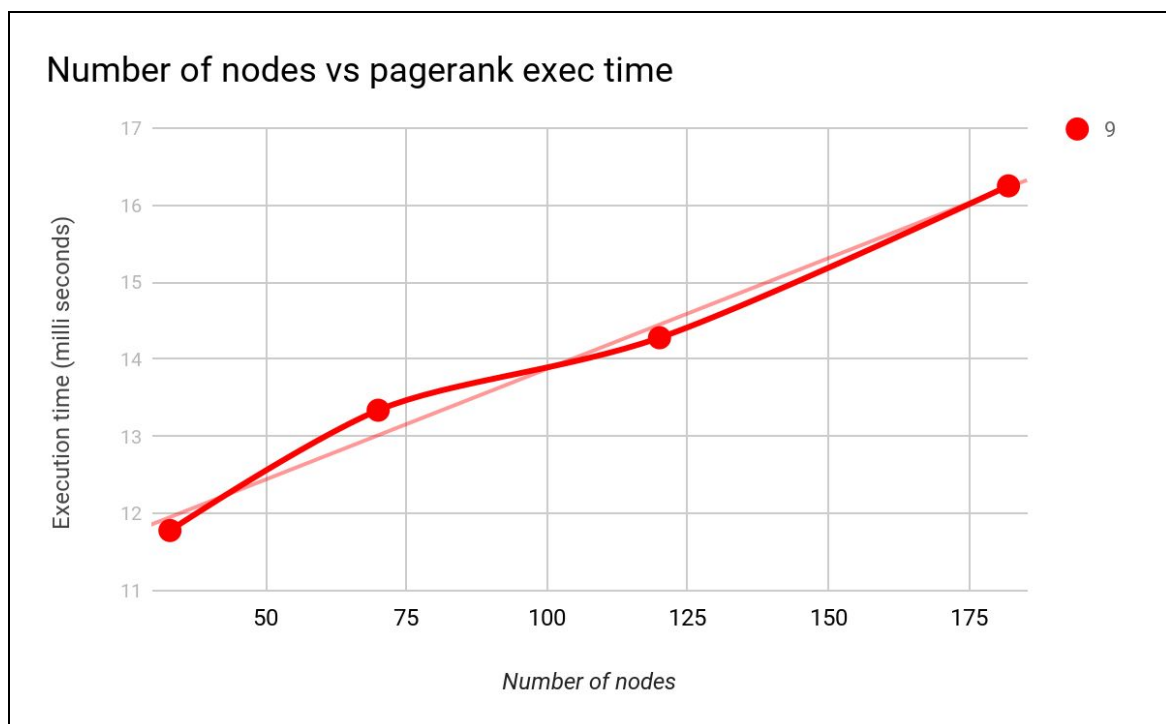
#### **Input Data format:**

Web graph data to apply pagerank -> .txt file,a list of links :

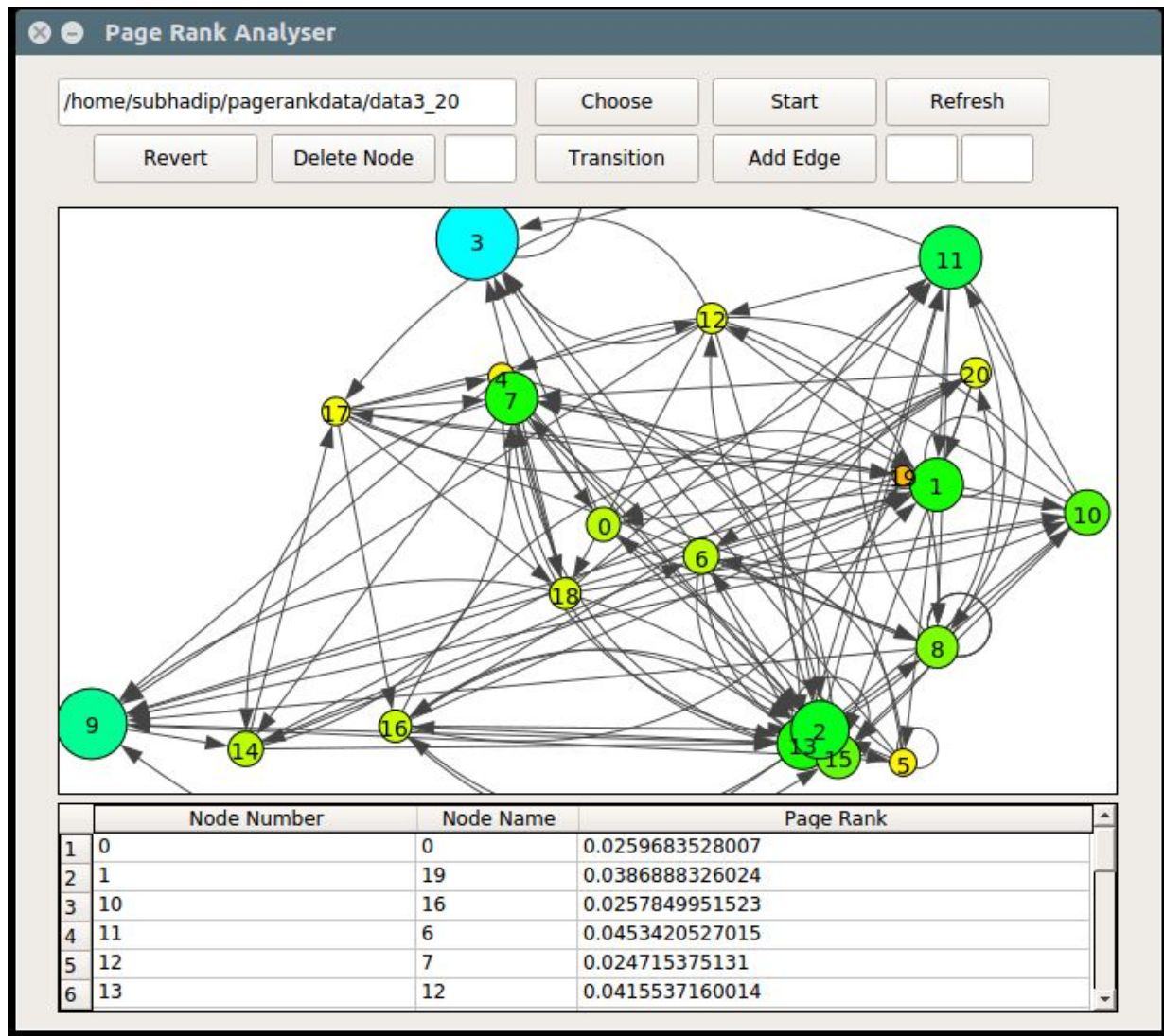
```
-----  
URL1 URL2  
URL3 URL4  
.  
.  
-----
```

#### **Execution Time Analysis:**

We tested the pagerank algorithm (utilising sparse matrix representation) on randomly generated graphs, simulating a web graph.



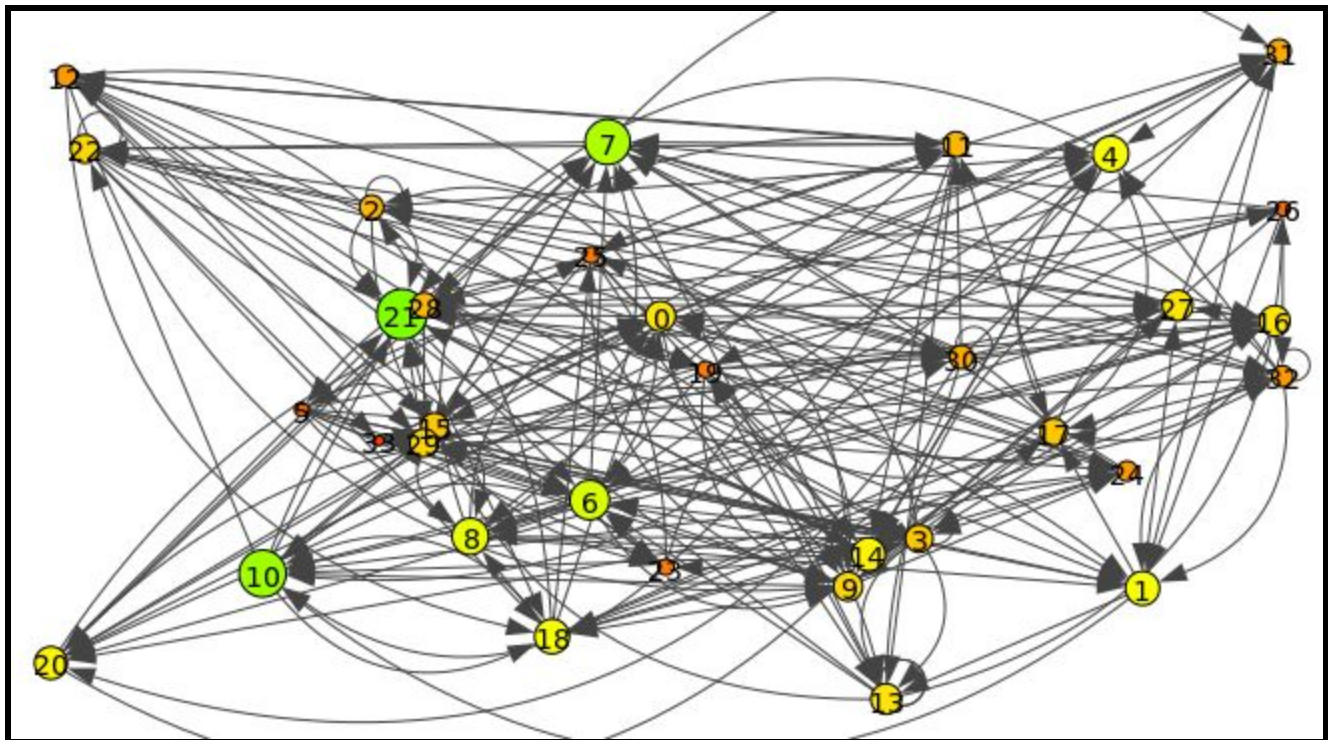
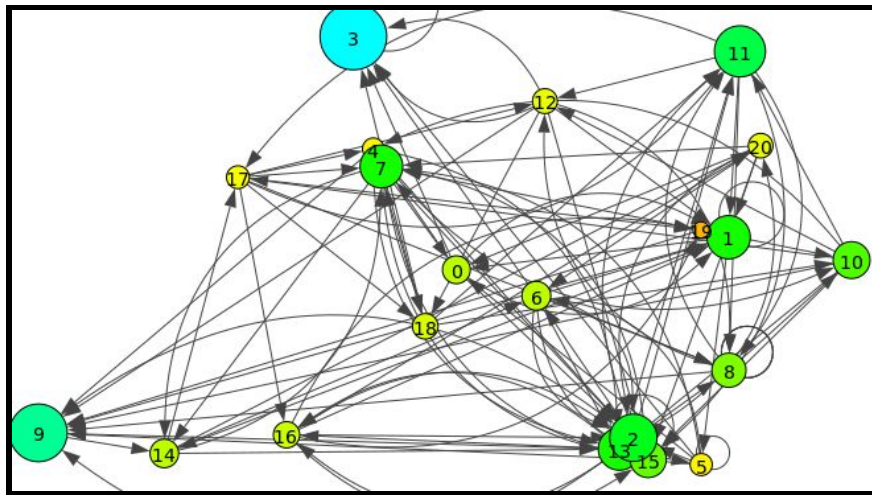
## Pagerank analysis application:



### Various elements in the GUI:

- Choose button: Choose a txt file containing links data.
- Start button: Start the pagerank algorithm.
- Delete button: Delete a node as entered in the text box.
- Add button: Add an edge as entered in the the two corresponding text boxes.
- Refresh button: Reapply page rank after modifying the web graph.
- Revert button: Undo all changes to web graph.
- Transition button: Visualising each iteration and change in each page's score graphically
- A table to check the URL (node name) of a node and corresponding pagerank value.

## PageRank visualisation:



Every node(URL) is assigned a unique integer, range  $0 \leq N < \text{number\_of\_nodes}$ . The color and size of a node depends on its page rank. Higher the pagerank, more bluish is the color, while lower the pagerank more reddish is the node color. Nodes having moderate pagerank value or those nodes that fall in between in terms of node ranking having green/yellowish color.

The application also provides the option of deleting nodes or adding more edges in the web graph, to analyse how the pagerank of the graph nodes change.

The transition button displays using a gif, how the pagerank rank matrix changes (and consequently the visualisation) across iterations during power iteration.

\*\*\*