Part A

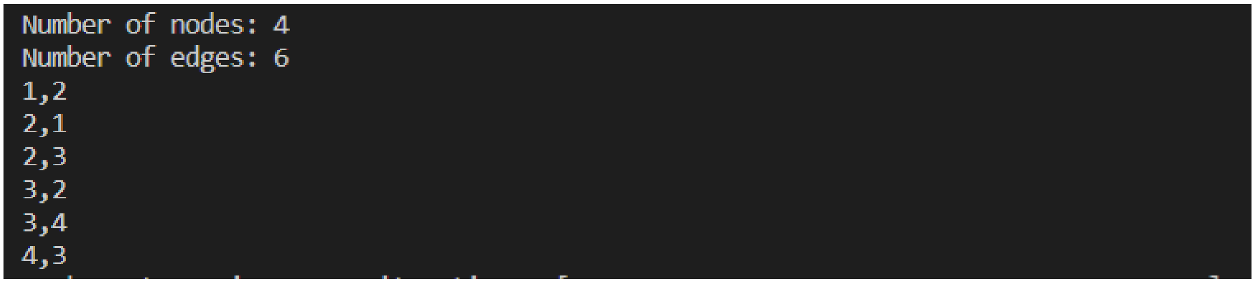
Page rank

PageRank is a way of measuring the importance of website pages. Page Rank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites. Our code has 4 major implementations

- Making the adjacency matrix
- Including the random teleportation
- Calculating rank vector using linear algebra packages
- Calculating rank vector using power iteration method

Inputs:

Takes in number of nodes and edges followed by the edges from one node to the next.



```
Number of nodes: 4
Number of edges: 6
1,2
2,1
2,3
3,2
3,4
4,3
```

The program then finds the rank matrix using two different method:

1. Using linear algebra packages

   By calling the pack() function, with adjacency matrix as a parameter.

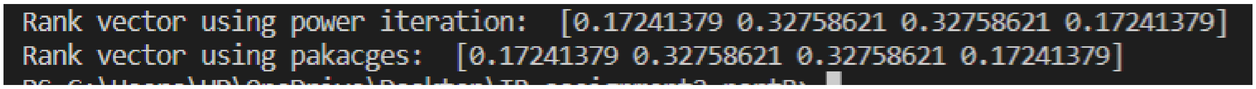
2. Using power iteration method

   By calling the powerItr() function, with rank vector and adjacent matrix as parameters.

Each function also calls randTele() function which implements random teleportation with probability of random teleportation 0.1

Output:

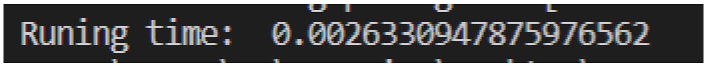
Output is rank vector calculated by the two methods



```
Rank vector using power iteration:  [0.17241379 0.32758621 0.32758621 0.17241379]
Rank vector using pakacges:  [0.17241379 0.32758621 0.32758621 0.17241379]
```

Running time:

The running time is between 0.0025 to 0.003 seconds.



```
Runing time:  0.0026330947875976562
```

Part B

HITS

Our code returns the near-steady state values of the Hub & Authority scores of the nodes in the given web graph.
Our code performs following operations:

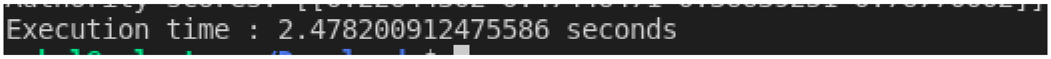
- **Preprocessing:** For preprocessing the graph dataset, we do stopword removal, tokenization, and then followed by Porter Stemming.
- **Root set generation:** generates the root set by taking query as input parameter.
- **Base set generation:** generates the base set from the root set
- **Adjacency matrix for the base set:** We also show the adjacency matrix based upon the base set generated for a particular query.
- **Calculation of Hub and Authority score:** We finally calculate the hub and authority score vector for the nodes in the base set.

Algorithm we used for calculating hub and authority score :

```
a(0) = (1, …, 1)T , h(0) = (1, …, 1)T
h(i+1) = A a(i)
h(i+1) = h(i+1) / | h(i+1) | // re-normalize h
a(i+1) = AT h(i)
a(i+1) = a(i+1) / | a(i+1) | // re-normalize a
```

When the code is run, the user enters a single word query . Based on the query root set and base set is generated.

The execution time for our algorithm is(including time taken to input query):



```
Execution time : 2.478200912475586 seconds
```

Overall, we use 3 functions in our implementation (apart from main):

1) makeBaseSet(): finds the base set
2) makeRootSet(): finds the root set
3) out_edges(): finds the list of outgoing edges