Computing PageRank for a set of nodes in a directed graph using NetworkX

Description:
In order to compute the pagerank for a set of nodes in a directed graph, networkx library of Python is used. Initially the graph is created as DiGraph by the networkx library and the inbuilt page rank function is invoked over this graph.

Parameter configurations: Damping parameter =0.85, Max iteration =100, convergence error = 10^{-6}

Input File Format: If G= [Q12→Q144, Q17→Q21, Q50→Q171,….]

```
Q12 Q144
Q17 Q21
Q50 Q171
…
```

Output File Format: Node_name Pagerank_score

```
Q12 0.12893
Q144 0.0002232
Q17 0.079722
….
```

Variants: Instead of using inbuilt pagerank package of networkx, packages like **pagerank_numpy, pagerank_scipy** can also be used.

Run format: nx_pagerank.py <input_file_path> <output_file_path>

Computing PageRank for a set of nodes in a directed graph using OpenTapioca

Description:
In order to compute the pagerank for a set of nodes in a directed graph, OpenTapioca is used. Initially the graph is represented as adjacency list where for each node, a set of nodes it points to along with the number of occurrences of the edge in the graph which is then converted into a numpy sparse adjacency matrix to be given as input. The output is a dense matrix.

Installing OpenTapioca using Solar:
Install Java version >= 1.8.0
For installing Solar please refer to https://lucene.apache.org/solr/
And install Solar version >=7.4.0 from http://archive.apache.org/dist/lucene/solr/
For installing OpenTapioca refer, https://opentapioca.readthedocs.io/en/latest/install.html

Input File Format: If G= [Q12→Q144, Q17→Q21, Q12→50, Q17→Q21, Q17->Q81….]
Q12 [Q144,Q50] [1,1]
Q17 [Q21,Q81] [2,1]
…

Output File Format: Numpy Dense matrix output.npz

Run format:
Sort nodes in numerical order

```
sort -n -k 1 unsorted_input.tsv > sorted_input.tsv
```

Convert sorted nodes into Numpy Sparse Adjacency Matrix

```
tapioca compile sorted_input.tsv
```

Compute Pagerank from Numpy Sparse Matrix

```
tapioca compute-pagerank sorted_input.npz
```

Output will be stored as **sorted_input.pagerank.npy**