Project: network analysis

1. Random graphs generated for:

    gnp1 = nx.gnp_random_graph(100, 0.05, seed=1234)

    gnp2 = nx.gnp_random_graph(2000, 0.01, seed=5130303)

    gnm1 = nx.gnm_random_graph(100, 1000, seed=27695)

    gnm2 = nx.gnm_random_graph(1000, 100000, seed=9999)

    Processing graph gnm1

        Power Law calculations:

            2.887544867499438

            Between 2 and 3 hence scale free

    Processing graph gnm2

        Power Law calculations:

            9.620663765606892

            Not between 2 and 3 hence not scale free

    Processing graph gnp2

        Power Law calculations:

            54.58226131097677

            Not between 2 and 3, not scale free

    Processing graph gnp1

        Power Law calculations:

            4.939081128857231

            Not between 2 and 3 not scale free

    Stanford graphs:

    amazon.graph.large

Power Law calculations:

1.3255773302967864

The graph is not scale free

amazon.graph.small.csv

Power Law calculations:

2.3948604146303003

The graph is scale free

dblp.graph.large

Power Law calculations:

1.3143917250729842

The graph is not scale free

dblp.graph.small

Power Law calculations:

1.6077866723222844

The graph is not scale free

youtube.graph.large

Power Law calculations:

1.560511013438854

The graph is not scale free

youtube.graph.small

Power Law calculations:

1.367441761614622

The graph is not scale free

2.

2.1) calculating the shortest path distance and finding the inverse of all neighbours. More closeness implies more neighbours with less distance.

o/p: +---+-------------------+

```
20/02/19 11:57:47 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
Reading in graph for problem 2.
Generating GraphFrame.
Calculating closeness.
+---+-------------------+
| id|          closeness|
+---+-------------------+
|  F| 0.07142857142857142|
|  C| 0.07142857142857142|
|  H| 0.06666666666666667|
|  D| 0.06666666666666667|
|  B|0.058823529411764705|
|  E|0.058823529411764705|
|  G| 0.05555555555555555|
|  A| 0.05555555555555555|
|  I|0.047619047619047616|
|  J|0.034482758620689655|
+---+-------------------+
```

2.2)

The two machines that have the maximum closeness would be the best candidates to hold the data. The closeness is measured by the number of nodes connected directly with least distance.

Here C and F with closeness : [07142857142857142] and [07142857142857142] would be the best candidates.

3.

3.1

In the example of terrorist network, the isolation of articulation points can be the best target to disrupt the communication in the organisation.

This is calculated by calculating the connected components before and after removal of that node/ terrorist. If the number of connected components increase, then it is an articulation point, removal of which can disrupt the communication in the organisation.

Here the following are the the best targets.

```
Processing graph using Spark iteration over nodes and serial (netwo
dness calculations
Execution time: 330.215735912 seconds
Articulation points:
+----------------------+------------+
|id                    |articulation|
+----------------------+------------+
|Mohamed Atta          |1           |
|Usman Bandukra        |1           |
|Mamoun Darkazanli     |1           |
|Essid Sami Ben Khemais|1           |
|Djamal Beghal         |1           |
|Nawaf Alhazmi         |1           |
|Raed Hijazi           |1           |
+----------------------+------------+
```


+----------------------+------------+

|id |articulation|

+----------------------+------------+

|Mohamed Atta |1 |

|Usman Bandukra |1 |

|Mamoun Darkazanli |1 |

|Essid Sami Ben Khemais|1 |

|Djamal Beghal |1 |

|Nawaf Alhazmi |1 |

|Raed Hijazi |1 |

+----------------------+------------+