Data Set: LastFM Asia Social Network
(**https://snap.stanford.edu/data/amazon0302.html**)

Disclaimer:
A different data set was chosen because the previous dataset had 300 thousand nodes and crashed before the code could even finish running.

About the dataset :
The Nodes are LastFM users from Asian countries, and the edges are follower relationships between them (both follow each other). *The graph is undirected.*

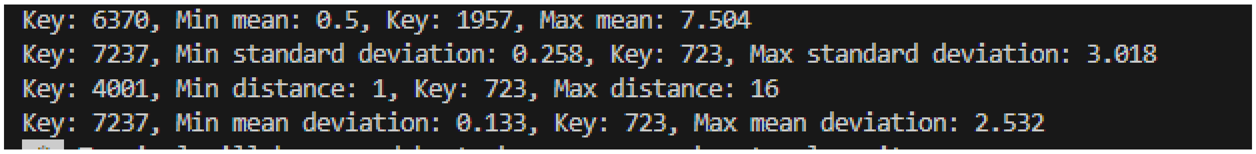
Methods:
A read function is created to read a file with two columns (node, edge). The function keeps track of the node, and edge pairs read to ensure no duplicates. (node, edge) is added to a vector of a vector representing an adjacency list. In this case, though the data set is set to be undirected, it could be disconnected at some connections.

The adjacency list is then passed to a BFS function, which traverses the whole list to calculate the distances between the node and edges. A HashMap of the node and a vector of distances are returned.

4 stat functions were made to compute the minimum and maximum mean, standard deviation, minimum and maximum distances, and mean deviation of the distances of each node. The associated keys are returned as well. The condition of min values not being equal to 0 is also a requirement for these functions. We aren't concerned with values of 0 since this means the specific node can only reach itself.

Context:
The standard and mean deviations are calculated with each node since I wanted to see how uniform the distances are. Lower values indicate little distance between the distances for that specific node and the opposite if the values are high. Standard deviation gets affected by outliers but mean deviation doesn't, so both were used to test the uniformity/spread. The mean is done the same way since it's more meaningful to see which node has the highest or lowest average distances. The min and max distances are the same, seeing which node has the lowest and highest distances. As always, there could be multiple nodes with the same minimum and maximum results, so I chose to return one for easier comparison.

Findings



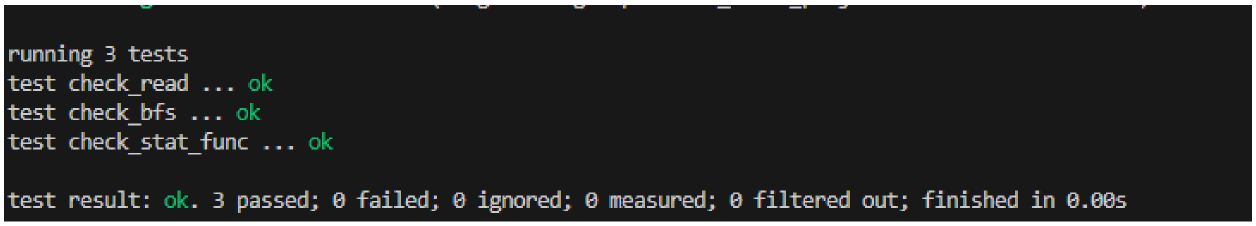
```
Key: 6370, Min mean: 0.5, Key: 1957, Max mean: 7.504
Key: 7237, Min standard deviation: 0.258, Key: 723, Max standard deviation: 3.018
Key: 4001, Min distance: 1, Key: 723, Max distance: 16
Key: 7237, Min mean deviation: 0.133, Key: 723, Max mean deviation: 2.532
```

The node changes every run, but the values are the same. In this specific run, we see the nodes associated with the minimum and maximum values from the stat functions. The key is the node in this case. We can see the least mean is 0.5, the standard deviation is 0.258, the minimum distance is 1, and the minimum mean deviation is 0.133. There could be multiple nodes with some or all of these values. If a node has a low mean it means it is the closest to all other nodes on average, and if the standard deviation and/or mean deviation is low, the distances are very close to each other (specifically the mean). We can see an example of node 7237 having the lowest standard deviation and mean deviation which we can conclude the distances from this node to other nodes is uniform. Node 6370 has the lowest mean, which means on average it is the closest to other nodes. These observations are the opposite for maximum values, high mean equates to furthest from other nodes on average. High standard deviation and mean deviation represent distances being far from each other and specifically the mean.

Tests

3 tests were made, to check the read function, the BFS algorithm, and the stat functions. The test for the read function is to test if the adjacency list is built correctly, and the BFS test is to make sure the distances are calculated correctly. The stat functions only test the values not the keys since the keys vary every run.

<u>All Passed</u>



```
running 3 tests
test check_read ... ok
test check_bfs ... ok
test check_stat_func ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

Websites Used

- https://www.programiz.com/rust/hashset
- https://crates.io/crates/round