

Programming challenge:

The text file contains a snapshot of the structure of the Internet at the autonomous system level in adjacency list format. The $n = 22\,963$ vertices in the network are numbered in order from zero (note: not from 1) up to 22962, with one line of the file for each one. A single line of the file looks, for example, like this:

```
112 3 12 22 24
```

This means that vertex 112 has degree 3, and its three neighbors are the vertices numbered 12, 22, and 24.

Write a program in the programming language of your choice to do the following:

- (i) Read the data from the file into a data structure of your choice, to represent the network, in adjacency list format, in the memory of the computer. For example, in C you might use a dynamically allocated 2D integer array with a line for each vertex, and another 1D array for the degrees. In Python you might use an array of n lists or sets to store the neighbors of each vertex, and an array of integers for the degrees.
- (ii) Write code to calculate, using breadth-first search, the shortest distance from a given single vertex to every other, then average those distances to calculate the closeness centrality of the given vertex.

Use your program to calculate the closeness centrality of the vertex numbered 0 in the network.

For full extra credit, turn in a complete printout of your program, and a printout of it in action, showing the answer it finds.

(Note: There are programs or libraries that you could download from the Internet that will do the closeness calculation for you. While it is perfectly legitimate under other circumstances to make use of such programs, doing so will not get you any credit on this question. You have to write your own breadth-first search program to get credit on this question.)