Task 10: Properties of networks

Step 0: Data

In this work, we will experiment with the data `simple_network.edges`, which is derived from this data described in this paper. Familiarise yourself with this data.

Step 1: Graph visualisation

In this step you are given an undirected, unweighted graph and asked to visualise and calculate various network statistics. To visualise your graph, you will first need to download Gephi from here. Once downloaded, load the `simple_network.edges` file and select the undirected option. Use the ForceAtlas 2 layout (see the left side of the screen) to arrange the graph. Several clusters should appear in the network.

You will now ask Gephi to determine three statistics for you. First, we will compute the **diameter**, which is a measure of the longest shortest path between any two vertices, i.e. $\max_{u,v} d(u,v)$ where $d(u,v)$ is the shortest path between any two vertices u and v. In Gephi, run the network diameter statistic (right hand side), and you should get 8. The next two network statistics are vertex **degree** and **betweenness centrality**. The degree of a vertex is the number of edges incident to it. Betweenness centrality is a measure of centrality that will be covered in detail in the next session. Use the appearance pane (top left) to give your vertices a size that depends on their degree attribute, and a colour that depends on their betweenness centrality attribute. The resulting graph should give you a good idea of which vertices are the most significant in the network.

How does this correspond to what you were told about how the network was created? Is anything surprising?

Step 2: Calculating the degree of a node

Write code to load the network as an undirected graph, and calculate the degree of each vertex (the number of neighbours it has).

Step 3: Calculating the diameter of a graph

Write code to compute the diameter of a graph (the longest shortest path in the network). You should implement a simple breadth-first search algorithm and iterate it over the nodes. Make sure to take note of visited vertices to prevent looping. Note that there is limited memory on the execution server, so you shouldn't try to store all the distances between every pair of nodes: keep only the maximum distance you find for each breadth-first search.

Starred tick

Two further networks are available as edgelist files on Moodle: `phenomenology.edges`, which is derived from this data, and `theory.edges`, which derived from this data.

1. Read the descriptions of the networks on the SNAP website (linked).
2. Investigate the properties of each network. For instance, do they each form a connected graph? If not, is there a **giant component** (see slides)? What is the mean degree of each node?
3. Are there differences between the two networks? If so, what does this suggest about collaboration in these two subareas of physics?