

## Task 11: Betweenness

This is the first part of Tick 8.

### Node betweenness centrality

The betweenness centrality of a node is based on the number of shortest paths which pass through that node. Where multiple shortest paths between a pair of nodes exist, the proportion of those paths that go via the node is considered. In general, this is a relatively expensive algorithm.

In this practical, you should implement Brandes' algorithm, which is relatively efficient. You can find the description of the algorithm (including pseudocode) here: <http://www.sciencedirect.com/science/article/pii/S0378873307000731>

You can use -1 where the pseudocode says infinity.

The original paper is: <http://www.algo.uni-konstanz.de/publications/b-fabc-01.pdf>

The pseudocode is essentially the same but the later version is a bit easier to read.

Run your implementation on the `simple_network.edges` as for Task 10. Your code should return betweenness centrality for each node.

What does the distribution of betweenness centralities look like? Can you see that this could give you good splitting points for clique finding in the next task?

### Starred tick

Try investigating the efficiency of your implementation on some of the real networks you can download from <http://snap.stanford.edu/data/> or automatically generate some random networks.

Brandes (2001) plots time against number of vertices for various networks. There's been a big improvement in computing power since that paper was written - can your implementation cope with significantly larger networks?