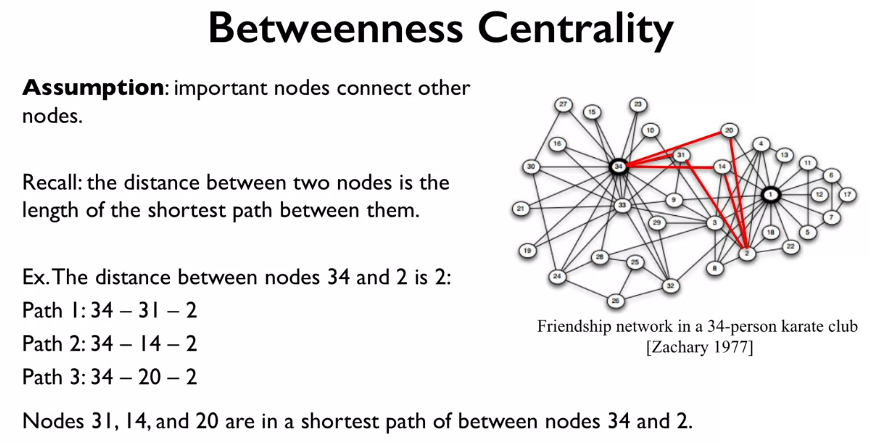
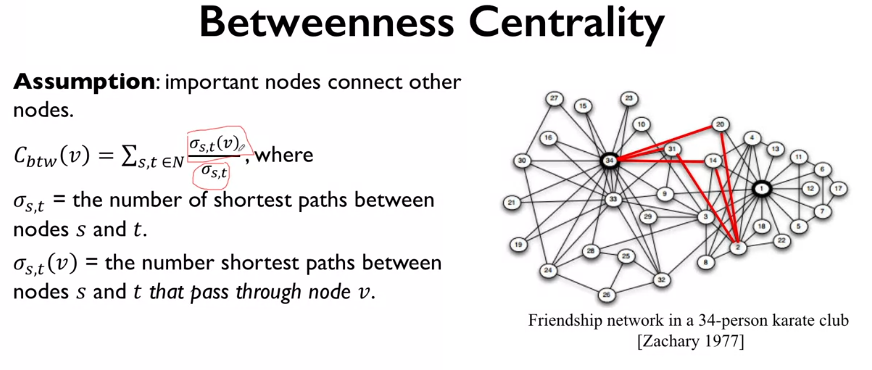
**Betweenness Centrality:**

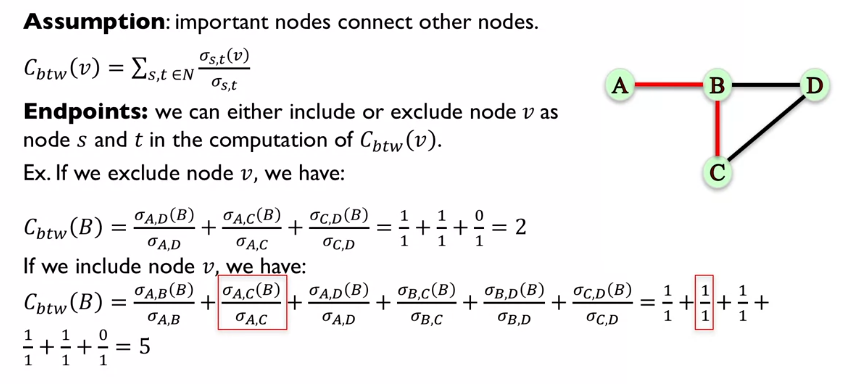
This is used to find the nodes that show up between the shortest path connection of two other nodes.

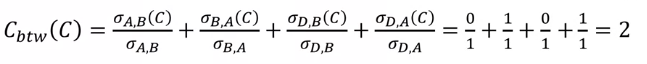
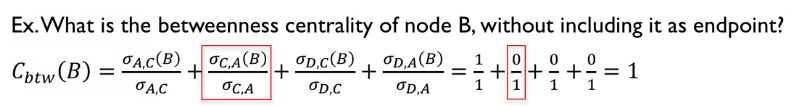
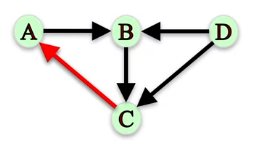
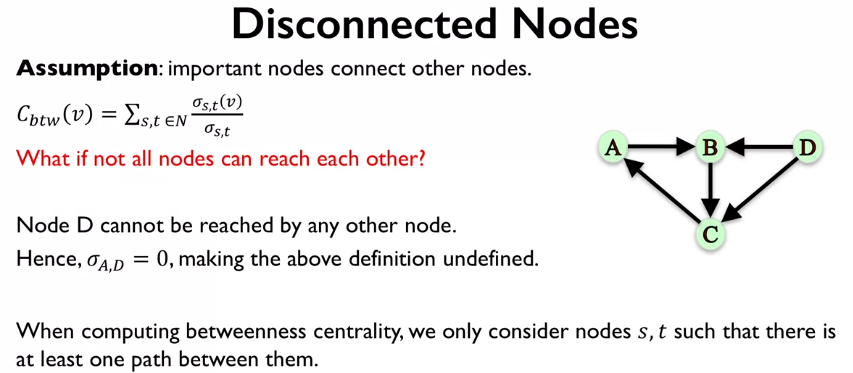




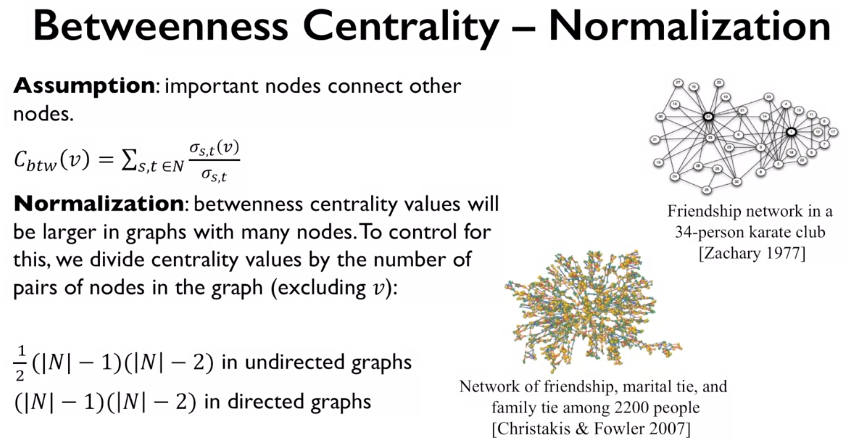
The above equation above calculates the for a node (v) the sum of the ratio of how many times node v is in the shortest path and the number of shorted paths between all nodes.

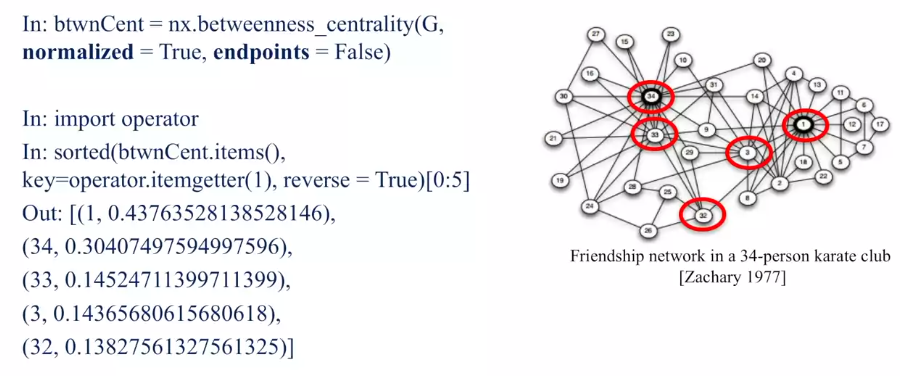
The question is, should we include node V as a possible node S,T?



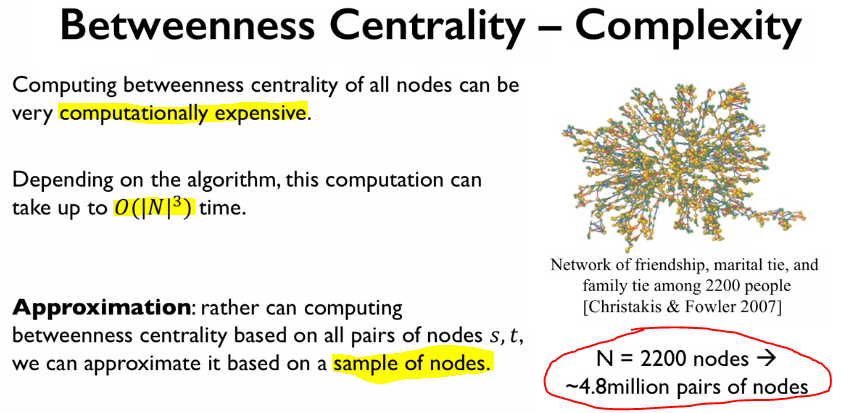


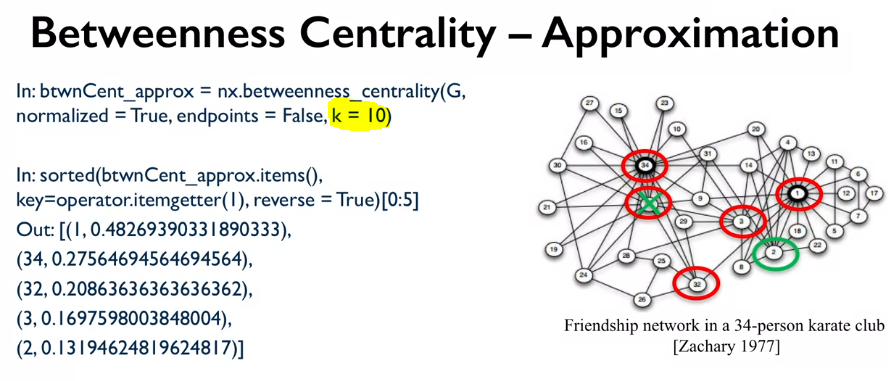
We often need to normalize the betweenness centrality in order to compare networks of different sizes, this is because, there will be many more connecting nodes in large graphs just by chance.





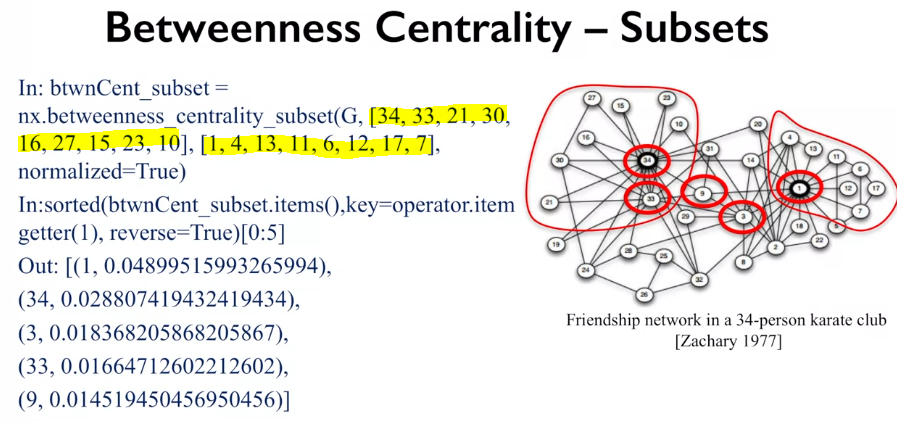
The above function (betweenness\_centrality()) is used to find the betweenness centrality in network G, we can set if we want normalization or not, and we could also set if we want to include V in our summation or not.



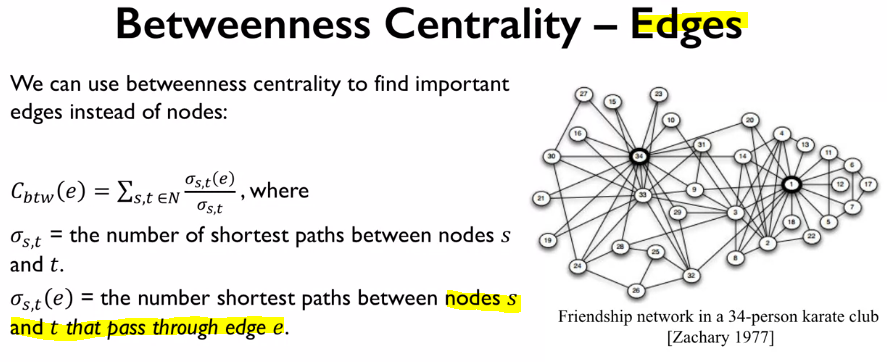


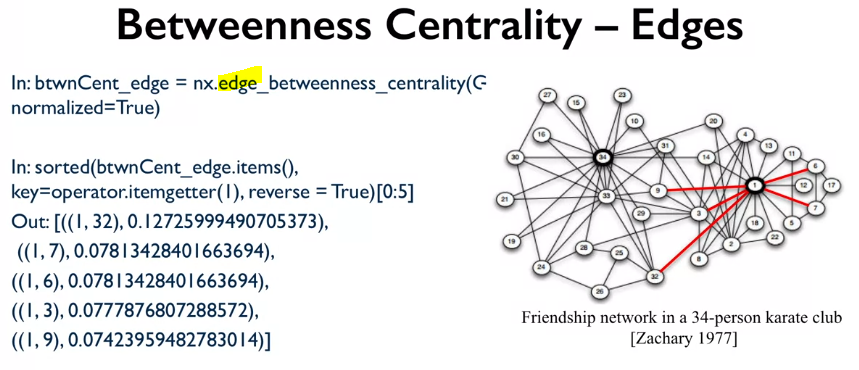
The nodes found from the approximation are very similar to the original ones found.

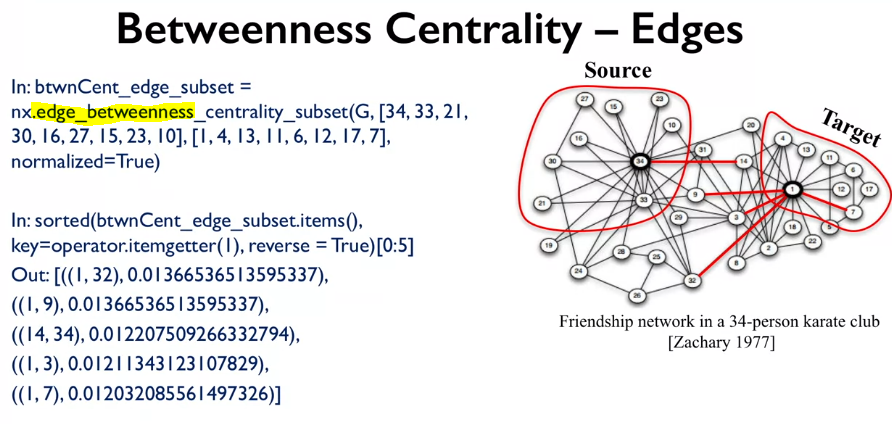
Sometimes we’re more interested in the betweenness centrality of nodes that connect subsets of a network rather that the whole thing. We can do this in NetworkX with the below code:



This is the first time that node 9 has shown up, telling us that this node is important for the connection between these two groups.







We can see that the edges that tend to connect the two groups go from a node inside a set to outside the set, and that they tent to be connected to very important nodes in the network.

