Please do your homework using an R script. Homework is NOT collected in this class. However, on the day it is due, you may be asked to share your screen and run some of your code when we discuss this homework in class. This will count towards your participation grade.

1. Use the UKfaculty data set from igraphdata, access it with data(UKfaculty).  
   1. Show the summary of the graph.
   2. Set the vertex names to the vertex indices.
   3. Plot the graph, changing the igraph\_options as needed.
   4. Note that there is a vertex attribute called group. Plot the graph with a different color for each group (do not change the graph attributes). Choose your preferred layout.
   5. Give the density for this graph.
   6. Create a new graph that coarsens this graph by group. Plot this but keep the colors for the group the same as before. You may call this graph UK.
   7. This graph does not look very good, too many edges! Simplify it.
   8. Rename the vertices with the group numbers and set their color attributes to the colors determined previously for the groups.
   9. Let the edge widths be determined by the edge weights.
   10. Let the vertex sizes be determined by the number of vertices in each group in the original graph UKfaculty.
   11. Let the edge colors be black.
   12. Plot both the original graph and the new graph side-by-side.
   13. What is the density of the new graph? What does this mean, what type of graph is this?
   14. What can you determine about the network UKfaculty from the visualization of the coarsened network UK?
2. Use the UKfaculty data set from igraphdata, access it with data(UKfaculty).  
   1. Create the undirected graph UK2 equal to the underlying undirected graph of UKfaculty. Set the vector names to the indices of the vertices.
   2. Confirm that the graph is connected. Find its density.
   3. Find the vertex connectivity and the edge connectivity of UK2. Are there any cut vertices?
   4. Find all the neighbors of vertex “1”. Is “59” a neighbor? How many vertex independent paths are there from vertex “1” to “59”? What is the shortest path between them? How many edge-independent paths?
   5. Use hierarchical clustering to partition UK2. How many subgraphs are there in the partition?
   6. Show the dendrogram.
   7. Create the subgraphs for this partition. What subgraph are “1” and “59” in? How many vertex- independent paths are there from vertex “1” to vertex “59” in their subgraph? What is the shortest path between them? How many edge-independent paths? Did vertex “1” lose any neighbors due to the partition?
   8. Find the density of all subgraphs. Are they more dense than UK2?
   9. Find a cut vertex in subgraph 3, then create a new graph called UK3 with this vertex removed. Plot UK3, giving each component a different color.
   10. Change the partitioning so that UK2 is partitioned into 4 subgraphs instead.
   11. Create a contingency table for this new partition versus the vertex attribute Group. How well did the partition do? Is it similar to the groups, or not?
   12. Use spectral clustering to partition UK2. How many subgraphs are there in the partition?
   13. Create a contingency table for the spectral partition versus the attribute Group. How well did the partition do? Is it similar to the groups, or not? Which partition seems to be better at recognizing the different groups?
   14. Create a contingency table for the original hierarchical partition (so not the one with 4 subgraphs) versus the spectral partition. How similar are they?
   15. Find the Fiedler value for UK2.
   16. Use the Fiedler vector to break the graph into two subgraphs.
   17. Create a contingency table for this partition (into 2 subgraphs) versus the vertex attribute Group. How well did the partition do?
3. Use the yeast data set from igraphdata, access it with data(yeast). Note that this is the class problem.  
   1. What does the network represent and how large is it (order and size)?
   2. Create a frequency table and a histogram of the degrees. What does this tell you?
   3. Confirm that the network is not connected. Split it into components. How many components are there? There appears to be a giant component; how big is it? Save this giant network as giant. What percentage of the proteins does the giant contain?
   4. Find the mean distance (use mean\_distance) and the diameter of the giant. What do they mean in this context? Does this seem small to you compared to the order of the graph?
   5. Find the vertex connectivity and the edge connectivity of the giant. Are there any cut vertices? If so, how many? What percentage is that of the total number of vertices (proteins) of the network? What does that mean?
   6. There appears to be a vertex attribute called Class. Show a frequency table for this attribute of the giant.
   7. Use hierarchical partitioning to partition the giant. How many subgraphs are there?
   8. Create a contingency table between membership in a subgraph according to the hierarchical partition, and the vertex attribute Class. Can you say something interesting about this table?
   9. Use spectral partitioning to partition the giant. How many subgraphs are there now?
   10. Create a contingency table between membership in a subgraph according to the spectral partition, and the vertex attribute Class. Can you say something interesting about this table?