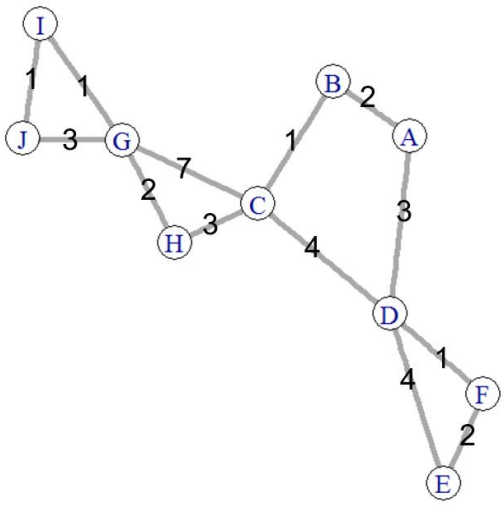
Exercises day 1

Introduction to igraph

1. Load the igraph package and make yourself familiar with it. Find out the functions to see names of vertices, edges, and how to add attributes to them.



2. For the above example of a graph:
   2.1. Recreate the graph in igraph and rename the nodes.
   2.2. Set the edge weights to the ones pictured
   2.3. Plot the graph and try out different layouts (circular, Fruchterman-Reingold, mds). Also visualize the edge weights using the *edge.width* argument in the plot function.

3. Write a function to compute the node degree of all the vertices in a graph (do not use the built-in degree function).

4. Finding the shortest path between two nodes in a graph is a common problem. It has applications ranging from biological system -studying how infectious diseases spread- to navigation to find the shortest route from one city to another. A widely used algorithm to determine the shortest path is Dijkstra. In this exercise, you have to determine the shortest path between the point E to J of the graph above. For this purpose, you will need to implement the Dijkstra algorithm.

   Change the names of the nodes to 1:10 *(V(mygraph)$name <- 1:10)*

5. (**Optional)** Create three 100-node graphs with different topologies, using the functions *erdos.renyi.game()* with an edge probability of 0.3, *barabasi.game(),* and *make_lattice()* (2-dimensional lattice).
   5.1. Calculate and plot their degree distributions.
   5.2. Just from the degree distributions, which one seems closest to what you would expect from a biological network, and why?