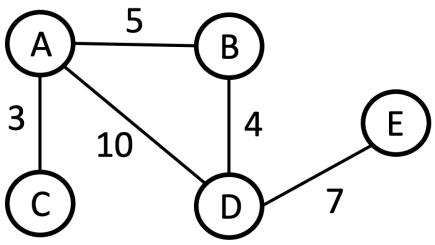
DS 2500: Intermediate Programming with Data
Prof. Rachlin and Prof. Park
Northeastern University

Weighted Graphs and Network Analysis

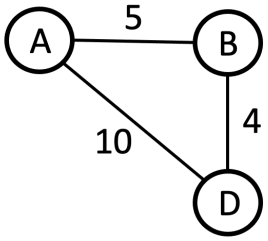
In this assignment we will use object-oriented methods to implement a weighted graph. Weighted graphs have a number (the weight) associated with each edge. The weight might represent a distance between two intersections, the cost of a flight between two cities, or the number of times two grocery items have been placed together in a shopping cart.

1. Create a new class, *WeightedGraph,* that inherits from the Graph model used in class. Now, when adding an edge, you must specify a third parameter: *the weight*! A weighted graph uses an adjacency list representation, just like a regular graph, except now the adjacent list contains tuples indicating, for each adjacent vertex, the weight of the connecting edge:



A → [(B, 5), (C, 3), (D, 10)]
B → [(A, 5), (D, 4)]
C → [(A, 3)]
D → [(A, 10), (B, 4), (E, 7)]
E → [(D, 7)]

2. Add a method called *subgraph* which takes a list of nodes and returns the subgraph of the original graph. The subgraph contains the listed vertices and any weighted edges from the original graph. For example, if we were to ask for the subgraph of the graph above consisting of the vertices A, B, and D, the method should return a new graph:



3. Read the GAD data (gad_data.csv) into your weighted graph model. GAD is the Genetic Association Database, identifying connections between genes and diseases. When publications report a "positive association" that means that one or more research studies found a connection between a gene and the occurrence or prevalence of a particular disease. When reading the data you can ignore the chromosome,

disease_class, num_negative, and num_unknown columns. The gene column and disease column will become your graph vertices. Each row represents a specific edge connecting a gene to a disease. The num_positive value represents the number of publications that have found a positive connection between the gene and the disease. This is your edge weight. You can ignore any rows (edges) where the number of positive associations is zero. (This may not actually ever occur.)

4. Add another method to the graph called *degree_distribution.* Recall that the degree of a vertex is the number of adjacent vertices, i.e, the number of edges emanating out of the vertex. To find the degree distribution, find the number of vertices with degree 0, degree 1, degree 2, and so on. Plot the number of vertices (y-axis) vs. degree (1, 2, 3, etc.) Try scaling both the x and y axis using *log* scaling. If your resultant degree distribution is a straight line when plotted on a *log-log* scale, this indicates a *scale-free* network. Many interesting graphs or networks are scale-free, particularly in biology and social networks. When a network is scale free it means that many vertices have very few connections while a few vertices have many connections. Is the gene-disease association network scale free?

5. Perform the following analysis.
    a. Start with the node called “asthma”
    b. Find all adjacent genes – i.e., all genes associated with asthma.
    c. For all of these asthma-linked genes, find all the other diseases that asthma-linked genes are connected to, including of course asthma!
    d. Find the subgraph associated with the asthma-linked genes found in step (b) and the diseases you identified in step (c).
    e. Use the networkx library to generate a visualization of this subgraph. Can you identify what other disease is strongly connected to genes linked to asthma?

What to submit:

Submit your code, plot of your degree distribution, and GAD subgraph visualization for grading.