1. Download HW4.zip and then implement the following functions in Matlab:
   
   (a) \( \text{UG} = \text{moralize(DAG)} \)
   
   (b) \( \text{order} = \text{elimOrderGreedy(UG, sizes)} \). The algorithm should try to eliminate the first simplified node (one that is connected to all its uneliminated neighbors, so that no fill-in edges are necessary); if there are no such nodes, it should eliminate the node that results in an induced clique of minimal weight, where the weight of a clique is the product of the sizes of the nodes it contains:
   
   \[
   w(C) = \prod_{i \in C} s(i)
   \]
   
   where \( s(i) = \text{sizes}(i) \) is the number of values node \( i \) can take on.
   
   (c) \( [\text{GT, cliques, fillIns}] = \text{triangulate(UG, order)} \) that triangulates an undirected graph with the specified order. This returns the triangulated version, the maximal cliques, and the fill-in edges.
   
   (d) \( \text{J} = \text{jtreeFromMaxCliques(cliques)} \) that builds a junction tree from the maximal cliques of a chordal graph. You may use the provided function \text{minSpanTree}.

2. Consider the undirected graph below. Suppose all nodes have size 2, except for the following: \( D = 4, E = 5, F = 6, G = 7 \).

   (a) What elimination ordering does your function \text{elimOrderGreedy} produce in this case? What is the sum of the weights of the cliques?
   
   (b) Construct a better elimination ordering; what is the sum of the weights of the cliques in this case?
   
   You might find the function \text{hw4-q1.m} helpful.

3. Construct an optimal junction tree (i.e., one which minimizes the sum of the clique weights) for the Bayes net below. Assume all nodes have the same size (weight). Note: The answer may not be unique. You can verify correctness using \text{jtreePropertyCheck(jtree, cliques)}. How can you be sure your answer is optimal in this case?