COMP 4030/6030: Assignment 6
Due date: 04/18/2016

Email programming solutions to the TA (Quang Tran, qmtran@memphis.edu). Put "COMP 4030/6030 assignment 6" on the subject line.

A graph G consists of a set of vertices $\{v_0, \cdots, v_{n-1}\}$ and a set of edges $\{e_1, \cdots, e_{m-1}\}$. An edge connects two vertices v_i and v_j. You should test your programs using this graph module: https://github.com/vtphan/Graph. Download the file graph.py and write small Python examples to use it.

The solutions to these problems should be based on the backtracking technique.

1. (50 points) Given a weighted graph, the goal is to place cameras at vertices to "watch" the edges. If a camera is placed at vertex v, it can watch all the edges that are connected to vertex v. Write a Python program using backtracking to determine the minimal amount of cameras that can be placed to watch all edges for each given graph.

2. (50 points) The goal of this problem is to identify the largest clique in a social network. A social network can be represented as a graph, G. A clique is a set of vertices $\{v_1, \cdots, v_k\}$ such that for any pair of vertices a and b in this set, (a, b) is an edge. The size of a clique is the number of vertices in the set. Write a Python program that finds a maximum clique.

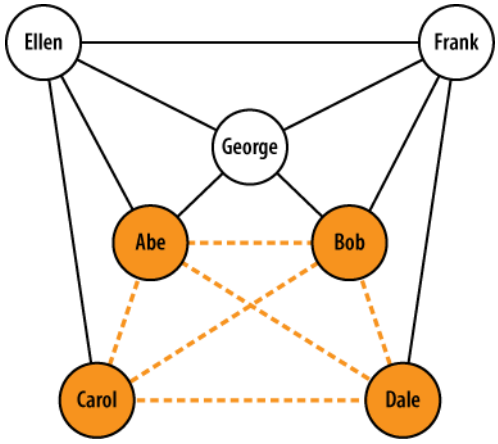

Figure 1: There are cliques with 1, 2, 3, and 4 vertices. $\{Ellen, Frank, Bob\}$ is not a clique because there is no edge between Ellen and Bob. $\{Ellen, Abe, George\}$ is a clique with 3 vertices. $\{Abe, Bob, Carol, Dale\}$ is a maximum clique with 4 vertices. There can be more than one maximum cliques.

3. (50 points) Write a Python program that prints all cliques with at least k vertices. k is a user-defined variable.