COMP 4030/6030 Spring 2018. Assignment 8
Due: April 24, 2018 (before class starts)

Plagiarism Policy:
You can discuss how to solve the problems with your classmates, but the solution must be your own. Using other people's solution will result in a zero for the assignment and possible additional penalties.

Submission:
Put your name as part of the file name and upload your submission to eCourseware Dropbox.

To do this assignment, you will need the following files:

- graph.py – a module to generate random graphs
- hw8_helper.py – an example how to use graph.py to generate graphs to this assignment.
- rg_10_15_2018.svg -- a graph generated by hw8_helpher.py. It can be viewed by a browser.
- rg_10_20_2018.svg -- a graph generated by hw8_helpher.py. It can be viewed by a browser.
- rg_10_22_2018.svg -- a graph generated by hw8_helpher.py. It can be viewed by a browser.

Note that these 3 graphs serve as examples for you to test your programs. Your programs should work for any graph with any number of nodes, edges and different parameters.

Imagine a graph representing a network of stations (nodes) connecting by roads (edges). If a camera is placed at a node, it can "watch" all roads connected to it. For example, in the network rg_10_15_2018, a camera placed at station 8 can watch the roads that connect station 8 to stations 3, 4, and 0. In this network, cameras placed at stations 3, 4, 0 and 1 can watch all roads. Therefore, {3, 4, 0, 1} is a camera placement on this network that covers all roads.

1. (30 points) Write a Python program, using the backtracking technique we learned, to generate all possible camera placements on a network. Test your program on the 3 networks above.

2. (30 points) We are limited with resources and have budget for only 6 cameras. Write a Python program, using the backtracking technique we learned, to generate all possible camera placements on a network that use only 6 or fewer cameras. Test your program on the 3 networks above.

3. (30 points) Write a Python program, using the backtracking technique we learned, to find the minimum number of cameras that can be placed to watch all roads of a given network. Print out the placement that gives the minimum number of cameras. Test your program on the 3 networks above.

4. (10 points) Again we have a budget of k cameras. For example, k=6. We want the camera to cover as many roads as possible. It is possible we won’t be able to cover all roads. Write a Python program, using the backtracking technique we learned, to find the camera placements that cover as many roads as possible.