INEG 56103: Introduction to Optimization Theory
Computational Assignment 2

Instructions:

1. Create a new project folder titled “ca-02-lastname” where you will include all of your code and documentation for this assignment. Create a single .ipynb file in which you will include all of your code. Run the .ipynb file so all output is visible after opening the notebook. The code should clearly present the results for each problem, and it should contain (at a minimum) markdown to clearly identify your name and organize the notebook by problem. Create a document titled “ca-02-lastname-results” in which you summarize output from your code.

2. Given an undirected network $G = (N, E)$ with node weights $w_i, i \in N$, use Pyomo to implement an integer programming model that finds a maximum-weight subset of nodes such that no pair of nodes in the subset is connected by an edge. Your code should take as input data files of the same format as given in graph10_nodes.csv and graph10_edges.csv. Solve the “graph10” instance using CPLEX and output the optimal solution and objective value. Add a section “Problem 2” to your document and include this output in your document.

3. Use CPLEX to solve the “graph_ws1000” and “graph_er5000” instances, each with a time limit of 30 minutes. Output the upper bound, lower bound, absolute optimality gap, and relative optimality gap for each instance. Add a section “CPLEX Output” to your document and include this output in the section. Briefly explain the meaning of each of the four values in the context of one of the two instances.

4. Implement the following greedy heuristic, where $\deg_G(i)$ is the number of neighbors of node i in graph G.

Greedy Heuristic

> **Set** $A = \emptyset$; # Nodes will be added to this list if they are included in the subset
> **Set** $G' = G$; # Create a copy of G that will keep track of components not connected to A
>
> **While** G' contains at least one node:
> Find a node j that maximizes $w_i/(\deg_{G'}(i) + 1)$ over nodes i of G'
> Add node j to A
> Remove node j and all of its neighbors from G'
>
> **Return** A and its objective function value;

5. For the “graph_ws1000” and “graph_er5000” instances, run your greedy heuristic and output the solution and objective value. Add these values to a section “Heuristic Output” in your document.

6. Add a section “Reflection” to your document. In this section, briefly compare the heuristic results to that of the optimization solver and briefly reflect on the relative advantages and disadvantages of each approach. It should be possible to do this in no more than a few sentences.

7. (Optional for bonus credit) Can you come up with an instance for which the greedy heuristic performs very poorly? More credit will be awarded for instances with smaller values of (heuristic objective value)/(optimal objective value) and smaller networks.

8. Add a pdf of your document to the main project folder. Zip the folder and submit it to blackboard.