STOR 415, Fall 2019
Homework Assignment No. 10

For each problem that requires Jupyter-GAMS coding:

- Create an ipynb file with exactly the same name as required in the problem. In the GAMS code, declare variables with names given in the problem. Then, in the last cell of your notebook, write the following codes to display values of all variables (replace "var1", "var2" and "var3" with names of variables in the problem):

  %gams display var1.l, var2.l, var3.l;
  %gams_lst -e

- Download your ipynb file(s) to your local computer, and then submit them on Sakai as attachments to this assignment.

---

1. Three courses, which we call A, B, and C, need final examination rooms. Three rooms, X, Y, and Z, are available. Any course can be assigned to any room, but since the exams take place at the same time only one course can be put in each room. The value of placing each course in each room is given in the following table, and the total value is the sum of the values of the three placements that will be made.

Course	Room X	Room Y	Room Z
A	10	2	-8
B	4	6	9
C	5	-2	10

   (a) **Non-coding.** Formulate an minimum-cost-network-flow-problem (MCNFP) to maximize the total value; write down the mathematical formulation including the definition of variables, the objective function and all constraints. Then draw a graph to include all information about the MCNFP. Which algorithm should be used to solve the MCNFP? Why?

   (b) **Coding.** Create a Jupyter notebook named *classroom.ipynb* to solve the problem formulated in part (a). Display values of all variables. (The optimal value is 26.)

2. The data given in the following table are distances of arcs between pairs of nodes in a network. An "X" indicates that there is not an arc between that pair. (For example, the arc from node 1 to node 2 has length 10, and there is not an arc from node 2 to node 1.)

	To					
From	1	2	3	4	5	6
1	X	10	3	1	X	X
2	X	X	1	1	2	X
3	X	X	X	1	2	4
4	1	X	X	X	2	9
5	1	6	X	X	X	6
6	2	3	4	X	X	X

(a) **Non-coding.** Formulate a single MCNFP whose solution can be used to find the shortest paths from node 1 to each of the other nodes. (The requirement here is to solve a single MCNFP to find the shortest paths from node 1 to all other nodes. Thus, you should not solve more than one MCNFP.)

(b) **Coding.** Create a Jupyter notebook *shortestpathtree.ipynb* to solve the problem. Display values of all variables.

(c) **Non-coding.** Using the optimal solution of that MCNFP, provide a shortest path from node 1 to node i, for each $i = 2, \cdots, 6$.