DSO-570 Final Exam (Fall 2021)

Version 7

Instructions:

First ensure that the version number is the same as the one emailed to you prior to the exam.

Begin only after the instructor says to begin. Once the 100 minutes is up, make sure to save, and then immediately email the completed Jupyter notebook to the professor by replying to the email by which you received the exam. Afterward, open your email attachment and double check that you sent the right file. If you want to attach photos of handwritten notes, do that separately by replying to the email again, so that you do not risk delaying your submission of the Jupyter notebook. **The Jupyter notebook must be received within 3 minutes after time is up, and all photos must be received within 10 minutes after time is up.**

You can consult any textbooks, notes, or online material. However, you **CANNOT:**

- ask anyone other than the professor for help.
- share your exam or any part of your work with anyone else other than the professor.
- communicate with another student during the exam in any way, including talking, calling, texting, emailing, messaging, passing notes, or any other communication.

Please do not violate academic integrity! A violation will result in a failing grade and leave an inerasable mark on your record. Employers will not hire a cheater. It's not worth it!

Please type your name below to certify that you have adhered to all university policies regarding ethical behavior in preparing for and completing this exam.

Name:

Q1. Exercise Selection in Course Design (Concrete Formulation; 10 points)

Professor Shi would like to select an optimal set of exercises for his course, so that students learn a lot from doing the exercises and the level of difficulty builds up gradually. He has in mind a fixed set of exercises to select from. For each exercise, he has assigned a difficulty score out of 10, and a learning score out of 10. A higher difficulty score means that the exercise is more difficult, and a higher learning score means that it has higher potential to teach the students useful skills for the real world. As an example, the following table lists a set of 11 exercises to select from, along with their assigned scores.

Exercise #	Difficulty Score	Learning Score
0	0	-
1	1	2
2	2	3
3	3	4
4	3	6
5	4	7
6	5	8
7	6	7
8	7	8
9	9	9
10	10	-

In the above, Exercise 0 is the easiest and corresponds to an introductory exercise that sets the baseline for the course, and Exercise 10 is the hardest and represents the final exam. Both of these must always be included in the course, so their learning scores are not relevant for the optimization.

Write a concrete formulation of a linear optimization model to decide which of the exercises 1 through 9 to select in order to maximize the total learning score of the selected exercises, subject to the following constraints:

1. Among Exercises 1 through 9, at most four can be selected due to time limits.

2. In order to ensure that there is no big jump in difficulty between two consecutive exercises that are selected, the maximum gap in difficulty score between two consecutive selected exercises is at most 3.

- For example, if the set of all selected exercises is $\{0, 1, 5, 7, 9, 10\}$, then this would satisfy this constraint, as the gap in difficulty score between the pair of consecutive exercises $(0, 1)$ is 1, between the pair $(1, 5)$ is 3, between the pair $(5, 7)$ is 2, between the pair $(7, 9)$ is 3, and between the pair $(9, 10)$ is 1.
- However, if the set of all selected exercises is $\{0, 1, 6, 7, 9, 10\}$, then this constraint would be violated as the difficulty gap between the pair $(1, 6)$ is 4. Similarly, $\{0, 1, 3, 5, 7, 10\}$ would violate this constraint, as the difficulty gap between the pair $(7, 10)$ is 4.

3. Due to overlapping content, certain pairs of exercises should not be both selected. In this example, Exercises 2 and 8 should not be both selected. Moreover, Exercises 4 and 9 should not be both selected.

Your constraints must allow for every feasible selection of exercises according to the above rules, and must disallow every infeasible selection. In other words, do not base the logic of your constraints on the numerical values of the learning scores given above.

Decision Variables:

Objective:

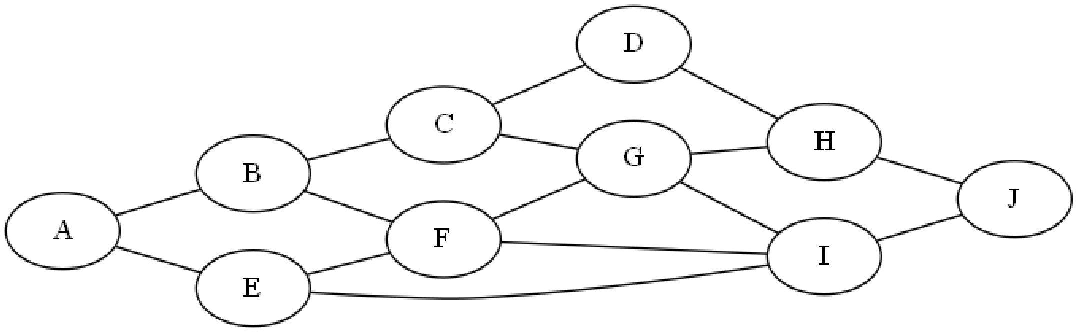
Constraints:

Q2. Campus Security under Limited Resources (Abstract Formulation; 9 points)

USC is on holiday break and the Department of Public Safety (DPS) has a limited number of staff. Nevertheless, DPS would still like to use its resources optimally to protect the area around campus. To do so, DPS can station staff at intersections:

- An intersection is said to be **protected** if either it is staffed or if at least one of its neighboring intersections is staffed.
- An intersection is said to be **well-protected** if either it is staffed or if at least two of its neighboring intersections is staffed.
- The **total protection score** is defined to be the total number of protected intersections plus the total number of well-protected intersections.

For example, in the following map, suppose that intersections A and F are both staffed, and no other intersection is staffed. Then the set of protected intersections is $\{A, B, E, F, G, I\}$. The set of well-protected intersections is $\{A, B, E, F\}$. The total protection score is 6+4=10. (By definition, every well-protected intersection is also protected, so a well-protected intersection effectively has a weight of 2 in the total protection score, whereas a protected intersection that is not well-protected has a weight of 1.)



Write an abstract formulation of a linear optimization model to maximize the total protection score subject to staffing at most k intersections. Here, k is an input data variable and denotes the maximum number of intersections that can be staffed. Your formulation must be correct not only for the above example, but also for an arbitrary number of intersections and an arbitrary map describing which pairs of intersections are neighbors. You are free to define any data variables that can be straightforwardly obtained from the map.

Data:

Decision Variables:

Objective:

Constraints:

Q3. Reusable Software for Exercise 11.3 (Gurobi Code; 9 points)

This problem asks you to implement the following abstract formulation for Exercise 11.3 (Assigning Consultants to Projects). The original problem description can be found in the handout for Week 11 Session 22, but you don't necessarily need it to solve this question.

Data:

- I: the set of consultants.
- J: the set of projects.
- K: the set of skills.
- a_{ik}: whether consultant i possesses skill k. (Binary)
- r_{jk}: the number of consultants of skill k needed for project j.
- S: the set of project pairs (j_1, j_2) that conflict with one another.
- c_{ij}: travel cost of consultant i to project j.

Decision Variables:

- x_{ij}: whether to assign consultant i to project j. (Binary)

Objective and Constraints:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{i \in I, j \in J} c_{ij} x_{ij} \\
\text{subject to:} \quad & \sum_{i \in I} a_{ik} x_{ij} \geq r_{jk} && \text{for each project } j \in J \text{ and each skill } k \in K. \\
& x_{ij_1} + x_{ij_2} \leq 1 && \text{for each consultant } i \in I \text{, and each pair } (j_1, j_2) \in S.
\end{aligned}$$

Write a function called `assignConsultants` with one input argument:

- **inputFile**: An Excel file containing all of the input data needed for the formulation. The format is exactly as in the `Q3-input1.xlsx` and `Q3-input2.xlsx` files attached to the exam. The `Q3-input1.xlsx` file corresponds to the sample data in the original description of Exercise 11.3 in Week 11 Session 22.

Your function should print for each consultant his or her assigned projects, as well as the optimal objective value. The output format must be exactly as in the sample outputs below. (It is okay however if your code returns a different optimal solution with the same objective value.)

All of your final code should be in the following cell, so that if one restarts the kernel and only runs this cell, the test code will run

```
[1]: # Final Code (Make sure to include all necessary imports here)
```

```
[2]: # Test Code 1
     assignConsultants('Q3-input1.xlsx')
```

```
Alice:  P2
Bob:  P1  P3
Charlie:  P2
Daphne:  P1  P3
Minimum total cost: 36.0
```

```
[3]: # Test Code 2
     assignConsultants('Q3-input2.xlsx')
```

```
Amy:  Project3  Project4
Brian:  Project5
Catherine:  Project1  Project3
George:  Project3  Project4
Frank:  Project5
Stewart:  Project2  Project5
Su:  Project4
Xavier:  Project1  Project5
Abdul:  Project2  Project5
Lucy:  Project2  Project3
Minimum total cost: 115.0
```