ISyE524: Introduction to Optimization
Problem Set #5
Due: *Tuesday* April 28, at 11:59PM

Notes and Deliverables:

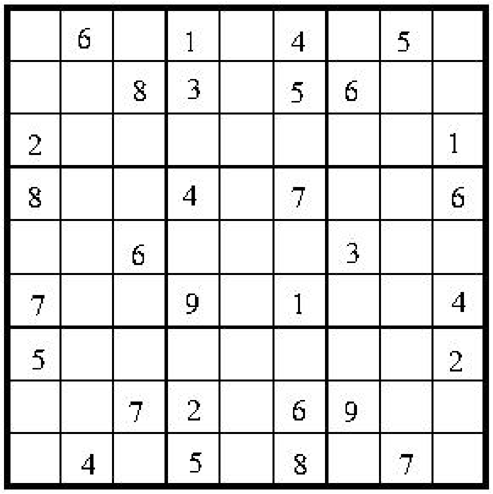
- Submit solutions to all problems in the form of a single PDF file of the IJulia notebook to the course dropbox. (If you do not submit as a single PDF you will lose points.)
- Be sure that your answers (including any pictures you include in the notebook) are in the correct order. (If you do not submit the problems in order, you will lose points.)

1 Ahhhhhhh, Daniel-son. You SuDoku Master.

Sudoku is a puzzle game craze that once upon a time was very popular. is sweeping the world. According to a popular sudoku web site (`https://www.learn-sudoku.com/sudoku-requires-no-math.html` "Sudoku requries no math to solve." But we will show them that we can needlessly and endlessly complicate *all* things with math, even a fun puzzle like sudoku.

In sudoku, you are given an $n \times n$ grid with some of the entries of the grid being filled in with numbers from $\{1, 2, \ldots n\}$. Typically $n = 9$, and in every sudoku puzzle $\sqrt{n}$ is an integer. The object of the game is to fill in the grid so that every row, every column, and every $\sqrt{n} \times \sqrt{n}$ "box" contains the digits 1 through n. Figure 1 is a sample sudoku.

Figure 1: Sample Sudoku



	6		1		4		5	
		8	3		5	6		
2								1
8			4		7			6
		6				3		
7			9		1			4
5								2
		7	2		6	9		
	4		5		8		7	

You may (and probably should) use the class example notebook `Sudoku.ipynb` to solve these problems.

Let me help you with this problem by fixing some notation.

- Rows/Columns/Digits of the grid: $[n] = \{1, 2, \ldots, n\}$, we have $n = 9$ in standard sudoku, but you can make bigger instances.

- Clusters $\mathcal{C}$, which is a set of sets. Each set contains the grid positions (pairs) in that cluster.
- "Fixed" positions given as sets $\{F_1, F_2, \ldots F_n\}$ such that

$$F_k = \{(i,j) \mid \text{ the grid position } (i,j) \text{ is the digit } k \text{ in the given sudoku instance } \}$$

In the sudoku in Figure 1, we could have the math notation that

$$\mathcal{C} = \Big\{\{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)\},$$
$$\{(1,4),(1,5),(1,6),(2,4),(2,5),(2,6),(3,4),(3,5),(3,6)\},$$
$$\ldots,\{(4,4),(4,5),(4,6),(5,4),(5,5),(5,6),(6,4),(6,5),(6,6)\},\ldots\Big\}$$

where there are 9 different sets containing the pairs of grid elements.

$$F_2 = \{(3,1),(7,9),(8,4)\}$$
$$F_4 = \{(1,6),(4,4),(6,9),(9,2)\}$$

1-1 Problem Write an integer program that will demonstrate how to complete a given sudoku instance of size n, with clusters given by $\mathcal{C}$, and fixed positions $F_1, F_2, \ldots F_n$.

1-2 Problem Solve the model you write in Problem 1-1 in Julia and JuMP. Be sure to print out your solution.

In real sudoku games, there is only *one* feasible solution to the IP formulation you built in Problem 1-1. (That is why it shouldn't matter what your objective function is). In this problem, we make the game more challenging. We will call my new game *Jeff-Doku.* In Jeff-Doku, we the set of main "diagonal" grid locations as

$$D = \{(1,1),(2,2),\ldots(n,n)\}$$

and we maximize the sum of the elements placed on the diagonal. But we need to "relax" the restriction that we have to match *all* of the given squares.

1-3 Problem Suppose now, that you need not fix *all* of the values given in the instance, but rather you need fix only at least $K = 24$ of them. Modify your instance from Problem 1-1 to solve this more challenging version of the game that maximizes the sum of the elements on the main diagonal. Even if you can solve sudoku by hand, I'll bet you *can't* solve Jeff-doku by hand!

1-4 Problem Create your model from Problem 1-3 in Julia and JuMP and solve the Jeff-doku instance from Figure 1, where you need fix only at least $K = 24$ of the given values.

1-5 Problem Even that problem was "too easy". Now add the additional constraints and variables necessary to enforce the constraint that if you put 2 or more "9"s on the main diagonal in your Jeff-doku solution, then you must also put *exactly* 3 "5"'s on the diagonal. You need not solve this model. Just write the new constraints that you need when compared to your answer to Problem 1-3.

2 Integer Programming is Smarter than a Fourth Grader?

This was my son's homework assignment when he was in 4th grade. I tried to make him use integer programming to figure out the answer. He told me to buzz off, but since you hopefully have more respect for your professor than my son does for me, you will be *happy* to use integer programming to solve the following mind-bender:

Maxine, Mabel, Mavis, Millie, and Martha were grandmothers. Their daughters were named Carla, Carol, Cindy, Cathy, and Caren, and the daugthers were married to John, Jake, Jack, Joe, and Jason. Each of the couples had one son. The names of the grandsons were Tom, Tex, Tim, Tip, and Tab.

You also know the following facts/clues:

1. Maxine's Daughter was not Carla
2. Mavis' son-in-law was not named Jack, and Catchy was married to Joe, but their son was not named Tab
3. Tim's mother was not named either Carol or Carla, because Tim's mother was Jake's wife Cindy
4. Mabel, Millie, and Martha did not have daughters named Carla or Carol, and Martha's son-in-law was not John because John was married to Caren, and their son was named Tom
5. Millie was terrible with names, but she knew that her son-in-law was named either Joe or Jason, and her grandson was named either Tip or Tab.
6. Tab did not have a grandmother named Mavis.

Who was Maxine's grandson? In fact, we can determine everyone's grandson. This is a bit hard. Here are some hints.

- You will need at least the following binary variables in your model
  - $\mathsf{xD}_{g,d}$ if Grandma $g \in G$ has Daughter $d \in D$
  - $\mathsf{xH}_{g,h}$ if Grandma $g \in G$ has Son-In-Law $h \in H$. (That is, the daughter of Grandma g has husband h)
  - $\mathsf{xS}_{g,s}$ if Grandma $g \in G$ has Grandson $s \in S$
- Write the constraints that every grandmother has one daughter, and every daughter has one grandmother
- Same for son-in-law and grandson.
- Then implement each of the rules above. (Some of the bullets will require multiple constraints).
- **CHECK TO SEE IF YOU HAVE AN ANSWER AFTER IMPLEMENTING EACH CONSTRAINT**. The instance is not infeasible, so if you are getting infeasible, you have implemented a constraint incorrectly.

2-1 Problem Write a math model whose solution will determine for each grandmother, who her daughter, son-in-law, and grandson is.

2-2 Problem Implement your model from 2-1 and print out each grandmothers, daughter, son-in-law, and grandson. Here is some helpful code.

Julia Code

```julia
G = [:Maxine, :Mabel, :Mavis, :Millie, :Martha]
D = [:Carla, :Carol, :Cindy, :Cathy, :Caren ]
H = [:John, :Jake, :Jack, :Joe, :Jason ]
S = [:Tom, :Tex, :Tim, :Tip, :Tab ]

@variable(m, xD[G,D], Bin) # 1 iff grandma g has daughter d
@variable(m, xH[G,H], Bin) # 1 iff grandma g has (son-in-law) (daughter’s husband) h
@variable(m, xS[G,S], Bin) # 1 iff grandma g has grandson s


function printGrandmaSolution(xD, xH, xS)
    for g in G
        grandma = g
        daughter = :Unknown
        son_in_law = :Unknown
        grandson = :Unknown

        for d in D
            if value(xD[g,d]) > 0.5
                daughter = d
            end
        end
        for h in H
            if value(xH[g,h]) > 0.5
                son_in_law = h
            end
        end
        for s in S
            if value(xS[g,s]) > 0.5
                grandson = s
            end
        end
        println("Grandma ", g, " has daughter ", daughter, " son-in-law ", son_in_law,
        " and grandson ", grandson)
    end
end;
```

3 An Auror is Near

He-Who-Shall-Not-Be Named is back, and all wizards need protection by aurors. The wizarding world has been divided into eight districts. The time (in seconds) required to travel from one district to another via the floo network is shown in the table below:

	1	2	3	4	5	6	7	8
1		3	4	6	1	9	8	10
2	3		5	4	8	6	1	9
3	4	5		2	2	3	5	7
4	6	4	2		3	2	5	4
5	1	8	2	3		2	2	4
6	9	6	3	2	2		3	2
7	8	1	5	5	2	3		2
8	10	9	7	4	4	2	2	

The wizard population in district i (in thousands) is given by p_i for $i = 1, \ldots, 8$. In this instance, $p = (40, 30, 35, 20, 15, 50, 45, 60)$, so the population of district 1 is 40 thousand wizards, etc. The Ministry of Magic has declared that aurors will be located at 3 locations.

3-1 Problem Write an integer programming model that will determine the locations of the aurors that maximize the number of people who live within two seconds of an auror.

Hint: Your solution will need two sets of binary decision variables: one set to determine whether or not each district is assigned as an auror location, and the other set to determine for each district if it has an auror assigned to a district within 2 seconds.

3-2 Problem Implement your model from Problem 3-1 in Julia and JuMP to determine the best three locations for aurors.

4 Routing Apple Deliveries

A local apple orchard needs to deliver orders for its apples to six customers. The time (in minutes) between the orchard (O) and each customer, and also between each pair of customers, is given in the table below. The table also gives the size of each customer's order (in crates). (The time between locations is the same in both directions, so for each pair of locations the time is only reported once.)

	Distance						
	O	C1	C2	C3	C4	C5	C6
O	-	18	9	11	14	21	12
C1	-	-	9	16	4	5	10
C2	-	-	-	9	6	13	4
C3	-	-	-	-	14	21	22
C4	-	-	-	-	-	7	9
C5	-	-	-	-	-	-	12
		C1	C2	C3	C4	C5	C6
Order size (crates)		60	55	30	45	25	50

The orchard has one truck that can carry at most 100 crates of apples. (Therefore, the truck will have to make multiple trips to make all the deliveries) Each trip can visit more than one customer, as long as the *total* number of crates on the truck at the start of the trip does not exceed the capacity. However, each order must be delivered all at once (i.e., a customer's order should not be split into two trips.)

4-1 Problem Formulate an integer program to help the orchard assign customers to the trips so that the total time the truck has to travel per day is minimized. (Hint: model the problem as a set covering problem.)

4-2 Problem Implement and solve the model using the JuMP package in Julia, and explain the solution in terms of the problem.

5 Paint Production

As part of its weekly production, a paint company produces five batches of paints, always the same, for some big clients who have a stable demand. Every paint batch is produced in a single production process, all in the same blender that needs to be cleaned between each batch. The durations of blending paint batches 1 to 5 are 40, 35, 45, 32 and 50 minutes respectively. The cleaning times depend of the colors and the paint types. For example, a long cleaning period is required if an oil-based paint is produced after a water-based paint, or to produce white paint after a dark color. The times are given in minutes in the following matrix A where A_{ij} denotes the cleaning time after batch i if it is followed by batch j.

$$A = \begin{bmatrix} 0 & 11 & 7 & 13 & 11 \\ 5 & 0 & 13 & 15 & 15 \\ 13 & 15 & 0 & 23 & 11 \\ 9 & 13 & 5 & 0 & 3 \\ 3 & 7 & 7 & 7 & 0 \end{bmatrix}$$

Since the company has other activities, it wishes to deal with this weekly production in the shortest possible time (blending and cleaning).

5-1 Problem Write an integer programming model to minimize the time for weekly production and What is the corresponding order of paint batches? The order will be applied every week, so the cleaning time between the last batch of one week and the first of the following week needs to be accounted for in the total duration of cleaning.

5-2 Problem Implement your model from Problem 5-1 in Julia and JuMP to determine the minimum time and order of production