

**HOMEWORK 5**  
**Due: November 15, 11:59 PM**

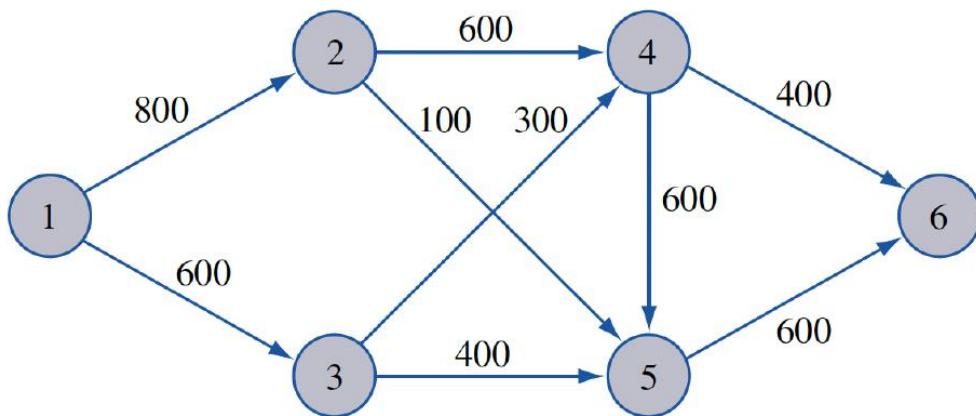
---

Homework assignments must be prepared individually and submitted online through Gradescope. See the syllabus for more information on Homework requirements and expectations. Any necessary modifications to this assignment will be posted to Brightspace as an announcement.

**NOTE: Please show your final solution clearly and include your all Excel settings such as model display and solver settings along with the final solution in your Gradescope submission. Do the same for Gurobi solutions.**

**(25 pts) Question 1:** Each hour, an average of 900 cars enter the network in the figure below at node 1 and seek to travel to node 6. The time it takes a car to traverse each arc is shown in the following table. In the figure, the number above each arc is the maximum number of cars that can pass by any point on the arc during a one-hour period.

- Formulate this problem as a minimum cost flow problem that minimizes the total time required for all cars to travel from node 1 to node 6. Clearly define in words what your decision variables represent. In the LP, label each constraint clearly for what the constraint corresponds to.
- Solve the Question using Excel Solver. Submit a screenshot of the result in Excel spreadsheet with Solver setting and Excel spreadsheet display.



Arc	Time (Minutes)
(1, 2)	10
(1, 3)	50
(2, 5)	70
(2, 4)	30
(5, 6)	30
(4, 5)	30
(4, 6)	60
(3, 5)	60
(3, 4)	10

$$a) \text{ Minimise } Z : \quad Z = 10x_{12} + 50x_{13} + 70x_{15} + 30x_{16} + 30x_{45} + 60x_{46} + 60x_{56} + 10x_{61}$$

st:

$$\begin{aligned} x_{12} + x_{13} &= 900 \quad \text{node 1} \\ x_{11} + x_{21} - x_{12} &= 0 \quad \text{node 2} \\ x_{31} + x_{35} - x_{13} &= 0 \quad \text{node 3} \\ x_{45} + x_{46} - x_{35} &= 0 \quad \text{node 4} \\ x_{56} - x_{45} - x_{46} &= 0 \quad \text{node 5} \\ x_{46} + x_{56} &= 100 \quad \text{node 6} \\ x_{12} \leq 800 &\quad \text{arc}(1 \rightarrow 2) \\ x_{12} \leq 600 &\quad \text{arc}(1 \rightarrow 3) \\ x_{15} \leq 600 &\quad \text{arc}(2 \rightarrow 4) \\ x_{21} \leq 180 &\quad \text{arc}(2 \rightarrow 5) \\ x_{31} \leq 300 &\quad \text{arc}(3 \rightarrow 4) \\ x_{35} \leq 400 &\quad \text{arc}(3 \rightarrow 5) \\ x_{45} \leq 600 &\quad \text{arc}(4 \rightarrow 5) \\ x_{46} \leq 600 &\quad \text{arc}(4 \rightarrow 6) \\ x_{56} \leq 600 &\quad \text{arc}(5 \rightarrow 6) \end{aligned}$$

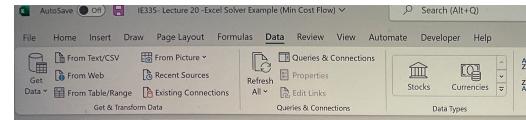
All variables  $\geq 0$  (can't negative)



Cell	Name	Original Value	Final Value
\$K\$55	=TC	0	95000
A12			
A13			
Solver Cells			
Cell	Name	Original Value	Final Value
\$D\$55	Amount	0	700
\$D\$56	Amount	0	200
\$D\$27	Amount	0	600
\$D\$21	Amount	0	100
\$D\$59	Amount	0	200
\$D\$22	Amount	0	300
\$D\$11	Amount	0	400
\$D\$12	Amount	0	400

Cell	Name	Original Value	Final Value
\$D\$55	Amount	0	700
\$D\$56	Amount	0	200
\$D\$27	Amount	0	600
\$D\$21	Amount	0	100
\$D\$59	Amount	0	200
\$D\$22	Amount	0	300
\$D\$11	Amount	0	400
\$D\$12	Amount	0	400

Cell	Name	Original Value	Final Value
\$D\$55	=TC	95000	95000
A12			
A13			
Solver Cells			
Cell	Name	Original Value	Final Value
\$D\$55	Amount	700	700
\$D\$56	Amount	200	200
\$D\$27	Amount	600	600
\$D\$21	Amount	100	100
\$D\$59	Amount	200	200
\$D\$22	Amount	300	300
\$D\$11	Amount	400	400
\$D\$12	Amount	400	400



FROM	TO	Amount	Capacity	Cost	TC
1	2	700	<=	800	10
1	3	200	<=	600	50
2	4	600	<=	600	30
2	5	100	<=	100	70
3	4	200	<=	300	10
3	5	0	<=	400	60
4	5	400	<=	600	30
4	6	400	<=	400	60
5	6	500	<=	600	30

Amount

x12/x13 900 900

x46/x56 900 900

x24/x25/x12 0 0

x34/x35/x13 0 0

x45/x46/x24/x34 0 0

x56/x45/x35/x25 0 0

Sheet1 Question 1 Answer Report 1 Sensitivity Report 1 Limits Report 1

Microsoft Excel 16.0 Sensitivity Report

Worksheet: [IE335\_Lecture 20\_Easy Solver Example (Min Cost Flow).xlsx]Sheet2

Report Created: 11/15/2022 12:30:03 PM

Variable Cells

Cell	Name	Final Reduced	Objective Coefficient	Allowable Increase	Allowable Decrease
\$D\$55	Amount	700	0	10	10
\$D\$56	Amount	200	0	20	20
\$D\$27	Amount	600	0	30	30
\$D\$21	Amount	100	0	10	10
\$D\$59	Amount	200	0	20	20
\$D\$22	Amount	300	0	10	10
\$D\$11	Amount	400	0	20	20
\$D\$12	Amount	400	0	20	20

Microsoft Excel 16.0 Limits Report

Worksheet: [IE335\_Lecture 20\_Easy Solver Example (Min Cost Flow).xlsx]Sheet2

Report Created: 11/15/2022 12:30:04 PM

Objective

Cell	Name	Value	Shadow Price	Constraint Coefficient	Allowable Increase	Allowable Decrease
\$K\$55	=TC	95000	0	500	0	1E+30
\$D\$55	Amount	950	10	10	0	300
\$D\$56	Amount	950	20	20	0	200
\$D\$27	Amount	950	30	30	0	100
\$D\$21	Amount	950	10	10	0	10
\$D\$59	Amount	950	20	20	0	20
\$D\$22	Amount	950	10	10	0	10
\$D\$11	Amount	950	20	20	0	20
\$D\$12	Amount	950	20	20	0	20

Microsoft Excel 16.0 Limits Report

Worksheet: [IE335\_Lecture 20\_Easy Solver Example (Min Cost Flow).xlsx]Sheet2

Report Created: 11/15/2022 12:30:04 PM

Variables

Cell	Name	Value	Shadow Price	Constraint Coefficient	Allowable Increase	Allowable Decrease
\$D\$55	Amount	700	0	10	10	1E+30
\$D\$56	Amount	200	0	20	20	200
\$D\$27	Amount	600	0	30	30	100
\$D\$21	Amount	100	0	10	10	10
\$D\$59	Amount	200	0	20	20	20
\$D\$22	Amount	300	0	10	10	10
\$D\$11	Amount	400	0	20	20	20
\$D\$12	Amount	400	0	20	20	20

Microsoft Excel 16.0 Limits Report

Worksheet: [IE335\_Lecture 20\_Easy Solver Example (Min Cost Flow).xlsx]Sheet2

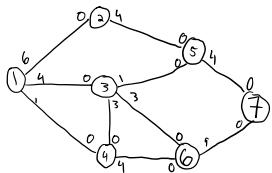
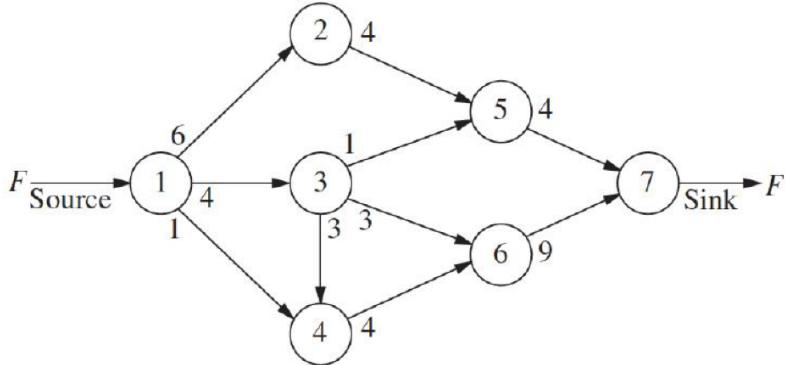
Report Created: 11/15/2022 12:30:04 PM

Constraints

Cell	Name	Value	Shadow Price	Constraint Coefficient	Allowable Increase	Allowable Decrease
\$D\$55	z12/z13	950	0	500	0	1E+30
\$D\$56	z12/z13	950	10	10	0	300
\$D\$27	z24/z25/z12	0	-10	0	700	0
\$D\$21	z24/z25/z12	0	-20	0	20	10
\$D\$59	z34/z35/z13	0	-10	0	20	10
\$D\$22	z34/z35/z13	0	-20	0	20	10
\$D\$11	z45/z46/z24/z34	0	-60	0	200	0
\$D\$12	z45/z46/z24/z34	0	-60	0	200	0

Sheet1 Question 1 Answer Report 1 Sensitivity Report 1 Limits Report 1

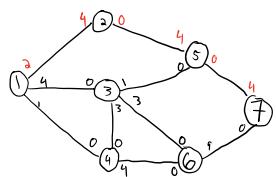
**(20pts) Question 2:** For the network shown below, use the augmenting path algorithm to find the flow pattern giving the maximum flow from the source to the sink, given that the arc capacity from node  $i$  to node  $j$  is the number nearest node  $i$  along the arc between these nodes. Show your work.



Round 1:

Path:  $\{1, 3, 5, 7\}$

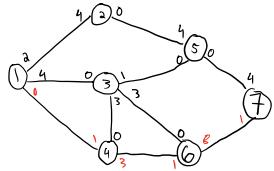
Min:  $\epsilon_{1,3,5,7} = 4$



Round 2:

Path:  $\{1, 4, 6, 7\}$

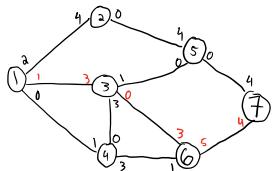
Min:  $\epsilon_{1,4,6,7} = 1$



Round 3:

Path:  $\{1, 3, 6, 7\}$

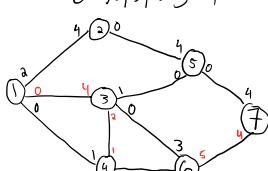
Min:  $\epsilon_{1,3,6,7} = 3$



Round 4:

Path:  $\{1, 3, 4, 6, 7\}$

Min:  $\epsilon_{1,3,4,6,7} = 1$

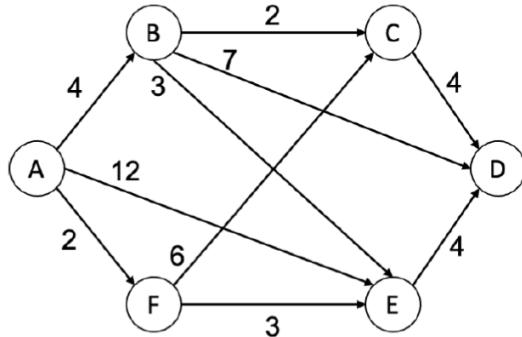


Arc Flow

$1-2$ : 4  
 $1-3$ : 4  
 $1-4$ : 1  
 $2-5$ : 4  
 $3-4$ : 1  
 $3-5$ : 0  
 $3-6$ : 3  
 $4-6$ : 2  
 $5-7$ : 4  
 $6-7$ : 5

Max Flow = 9

**(10pts) Question 3:** For the graph below, write the LP formulation to find the maximum flow from node A to node D. (Define/show any required variable, node, arc, etc.)



$$\text{Maximize: } Z = x_{AD} + x_{ED} + x_{BD}$$

$$\text{S.t.: } x_{AB} - x_{BC} - x_{BD} - x_{BE} = 0 \quad \text{Node B}$$

$$x_{AF} - x_{FC} - x_{FE} = 0 \quad \text{Node F}$$

$$x_{BC} + x_{FC} - x_{CD} = 0 \quad \text{Node C}$$

$$x_{FE} + x_{AE} + x_{BE} - x_{ED} = 0 \quad \text{Node E}$$

$$x_{AD} \leq 4 \quad \text{arc(A-B)}$$

$$x_{AF} \leq 2 \quad \text{arc(A-F)}$$

$$x_{AE} \leq 12 \quad \text{arc(A-E)}$$

$$x_{BC} \leq 2 \quad \text{arc(B-C)}$$

$$x_{BD} \leq 7 \quad \text{arc(B-D)}$$

$$x_{BE} \leq 3 \quad \text{arc(B-E)}$$

$$x_{CD} \leq 4 \quad \text{arc(C-D)}$$

$$x_{ED} \leq 4 \quad \text{arc(E-D)}$$

$$x_{FC} \leq 6 \quad \text{arc(F-C)}$$

$$x_{FE} \leq 3 \quad \text{arc(F-E)}$$

All arcs non negative ( $\geq 0$ )

**(15pts) Question 4:** Considering the following parameter table and LP formulation, solve this problem using **Gurobi in Python**. Submit a screenshot of your code and a screenshot of your results including the final objective value and non-zero decision variables.

	1	2	3	4	Supply
A	8	6	5	7	1
B	6	5	3	4	1
C	6	7	5	6	1
D	6	7	5	6	1
Demand	1	1	1	1	

$$\underset{x_{ij}}{\text{minimize}} \quad \sum_i \sum_j c_{ij} x_{ij}$$

$$\text{subject to } \sum_j x_{ij} = 1 \quad \text{for } i = A, B, C, D,$$

$$\sum_i x_{ij} = 1 \quad \text{for } j = 1, 2, 3, 4,$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } i, j$$

```
class HW5_QP_3:
    def __init__(self):
        from gurobipy import *
        # Create a new Gurobi Model
        self.m = Model("HW5_3")
        # Create new variables
        # (GRB.BINARY, GRB.INTEGER, GRB.CONTINUOUS, or GRB.SEMINT)
        at = m.addVar(name="at")
        bt = m.addVar(name="bt")
        ct = m.addVar(name="ct")
        dt = m.addVar(name="dt")
        at2 = m.addVar(name="at2")
        bt2 = m.addVar(name="bt2")
        ct2 = m.addVar(name="ct2")
        dt2 = m.addVar(name="dt2")
        # Set the objective function I
        m.setObjective(at + bt + ct + dt + 5*at2 + 5*bt2 + 3*ct2 + 4*dt2 + 5*at2 + 7*bt2 + 5*ct2 + 6*dt2 + 7*at2 + 5*bt2 + 6*ct2, GRB.MINIMIZE)

        # Add Constraints
        m.addConstr(at + bt + ct + dt == 1, "C1")
        m.addConstr(bt + bt2 + ct2 + dt2 == 1, "C2")
        m.addConstr(ct + ct2 + dt + dt2 == 1, "C3")
        m.addConstr(at + at2 + bt + bt2 + ct + ct2 + dt + dt2 == 1, "C4")

        # Optimize the model with 4 rows, 16 columns and 32 nonzeros
        Model.Fingerprint("HW5_3")
```

VARS

OBJ Function

```
PS C:\Users\conor\OneDrive\Documents\STATE 355 & C:\Users\conor\AppData\Local\Microsoft\WindowsApps\python3.8.exe "C:\Users\conor\OneDrive\Documents\STATE 355\STATE 355.ipynb"

$$\begin{array}{l} \text{# Import the required libraries} \\ \text{from pulp import *} \\ \\ \text{# Create a new LP problem} \\ \text{prob = LpProblem("STATE 355", LpMaximize)} \\ \\ \text{# Define the decision variables} \\ \text{a = LpVariable("a", lowBound=0, upBound=100, cat="Continuous")} \\ \text{b = LpVariable("b", lowBound=0, upBound=100, cat="Continuous")} \\ \text{c = LpVariable("c", lowBound=0, upBound=100, cat="Continuous")} \\ \text{d = LpVariable("d", lowBound=0, upBound=100, cat="Continuous")} \\ \\ \text{# Add the objective function} \\ prob += a + b + c + d, "Z" \\ \\ \text{# Add constraints} \\ prob += a + b + c + d == 1, "C1" \\ prob += a - b == 1, "C2" \\ prob += b - c == 1, "C3" \\ prob += c - d == 1, "C4" \\ prob += d - a == 1, "C5" \\ \\ \text{# Add more constraints} \\ prob += a + b + c + d == 1, "C6" \\ prob += a + b + c + d == 1, "C7" \\ \\ \text{# Solve the model} \\ prob.solve() \\ \\ \text{# Print the feasible solution if optimal.} \\ if prob.status == 1: \\     print("Optimal Solution found") \\     print(f"Function value: {prob.objective.value()}") \\     print(f"Variables: {prob.variables()}") \\     for v in prob.variables(): \\         print(f" {v.name}: {v.varValue}") \\ \\ \text{# Another way to print the variable values} \\ \\ \text{PROGRAMMING} \quad \text{COLUMNS} \quad \text{DEBUG CONSOLE} \quad \text{TERMINAL} \quad \text{APPROX} \\ PS C:\Users\conor\OneDrive\Documents\STATE 355 & C:\Users\conor\AppData\Local\Microsoft\WindowsApps\python3.8.exe "C:\Users\conor\OneDrive\Documents\STATE 355\STATE 355.ipynb" \\ \\ \text{Pyomo} \\ \text{Pyomo version: 5.8.0} \\ \text{Pyomo license: Restricted license - For non-production use only - exp: 2024-10-26} \\ \text{Gurobi Python interface version: 10.0.0-py3.8-win64 (win64)} \\ \\ \text{CPU model: Intel(R) Core(TM) i9-13900K CPU @ 3.00GHz, instruction set SSE2|AVX|AVX2} \\ Thread count: 4 physical cores, 8 logical processors, using up to 8 threads \\ \\ Optimize a model with 8 rows, 16 columns and 32 nonzeros \\ Model Statistics: 8 binary, 0 integer, 0 continuous \\ Coefficient statistics: Matrix range [1e+00, 1e+00] \\ Column range [1e+00, 1e+00] \\ Nonzero range [1e+00, 1e+00]$$

```

IE335\_HW5\_Q4.ipynb

```
#ie335_HW5_Q4.ipynb > ...
46 m.addConstr( (a1 + b1 + c1 + d1) == 1, "c4" )
49 m.addConstr( (a2 + b2 + c2 + d2) == 1, "c5" )
50 m.addConstr( (a3 + b3 + c3 + d3) == 1, "c6" )
51 m.addConstr( (a4 + b4 + c4 + d4) == 1, "c7" )
52
53 # Solve the model
54 m.optimize()

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

Objective range [3e+00, 8e+00]
Bounds range [0e+00, 0e+00]
RHS range [1e+00, 1e+00]
Presolve time: 0.01s
Presolved: 8 rows, 16 columns, 32 nonzeros

Iteration    Objective    Primal Inf.    Dual Inf.    Time
  0    2.1000000e+01    6.000000e+00    0.000000e+00    0s
  3    2.1000000e+01    0.000000e+00    0.000000e+00    0s

Solved in 3 iterations and 0.01 seconds (0.00 work units)
Optimal objective: 2.100000000e+01
Obj Function: 21.0
Optimal Solution: 21.0
at 0.0
a1 = 1
a2 = 0
a3 = 0.0
a4 = 0.0
b1 = 0.0
b2 = 0.0
b3 = 0.0
b4 = 1.0
c1 = 1
c2 = 0.0
c3 = 0.0
c4 = 0.0
d1 = 0.0
d2 = 0.0
d3 = 1
d4 = 0.0

```

**(15pts) Question 5:** Considering the following parameter table and LP formulation, solve this problem using **Gurobi in Python**. Submit a screenshot of your code and a screenshot of your results including the final objective value and non-zero decision variables.

	1	2	3	4(dummy)	Supply
A	\$800	\$700	\$400	0	50
B	\$600	\$800	\$500	0	50
Demand	20	20	20	40	

$$\begin{aligned} & \text{minimize}_{x_{ij}} \quad \sum_i \sum_j c_{ij} x_{ij} \\ & \text{subject to} \quad \sum_j x_{ij} = S_i \quad \text{for } i = A, B, \\ & \quad \sum_i x_{ij} = D_j \quad \text{for } j = 1, 2, 3, 4, \\ & \quad x_{ij} \geq 0 \quad \text{for all } i, j \end{aligned}$$

where  $c_{ij}$  are the shipping cost for each truck from  $i$  to  $j$ ,  $S_i$  is the maximum supply of trucks from plant  $i$ , and  $D_j$  is the exact demand of trucks shipped to distribution center  $j$ .

```

File Edit Selection View Go Run Terminal Help
File View Go Run Terminal Help
File View Go Run Terminal Help
IE 335 HW 2 Mccar12... IE 335 HW 2 Mccar12... IE 335 HW 2 Mccar12...
IE 335 HW 3 Mccar12... IE 335 HW 3 Mccar12... IE 335 HW 3 Mccar12...
IE 335 HW Excel.xlsx IE 335 HW Excel.xlsx IE 335 HW Excel.xlsx
IE 335 HW Q3.py IE 335 HW Q3.py IE 335 HW Q3.py
IE 335 Lecture 20 - Ex... IE 335 Lecture 20 - Ex... IE 335 Lecture 20 - Ex...
IE 335.py IE 335.py IE 335.py
In class submission 2 ... In class submission 2 ... In class submission 2 ...

```

```

File Edit Selection View Go Run Terminal Help
File View Go Run Terminal Help
File View Go Run Terminal Help
IE 335 HW 2 Mccar12... IE 335 HW 2 Mccar12... IE 335 HW 2 Mccar12...
IE 335 HW 3 Mccar12... IE 335 HW 3 Mccar12... IE 335 HW 3 Mccar12...
IE 335 HW Excel.xlsx IE 335 HW Excel.xlsx IE 335 HW Excel.xlsx
IE 335 HW Q3.py IE 335 HW Q3.py IE 335 HW Q3.py
IE 335 Lecture 20 - Ex... IE 335 Lecture 20 - Ex... IE 335 Lecture 20 - Ex...
IE 335.py IE 335.py IE 335.py
In class submission 2 ... In class submission 2 ... In class submission 2 ...

```

```

File Edit Selection View Go Run Terminal Help
File View Go Run Terminal Help
File View Go Run Terminal Help
IE 335 HW 2 Mccar12... IE 335 HW 2 Mccar12... IE 335 HW 2 Mccar12...
IE 335 HW 3 Mccar12... IE 335 HW 3 Mccar12... IE 335 HW 3 Mccar12...
IE 335 HW Excel.xlsx IE 335 HW Excel.xlsx IE 335 HW Excel.xlsx
IE 335 HW Q3.py IE 335 HW Q3.py IE 335 HW Q3.py
IE 335 Lecture 20 - Ex... IE 335 Lecture 20 - Ex... IE 335 Lecture 20 - Ex...
IE 335.py IE 335.py IE 335.py
In class submission 2 ... In class submission 2 ... In class submission 2 ...

```

```

Optional Solution: 34000.0
a1 0.0
a2 20.0
a3 20.0
a4 10.0
b1 20.0
b2 0.0
b3 0.0
b4 30.0
PS C:\Users\conor\OneDrive\Documents\IE 335>

```

```

Optional Solution: 34000.0
a1 0.0
a2 20.0
a3 20.0
a4 10.0
b1 20.0
b2 0.0
b3 0.0
b4 30.0
PS C:\Users\conor\OneDrive\Documents\IE 335>

```

```

Optional Solution: 34000.0
a1 0.0
a2 20.0
a3 20.0
a4 10.0
b1 20.0
b2 0.0
b3 0.0
b4 30.0
PS C:\Users\conor\OneDrive\Documents\IE 335>

```

```

IE 335
IE 335 HW 2 Mccar12...
IE 335 HW 3 Mccar12...
IE 335 HW Excel.xlsx
IE 335 HW Q3.py
IE 335 Lecture 20 - Ex...
IE 335.py
In class submission 2 ...

```

```

# Create a new model
m = Model(name="IE335")
# Create two new variables
a1 = m.addVar(lb=0, name="a1") # Variable a1
a2 = m.addVar(lb=0, name="a2") # Variable a2
a3 = m.addVar(lb=0, name="a3") # Variable a3
a4 = m.addVar(lb=0, name="a4") # Variable a4
b1 = m.addVar(lb=0, name="b1") # Variable b1
b2 = m.addVar(lb=0, name="b2") # Variable b2
b3 = m.addVar(lb=0, name="b3") # Variable b3
b4 = m.addVar(lb=0, name="b4") # Variable b4
# Set the objective function
m.setObjective(a1 + a2 + a3 + a4 == 10, "c1") # Objective function
# Add Constraints
m.addConstr(a1 + a2 + a3 + a4 == 10, "c1") # Constraint c1
m.addConstr(b1 + b2 + b3 + b4 == 20, "c2") # Constraint c2
m.addConstr(b1 + b2 + b3 == 10, "c3") # Constraint c3
m.addConstr(b3 + b4 == 20, "c4") # Constraint c4
m.addConstr(b4 == 40, "c5") # Constraint c5
# Solve the model
m.optimize()
# Print the feasible solution if optimal.
if m.status == GRB.Status.OPTIMAL:
    print("Optimal Solution:", m.objVal)
    for v in m.getVars():
        print(v.varName, v.x)

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Optional Solution: 34000.0
a1 0.0
a2 20.0
a3 20.0
a4 10.0
b1 20.0
b2 0.0
b3 0.0
b4 30.0
PS C:\Users\conor\OneDrive\Documents\IE 335>

```

```

Optional Solution: 34000.0
a1 0.0
a2 20.0
a3 20.0
a4 10.0
b1 20.0
b2 0.0
b3 0.0
b4 30.0
PS C:\Users\conor\OneDrive\Documents\IE 335>

```

```

Optional Solution: 34000.0
a1 0.0
a2 20.0
a3 20.0
a4 10.0
b1 20.0
b2 0.0
b3 0.0
b4 30.0
PS C:\Users\conor\OneDrive\Documents\IE 335>

```

```

Iteration Objective Primal Inf. Dual Inf. Time
0 3.400000e+04 0.000000e+00 0.000000e+00 0s
0 3.400000e+04 0.000000e+00 0.000000e+00 0s

Solved in 0 iterations and 0.00 seconds (0.00 work units)
Optimal objective 3.4000000000e+04
Optimal Solution: 34000.0
a1 = 0.0
a2 = 20.0
a3 = 20.0
a4 = 10.0
b1 = 20.0
b2 = 0.0
b3 = 0.0
b4 = 30.0
PS C:\Users\conor\OneDrive\Documents\IE 335>

```

**(15pts) Question 6:** Solve Problem 4.4-6 in the textbook using **Gurobi in Python**. Submit a screenshot of your code and a screenshot of your results including the final objective value and non-zero decision variables. Only use the Problem formulation from the Problem.

The screenshot shows two instances of Microsoft Studio Code running side-by-side. Both instances have the title bar "ie335\_HW5\_Q6.py - IE 335 - Visual Studio Code".

**Left Window:**

- File Explorer:** Shows several files including "ie335\_HW5\_Q6.py" (the current file), "ie335\_HW5\_Q4.py", "ie335\_HW5\_Q5.py", and "ie335\_HW5\_Q6.py" (another instance).
- Code Editor:** Displays the following Python code for a linear programming model using the gurobipy library:

```
1 # LP:
2 # min 3x + 5y + 6z
3 # s.t.
4 # 2x + y + z <= 4
5 # x + 2y + z <= 4
6 #
7 # x + y + 2z <= 4
8 # x + y + z <= 3
9 # x,y >= 0
10
11 from gurobipy import *
12
13 # Create a new Gurobi Model
14 m = Model( name="lp")
15
16 # Create two new variables
17 # (GRB.CONTINUOUS, GRB.BINARY, GRB.INTEGER, GRB.SEMICONT, or GRB.SEMIINT)
18 x = m.addVar(lb=0, name="x")
19 y = m.addVar(lb=0, name="y")
20 z = m.addVar(lb=0, name="z")
21
22 # Set the objective function
23 m.setObjective(3*x + 5*y + 6*z, GRB.MAXIMIZE)
24
25 #Add Constraints
26 m.addConstr(2*x + y + z <= 4, "c0")
27 m.addConstr(x + 2*y + z <= 4, "c1")
28 m.addConstr(x + y + 2*z <= 4, "c2")
29 m.addConstr(x + y + z <= 3, "c3")
```

**Annotations:** Handwritten notes are overlaid on the code:

- A yellow arrow points from the handwritten note "UP PROBLEM" to the first constraint line.
- A yellow arrow points from the handwritten note "WATINg UNS" to the objective function line.
- A yellow arrow points from the handwritten note "CONSTRAINTS" to the constraint block.
- A yellow arrow points from the handwritten note "OPTIMIZATION STATEMENT" to the print statement in the right window.

**Right Window:**

- File Explorer:** Shows the same file structure as the left window.
- Code Editor:** Displays the same Python code as the left window.
- Terminal:** Shows the output of the code execution:

```
1 Solved in 3 iterations and 0.00 seconds (0.00 work units)
2 y 1.3333333333333335
3 z 1.3333333333333333
```

**Annotations:** Handwritten notes are overlaid on the terminal output:

- A yellow arrow points from the handwritten note "Solved in 3 iterations and 0.00 seconds (0.00 work units)" to the "Solved in 3 iterations" line.
- A yellow arrow points from the handwritten note "OPTIMIZATION STATEMENT" to the "print('Optimal solution:', x.Objsense)" line.