HW 5 #4-(c)

the LP problem.

$$
\begin{aligned}
\text{minimize } w = {} & 40y_1 + 30y_2 + 20y_3 + 10y_4 + 10y_5 \\
& y_1 \qquad\quad +2y_3 \quad +y_4 \qquad\quad \geq 300 \\
\text{subject to } & y_1 \quad +y_2 \qquad\qquad +2y_5 \geq 400 \\
& \quad 2y_2 \qquad +2y_4 \quad +y_5 \geq 1000 \\
& y_1, \ldots, y_5 \geq 0 \\
& \text{and } y_1, y_2, y_3, y_4, y_5 \text{ are integers.}
\end{aligned}
$$

We import pulp, in the following two cells.

In [1]:
```
import sys
!{sys.executable} -m pip install pulp
```

```
Requirement already satisfied: pulp in /opt/conda/lib/python3.8/site
-packages (2.3.1)
Requirement already satisfied: amply>=0.1.2 in /opt/conda/lib/python
3.8/site-packages (from pulp) (0.1.4)
Requirement already satisfied: pyparsing in /opt/conda/lib/python3.8
/site-packages (from amply>=0.1.2->pulp) (2.4.7)
Requirement already satisfied: docutils>=0.3 in /opt/conda/lib/pytho
n3.8/site-packages (from amply>=0.1.2->pulp) (0.15.2)
```

In [2]:
```
import pulp
```

In [3]:
```
# Create a LP Minimization problem
Lp_prob = pulp.LpProblem('Your_LP_Problem', pulp.LpMinimize)  # We set
up theproblem using the command LpProblem in the PuLP package.
```

In [4]:
```
# Create problem Decision Variables
y_1 = pulp.LpVariable("y_1", lowBound=0, cat='Integer')
y_2 = pulp.LpVariable("y_2",lowBound=0, cat='Integer')
y_3 = pulp.LpVariable("y_3", lowBound=0,cat='Integer')
y_4 = pulp.LpVariable("y_4",lowBound=0, cat='Integer')
y_5 = pulp.LpVariable("y_5", lowBound=0,cat='Integer')
```

In [5]:
```
# Objective Function
Lp_prob += 40* y_1 + 30*y_2 + 20 * y_3 + 10 * y_4 + 10* y_5

# Constraints:
Lp_prob += y_1 + 2*y_3 + y_4 >= 300
Lp_prob += y_1 + y_2  + 2* y_5 >= 400
Lp_prob += 2*y_2 + 2*y_4 + y_5 >= 1000
Lp_prob += y_1 >= 0
Lp_prob += y_2 >= 0
Lp_prob += y_3 >= 0
Lp_prob += y_4 >= 0
Lp_prob += y_5 >= 0
```

In [6]:
```
# Display the problem
print(Lp_prob)
```

```
Your_LP_Problem:
MINIMIZE
40*y_1 + 30*y_2 + 20*y_3 + 10*y_4 + 10*y_5 + 0
SUBJECT TO
_C1: y_1 + 2 y_3 + y_4 >= 300

_C2: y_1 + y_2 + 2 y_5 >= 400

_C3: 2 y_2 + 2 y_4 + y_5 >= 1000

_C4: y_1 >= 0

_C5: y_2 >= 0

_C6: y_3 >= 0

_C7: y_4 >= 0

_C8: y_5 >= 0

VARIABLES
0 <= y_1 Integer
0 <= y_2 Integer
0 <= y_3 Integer
0 <= y_4 Integer
0 <= y_5 Integer
```

In [7]:
```
Lp_prob.solve()
pulp.LpStatus[Lp_prob.status]
```

Out[7]: 'Optimal'

In [8]:
```
# Printing the final solution
print("y_1=", pulp.value(y_1), "y_2=", pulp.value(y_2),  "y_3=", pulp.
value(y_3), "y_4=", pulp.value(y_4), "y_5=", pulp.value(y_5), "w=", pu
lp.value(Lp_prob.objective))
```

```
y_1= 0.0 y_2= 0.0 y_3= 0.0 y_4= 400.0 y_5= 200.0 w= 6000.0
```

Another way to show the solutions:

In [9]:
```
for a in Lp_prob.variables():
        print(a.name, "=", a.varValue)
print("Optimal value is w = ", pulp.value(Lp_prob.objective))
```

```
y_1 = 0.0
y_2 = 0.0
y_3 = 0.0
y_4 = 400.0
y_5 = 200.0
Optimal value is w =  6000.0
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: