Homework 2 (Vanderbei Exercise 1.2)

In [14]:
```
import sys
!{sys.executable} -m pip install pulp
```

```
Requirement already satisfied: pulp in c:\users\muntakim\appdata\local\programs\python\python38-32\lib\site-packages (2.4)
Requirement already satisfied: amply>=0.1.2 in c:\users\muntakim\appdata\local\programs\python\python38-32\lib\site-packages (from pulp) (0.1.4)
Requirement already satisfied: docutils>=0.3 in c:\users\muntakim\appdata\local\programs\python\python38-32\lib\site-packages (from amply>=0.1.2->pulp) (0.17.1)
Requirement already satisfied: pyparsing in c:\users\muntakim\appdata\local\programs\python\python38-32\lib\site-packages (from amply>=0.1.2->pulp) (2.4.7)
```

In [15]:
```
import pulp
from pulp import *
```

In [16]:
```
"""
The Airline Revenue Maximization Problem

Author: Muntakim Rahman 2020
"""
```

Out[16]:
```
'\nThe Airline Revenue Maximization Problem\n\nAuthor: Muntakim Rahman 2020\n'
```

Textbook Problem

1.2 A small airline, Ivy Air, flies between three cities: Ithaca, Newark, and Boston. They offer several flights but, for this problem, let us focus on the Friday afternoon flight that departs from Ithaca, stops in Newark, and continues to Boston. There are three types of passengers:

(a) Those traveling from Ithaca to Newark.
(b) Those traveling from Newark to Boston.
(c) Those traveling from Ithaca to Boston.

The aircraft is a small commuter plane that seats 30 passengers. The airline offers three fare classes:

(a) Y class: full coach.
(b) B class: nonrefundable.
(c) M class: nonrefundable, 3-week advanced purchase.

Ticket prices, which are largely determined by external influences (i.e., competitors), have been set and advertised as follows:

	Ithaca–Newark	Newark–Boston	Ithaca–Boston
Y	300	160	360
B	220	130	280
M	100	80	140

Based on past experience, demand forecasters at Ivy Air have determined the following upper bounds on the number of potential customers in each of the nine possible origin-destination/fare-class combinations:

	Ithaca–Newark	Newark–Boston	Ithaca–Boston
Y	4	8	3
B	8	13	10
M	22	20	18

The goal is to decide how many tickets from each of the nine origin/destination/fare-class combinations to sell. The constraints are that the plane cannot be overbooked on either of the two legs of the flight and that the number of tickets made available cannot exceed the forecasted maximum demand. The objective is to maximize the revenue. Formulate this problem as a linear programming problem.

Airline Revenue Maximization Problem

$$\text{Maximize}$$
$$\begin{bmatrix}300\\220\\100\end{bmatrix}\begin{bmatrix}I-N_Y & I-N_B & I-N_M\end{bmatrix}+\begin{bmatrix}160\\130\\80\end{bmatrix}\begin{bmatrix}N-B_Y & N-B_B & N-B_M\end{bmatrix}+\begin{bmatrix}360\\280\\140\end{bmatrix}$$

$$\text{Subject to}$$
$$\begin{bmatrix}I-N_Y & I-N_B & I-N_M\end{bmatrix}\leq\begin{bmatrix}4 & 8 & 22\end{bmatrix}$$

$$\begin{bmatrix}N-B_Y & N-B_B & N-B_M\end{bmatrix}\leq\begin{bmatrix}8 & 13 & 20\end{bmatrix}$$

$$\begin{bmatrix}I-B_Y & I-B_B & I-B_M\end{bmatrix}\leq\begin{bmatrix}3 & 10 & 18\end{bmatrix}$$

$$\begin{bmatrix}I-N_Y & I-N_B & I-N_M\end{bmatrix}+\begin{bmatrix}I-B_Y & I-B_B & I-B_M\end{bmatrix}\leq 30$$

$$\begin{bmatrix}N-B_Y & N-B_B & N-B_M\end{bmatrix}+\begin{bmatrix}I-B_Y & I-B_B & I-B_M\end{bmatrix}\leq 30$$

$$[I - N_Y \quad I - N_B \quad I - N_M], \quad [N - B_Y \quad N - B_B \quad N - B_M], \quad [I - B_Y \quad I - B_B \quad I$$

In [17]:

```
# Creates Lists of the Fare-Class Combinations for each Origin/Destination.
labels_IN = ['Ithaca-Newark_Y', 'Ithaca-Newark_B', 'Ithaca-Newark_M']
labels_NB = ['Newark-Boston_Y', 'Newark-Boston_B', 'Newark-Boston_M']
labels_IB = ['Ithaca-Boston_Y', 'Ithaca-Boston_B', 'Ithaca-Boston_M']

# Creates Lists of the Ticket Prices for each Origin/Destination.
ticket_prices_IN = [300, 220, 100]
ticket_prices_NB = [160, 130, 80]
ticket_prices_IB = [360, 280, 140]

# Creates Lists of the Forecasted Demand Bounds for each Origin/Destination.
potential_customers_IN = [4, 8, 22]
potential_customers_NB = [8, 13, 20]
potential_customers_IB = [3, 10, 18]

# Initialize Variables for Constraints.
max_passengers = 30
number_of_origin_destination_combinations = 3
```

In [18]:

```
# Create the 'LP_Prob' Variable to Contain the Problem Data for the LP Maximization Pro
LP_Prob = LpProblem("Ivy_Air_Problem", LpMaximize)
```

In [19]:

```
# Create Lists of Empty Strings to Contain the Referenced Variables.
tickets_IN = ['', '', '']
tickets_NB = ['', '', '']
tickets_IB = ['', '', '']

# Create Decision Variables.
for i in range(number_of_origin_destination_combinations) :
    tickets_IN[i] = LpVariable(name = str(labels_IN[i]), lowBound = 0, cat = 'Integer')
    tickets_NB[i] = LpVariable(name = str(labels_NB[i]), lowBound = 0, cat = 'Integer')
    tickets_IB[i] = LpVariable(name = str(labels_IB[i]), lowBound = 0, cat = 'Integer')
```

In [20]:

```
## Print Lists of Different Tickets -> Mainly for Debugging Purposes.
print(tickets_IN)
print(tickets_NB)
print(tickets_IB)
```

```
[Ithaca_Newark_Y, Ithaca_Newark_B, Ithaca_Newark_M]
[Newark_Boston_Y, Newark_Boston_B, Newark_Boston_M]
[Ithaca_Boston_Y, Ithaca_Boston_B, Ithaca_Boston_M]
```

In [21]:

```
Prob_IN = lpSum([ticket_prices_IN[i] * tickets_IN[i] for i in range(number_of_origin_de
Prob_NB = lpSum([ticket_prices_NB[i] * tickets_NB[i] for i in range(number_of_origin_de
Prob_IB = lpSum([ticket_prices_IB[i] * tickets_IB[i] for i in range(number_of_origin_de

## Print Lists of Different Tickets -> Mainly for Debugging Purposes.
print(f'Objective Function Coefficients for IN : {Prob_IN}')
```

```python
print(f'Objective Function Coefficients for NB : {Prob_NB}')
print(f'Objective Function Coefficients for IB : {Prob_IB}')

# The Objective Function is Added to 'LP_Prob' First.
LP_Prob += Prob_IN + Prob_NB + Prob_IB, 'Total_Revenue_of_Ivy_Air'
```

```
Objective Function Coefficients for IN : 220*Ithaca_Newark_B + 100*Ithaca_Newark_M + 300
*Ithaca_Newark_Y
Objective Function Coefficients for NB : 130*Newark_Boston_B + 80*Newark_Boston_M + 160*
Newark_Boston_Y
Objective Function Coefficients for IB : 280*Ithaca_Boston_B + 140*Ithaca_Boston_M + 360
*Ithaca_Boston_Y
```

In [22]:

```python
passengers_IN = lpSum([tickets_IN[i] for i in range(number_of_origin_destination_combin
passengers_IB = lpSum([tickets_IB[i] for i in range(number_of_origin_destination_combin
passengers_NB = lpSum([tickets_NB[i] for i in range(number_of_origin_destination_combin

# The Constraints are Added to 'LP_Prob'
LP_Prob += passengers_IN + passengers_IB <= max_passengers, 'Seating_Limit_1'
LP_Prob += passengers_NB + passengers_IB <= max_passengers, 'Seating_Limit_2'

customers_constraint = '_Potential_Customers'
for i in range(number_of_origin_destination_combinations) :
    LP_Prob += tickets_IN[i] <= potential_customers_IN[i], str(labels_IN[i]) + customer
    LP_Prob += tickets_NB[i] <= potential_customers_NB[i], str(labels_NB[i]) + customer
    LP_Prob += tickets_IB[i] <= potential_customers_IB[i], str(labels_IB[i]) + customer
```

In [23]:

```python
print(LP_Prob)
```

```
Ivy_Air_Problem:
MAXIMIZE
280*Ithaca_Boston_B + 140*Ithaca_Boston_M + 360*Ithaca_Boston_Y + 220*Ithaca_Newark_B +
100*Ithaca_Newark_M + 300*Ithaca_Newark_Y + 130*Newark_Boston_B + 80*Newark_Boston_M + 1
60*Newark_Boston_Y + 0
SUBJECT TO
Seating_Limit_1: Ithaca_Boston_B + Ithaca_Boston_M + Ithaca_Boston_Y
 + Ithaca_Newark_B + Ithaca_Newark_M + Ithaca_Newark_Y <= 30

Seating_Limit_2: Ithaca_Boston_B + Ithaca_Boston_M + Ithaca_Boston_Y
 + Newark_Boston_B + Newark_Boston_M + Newark_Boston_Y <= 30

Ithaca_Newark_Y_Potential_Customers: Ithaca_Newark_Y <= 4

Newark_Boston_Y_Potential_Customers: Newark_Boston_Y <= 8

Ithaca_Boston_Y_Potential_Customers: Ithaca_Boston_Y <= 3

Ithaca_Newark_B_Potential_Customers: Ithaca_Newark_B <= 8

Newark_Boston_B_Potential_Customers: Newark_Boston_B <= 13

Ithaca_Boston_B_Potential_Customers: Ithaca_Boston_B <= 10

Ithaca_Newark_M_Potential_Customers: Ithaca_Newark_M <= 22

Newark_Boston_M_Potential_Customers: Newark_Boston_M <= 20

Ithaca_Boston_M_Potential_Customers: Ithaca_Boston_M <= 18

VARIABLES
0 <= Ithaca_Boston_B Integer
```

```
0 <= Ithaca_Boston_M Integer
0 <= Ithaca_Boston_Y Integer
0 <= Ithaca_Newark_B Integer
0 <= Ithaca_Newark_M Integer
0 <= Ithaca_Newark_Y Integer
0 <= Newark_Boston_B Integer
0 <= Newark_Boston_M Integer
0 <= Newark_Boston_Y Integer
```

In [24]:

```
LP_Prob.writeLP('IvyAirModel.lp')
```

Out[24]:

```
[Ithaca_Boston_B,
 Ithaca_Boston_M,
 Ithaca_Boston_Y,
 Ithaca_Newark_B,
 Ithaca_Newark_M,
 Ithaca_Newark_Y,
 Newark_Boston_B,
 Newark_Boston_M,
 Newark_Boston_Y]
```

In [25]:

```
# The Problem is Solved Using PuLP's Choice of Solver.
LP_Prob.solve()
```

Out[25]: 1

In [26]:

```
current_flight_classes = 0

print(f'Status: {LpStatus[LP_Prob.status]} \n')

for variable in LP_Prob.variables() :
    print(f'{variable.name} = {variable.varValue}')

    current_flight_classes += 1
    if (current_flight_classes == len(labels_IN)) :
        current_flight_classes = 0
        print('\n')
if (LpStatus[LP_Prob.status] == 'Optimal') :
    print(f'Optimal Value : Z = {value(LP_Prob.objective)}')
else :
    print(f'No Optimal Value. Status Code : {value(LP_Prob.objective)}')
```

```
Status: Optimal

Ithaca_Boston_B = 10.0
Ithaca_Boston_M = 0.0
Ithaca_Boston_Y = 3.0


Ithaca_Newark_B = 8.0
Ithaca_Newark_M = 5.0
Ithaca_Newark_Y = 4.0


Newark_Boston_B = 9.0
Newark_Boston_M = 0.0
Newark_Boston_Y = 8.0
```

```
Optimal Value : Z = 9790.0
```