Assignment 1 : Question 3 (smart health)

Student_ID : 1115290

In [1]:

```
from gurobipy import *#--------------importing required libraries
import numpy as np

from string import*
from pandas import *
```

In [2]:

```
m = Model("Question 3 problem")#---creating the model
```

```
Using license file C:\Users\Aloukik Aditya\gurobi.lic
Academic license - for non-commercial use only
```

In [3]:

```



P = [
        [130,95],
        [118,83],

]


print(np.matrix(P))
```

```
[[130  95]
 [118  83]]
```

In [4]:

```
P[1]
```

Out[4]:

```
[118, 83]
```

In [ ]:

```

```

In [5]:

```
#y = m.addMVar((7,7), vtype=GRB.BINARY)
```

In [ ]:

```

```

In [6]:

```
M = m.addMVar((2,2), lb=0,ub=1, vtype=GRB.BINARY)#----------------creating 2 by 2 binary ma
```

In [ ]:

```

```

In [7]:

```

for i in range(len(P)):#-------------------this loop will do algorithm part, whihc means se
    const = 0
    const_c = 0

    for j in range(len(P)):#------------------for values
        const += M[j][i]#-------------------------it means that only values with 1 will be
        const_c += M[i][j] #--------------------it means value with 1 will be selected in
    m.addConstr(const == 1)
    m.addConstr(const_c == 1)




# for i in range(len(P)):
#     const_c = 0
#     for j in range(len(P)):#------------------for values
#         const_c += M[i][j]
#     m.addConstr(const_c == 1)
```

In [8]:

```
new = np.zeros([2,2]).astype(int)#-------------
```

In [9]:

```
#-- objective function
# obj = 0
# for v in M:
#     obj += v
#m.setObjective(obj, GRB.MINIMIZE)

obj = 0
for i in range(len(P)):

    for j in range(len(P)):
        obj += P[i][j]*M[i][j]








# obj2 = 0
# #--------------------------
# obj1 = 0
# for i in range(len(P)):
#     obj2 = 0
#     for j in range(len(P)):
#         obj2 += P[j][i]
#     obj1+= obj2 + P[i][j]
#obj = new.sum()

m.setObjective(obj, GRB.MINIMIZE)#------------- this will help minimizing time
```

In [ ]:

In [10]:

```
m.optimize()#-------------using optimize function and it will show the result
```

```
Gurobi Optimizer version 9.0.2 build v9.0.2rc0 (win64)
Optimize a model with 4 rows, 4 columns and 8 nonzeros
Model fingerprint: 0x0af49efd
Variable types: 0 continuous, 4 integer (4 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [8e+01, 1e+02]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Presolve removed 4 rows and 4 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds
Thread count was 1 (of 8 available processors)

Solution count 1: 213

Optimal solution found (tolerance 1.00e-04)
Best objective 2.130000000000e+02, best bound 2.130000000000e+02, gap 0.000
0%
```

In [11]:

```
for i in range(len(P)):

    for j in range(len(P)):
        new[i][j]= P[i][j]*M[i][j].X#--------------------for loop to see the selected grids
```

In [ ]:

```

```

In [12]:

```
new#----------------------grids which are selected are showed below
```

Out[12]:

```
array([[  0,  95],
       [118,   0]])
```

In [13]:

```
#-------------------- The problem here is in both cases the time is 213, I have considered
```

In [14]:

```
minutess = new.sum()
print("minutes", + minutess)#----------------------converting to hours
hours = minutess/60
print("hours", + hours)
```

```
minutes 213
hours 3.55
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
```

In [ ]:

```
```