MP-1 TUTORIAL-6
PRELAB

Problem 1:

Consider the transportation problem presented in the following table:

Factory	Warehouse			Supply
	W_1	W_2	W_3	
F_1	16	20	12	200
F_2	14	8	18	160
F_3	26	24	16	90
Demand	180	120	150	450

MP-1

Tutorial-6

Prelab

1.

Factory	W_1	W_2	W_3	supply
F_1	16	20	12	200
F_2	14	8	18	160
F_3	26	24	16	90
Demand	180	120	150	450

50 16	20	150 12	~~200~~ ~~50~~ 0
40 14	120 8	18	~~160~~ ~~40~~ 0
90 26	24	16	90 0
~~180~~ ~~130~~ 90 0	~~120~~ 0	~~150~~ 0	

Transportation cost = $50*16 + 40*14 + 150*12 + 120*8 + 90*26$

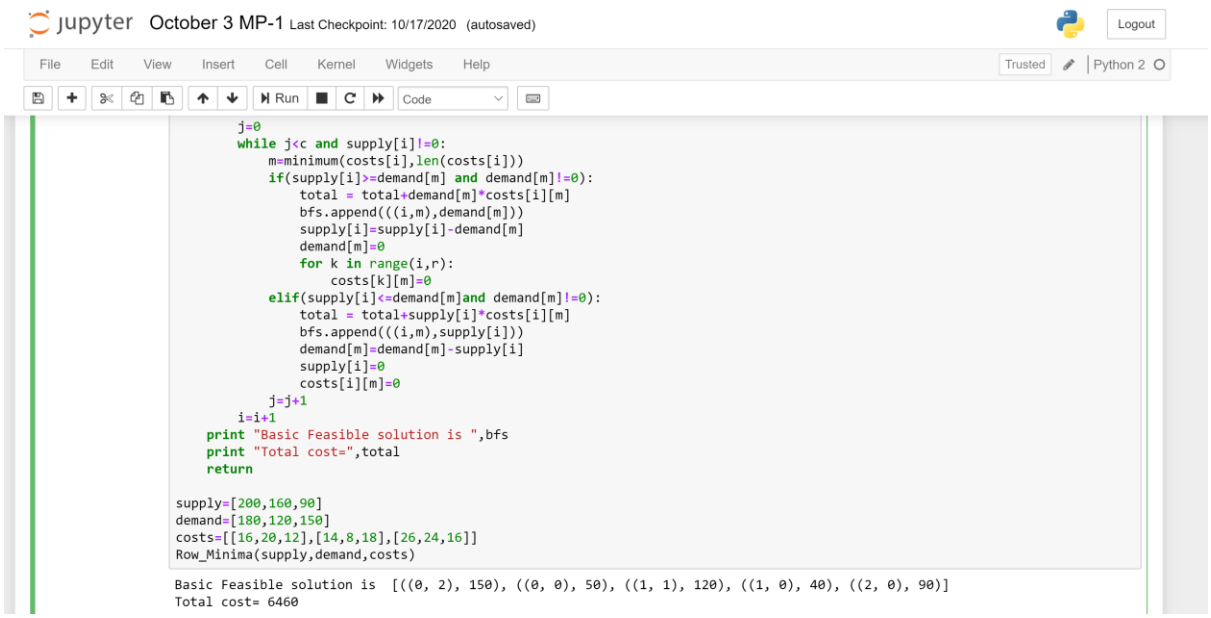
$= 800 + 560 + 1800 + 960 + 2340$

$= 6460$

Code:

```
def minimum(a,n):
   minpos = a.index(min(i for i in a if i>0))
   return minpos
def Row_Minima(supply,demand,costs):
   r=len(costs)
   c=len(costs[0])
   i=0
   j=0
   bfs=[]
   total=0
   while i<r:
      j=0
      while j<c and supply[i]!=0:
         m=minimum(costs[i],len(costs[i]))
         if(supply[i]>=demand[m] and demand[m]!=0):
            total = total+demand[m]*costs[i][m]
            bfs.append(((i,m),demand[m]))
            supply[i]=supply[i]-demand[m]
            demand[m]=0
            for k in range(i,r):
               costs[k][m]=0
         elif(supply[i]<=demand[m]and demand[m]!=0):
            total = total+supply[i]*costs[i][m]
            bfs.append(((i,m),supply[i]))
            demand[m]=demand[m]-supply[i]
            supply[i]=0
            costs[i][m]=0
         j=j+1
      i=i+1
   print "Basic Feasible solution is ",bfs
   print "Total cost=",total
   return

supply=[200,160,90]
demand=[180,120,150]
costs=[[16,20,12],[14,8,18],[26,24,16]]
Row_Minima(supply,demand,costs)
```


jupyter October 3 MP-1 Last Checkpoint: 10/17/2020 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 2
Run Code
j=0
while j<c and supply[i]!=0:
m=minimum(costs[i],len(costs[i]))
if(supply[i]>=demand[m] and demand[m]!=0):
total = total+demand[m]*costs[i][m]
bfs.append(((i,m),demand[m]))
supply[i]=supply[i]-demand[m]
demand[m]=0
for k in range(i,r):
costs[k][m]=0
elif(supply[i]<=demand[m]and demand[m]!=0):
total = total+supply[i]*costs[i][m]
bfs.append(((i,m),supply[i]))
demand[m]=demand[m]-supply[i]
supply[i]=0
costs[i][m]=0
j=j+1
i=i+1
print "Basic Feasible solution is ",bfs
print "Total cost=",total
return
supply=[200,160,90]
demand=[180,120,150]
costs=[[16,20,12],[14,8,18],[26,24,16]]
Row_Minima(supply,demand,costs)
Basic Feasible solution is [((0, 2), 150), ((0, 0), 50), ((1, 1), 120), ((1, 0), 40), ((2, 0), 90)]
Total cost= 6460

190031187 **NERELLA VENKATA RADHAKRISHNA**

INLAB

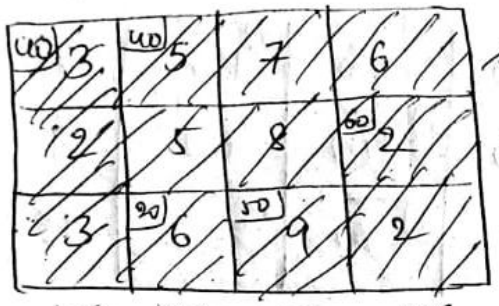
Problem 1:

The Ushodaya departmental store has three plants located throughout a state with production capacity 80, 60 and 70 kilo grams of rice. Each day the firm must furnish its four retail shops R_1, R_2, R_3, & R_4 with at least 40, 60, 50, and 60 gallons respectively. The transportation costs (in Rs.) are given below.

Store	Retail Shop				Supply
	1	**2**	**3**	**4**	
1	3	5	7	6	80
2	2	5	8	2	60
3	3	6	9	2	70
Demand	40	60	50	60	

Inlab

store	Retail shop				supply
	1	2	3	4	
1	3	5	7	6	80
2	2	5	8	2	60
3	3	6	9	2	70
Demand	40	60	50	60	



80 40 0

60 0

70 50

40 60 50 60

0 0 0 0

$40*3 + 40*5 + 20*6 + 50*9 + 60*2$

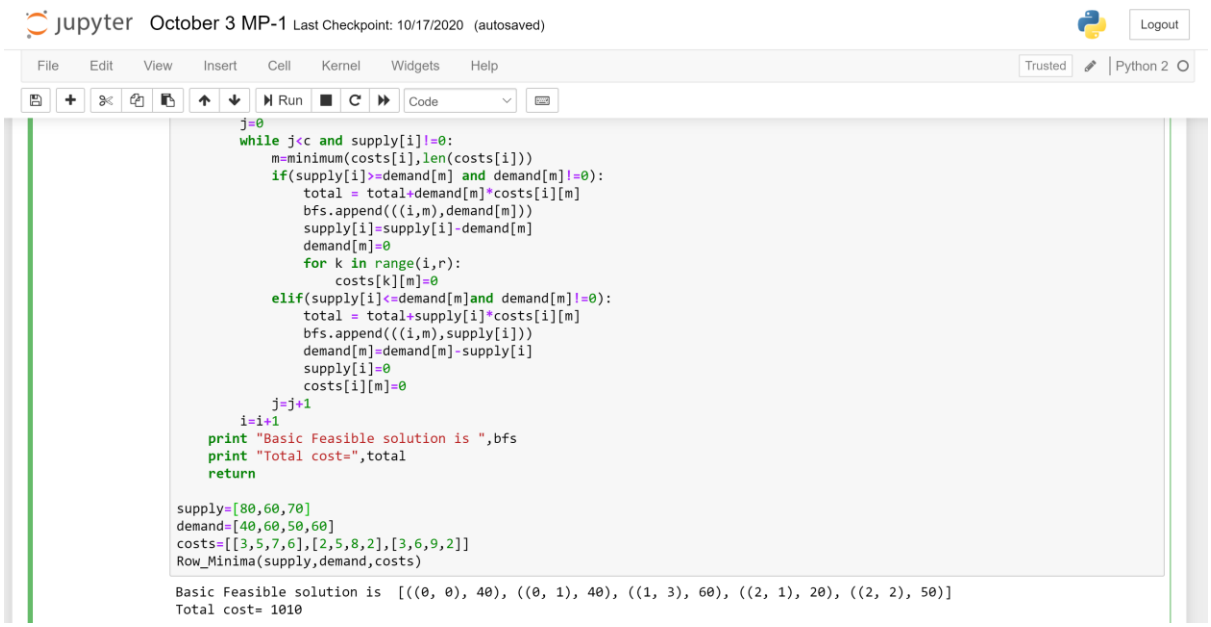
$120 + 200 + 120 + 450 \times 120$

1010

Code:

```
def minimum(a,n):
   minpos = a.index(min(i for i in a if i>0))
   return minpos
def Row_Minima(supply,demand,costs):
   r=len(costs)
   c=len(costs[0])
   i=0
   j=0
   bfs=[]
   total=0
   while i<r:
      j=0
      while j<c and supply[i]!=0:
         m=minimum(costs[i],len(costs[i]))
         if(supply[i]>=demand[m] and demand[m]!=0):
            total = total+demand[m]*costs[i][m]
            bfs.append(((i,m),demand[m]))
            supply[i]=supply[i]-demand[m]
            demand[m]=0
            for k in range(i,r):
               costs[k][m]=0
         elif(supply[i]<=demand[m]and demand[m]!=0):
            total = total+supply[i]*costs[i][m]
            bfs.append(((i,m),supply[i]))
            demand[m]=demand[m]-supply[i]
            supply[i]=0
            costs[i][m]=0
         j=j+1
      i=i+1
   print "Basic Feasible solution is ",bfs
   print "Total cost=",total
   return

supply=[80,60,70]
demand=[40,60,50,60]
costs=[[3,5,7,6],[2,5,8,2],[3,6,9,2]]
Row_Minima(supply,demand,costs)
```


jupyter October 3 MP-1 Last Checkpoint: 10/17/2020 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 2
Run Code
j=0
while j<c and supply[i]!=0:
m=minimum(costs[i],len(costs[i]))
if(supply[i]>=demand[m] and demand[m]!=0):
total = total+demand[m]*costs[i][m]
bfs.append(((i,m),demand[m]))
supply[i]=supply[i]-demand[m]
demand[m]=0
for k in range(i,r):
costs[k][m]=0
elif(supply[i]<=demand[m]and demand[m]!=0):
total = total+supply[i]*costs[i][m]
bfs.append(((i,m),supply[i]))
demand[m]=demand[m]-supply[i]
supply[i]=0
costs[i][m]=0
j=j+1
i=i+1
print "Basic Feasible solution is ",bfs
print "Total cost=",total
return
supply=[80,60,70]
demand=[40,60,50,60]
costs=[[3,5,7,6],[2,5,8,2],[3,6,9,2]]
Row_Minima(supply,demand,costs)
Basic Feasible solution is [((0, 0), 40), ((0, 1), 40), ((1, 3), 60), ((2, 1), 20), ((2, 2), 50)]
Total cost= 1010

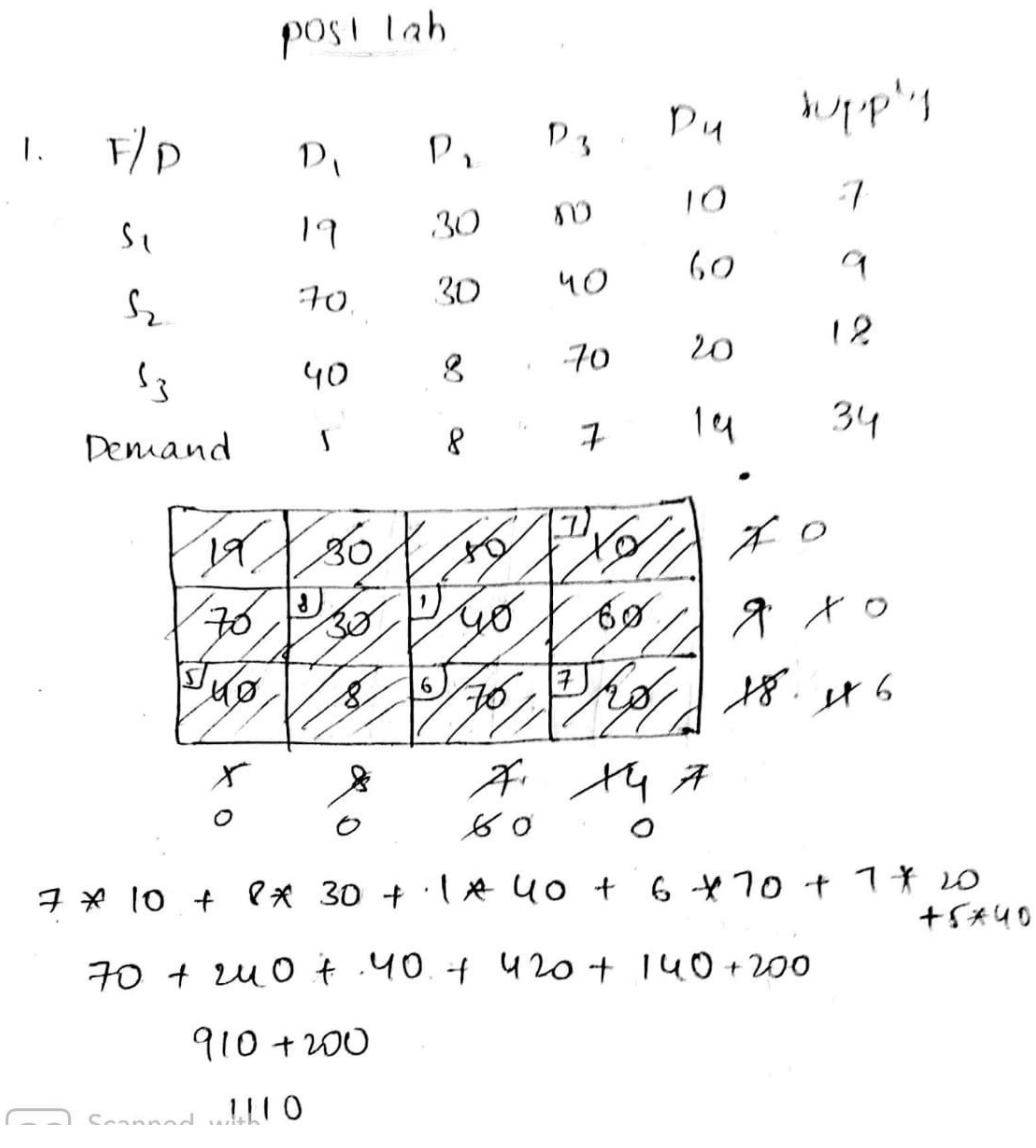
190031187 **NERELLA VENKATA RADHAKRISHNA**

POSTLAB

Problem 1:

The distribution manager of a company needs to minimize global transport costs between a set of three factories (supply points) S1, S2, and S3, and a set of four distributors (demand points) D1, D2, D3, and D4. The following table shows the transportation cost from each supply point to every demand point, the supply of the product at the supply points, and the demand of the product at the demand points

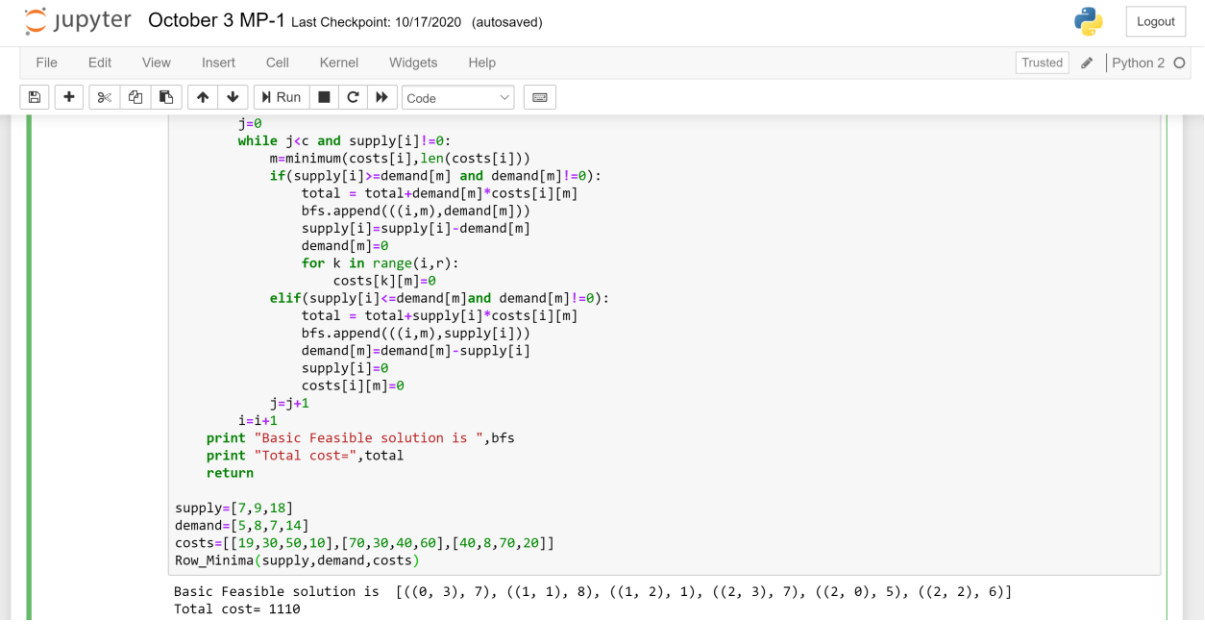
F/D	D1	D2	D3	D4	Supply
S1	19	30	50	10	7
S2	70	30	40	60	9
S3	40	8	70	20	18
Demand	5	8	7	14	34



Code:

```
def minimum(a,n):
    minpos = a.index(min(i for i in a if i>0))
    return minpos
def Row_Minima(supply,demand,costs):
    r=len(costs)
    c=len(costs[0])
    i=0
    j=0
    bfs=[]
    total=0
    while i<r:
        j=0
        while j<c and supply[i]!=0:
            m=minimum(costs[i],len(costs[i]))
            if(supply[i]>=demand[m] and demand[m]!=0):
                total = total+demand[m]*costs[i][m]
                bfs.append(((i,m),demand[m]))
                supply[i]=supply[i]-demand[m]
                demand[m]=0
                for k in range(i,r):
                    costs[k][m]=0
            elif(supply[i]<=demand[m]and demand[m]!=0):
                total = total+supply[i]*costs[i][m]
                bfs.append(((i,m),supply[i]))
                demand[m]=demand[m]-supply[i]
                supply[i]=0
                costs[i][m]=0
            j=j+1
        i=i+1
    print "Basic Feasible solution is ",bfs
    print "Total cost=",total
    return

supply=[7,9,18]
demand=[5,8,7,14]
costs=[[19,30,50,10],[70,30,40,60],[40,8,70,20]]
Row_Minima(supply,demand,costs)
```


jupyter October 3 MP-1 Last Checkpoint: 10/17/2020 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 2
Run Code
j=0
while j<c and supply[i]!=0:
m=minimum(costs[i],len(costs[i]))
if(supply[i]>=demand[m] and demand[m]!=0):
total = total+demand[m]*costs[i][m]
bfs.append(((i,m),demand[m]))
supply[i]=supply[i]-demand[m]
demand[m]=0
for k in range(i,r):
costs[k][m]=0
elif(supply[i]<=demand[m]and demand[m]!=0):
total = total+supply[i]*costs[i][m]
bfs.append(((i,m),supply[i]))
demand[m]=demand[m]-supply[i]
supply[i]=0
costs[i][m]=0
j=j+1
i=i+1
print "Basic Feasible solution is ",bfs
print "Total cost=",total
return
supply=[7,9,18]
demand=[5,8,7,14]
costs=[[19,30,50,10],[70,30,40,60],[40,8,70,20]]
Row_Minima(supply,demand,costs)
Basic Feasible solution is [((0, 3), 7), ((1, 1), 8), ((1, 2), 1), ((2, 3), 7), ((2, 0), 5), ((2, 2), 6)]
Total cost= 1110