

# FALL SEM – (2020-21)

## MAT2003

**SUBMITTED BY: SRIHARSHITHA DEEPALA**

**REG NO : 19BCD7246**

**LAB NO : 9**

**SLOT : L6**

### QUESTION:

**Example 9.2** Using the following cost matrix, determine (a) optimal job assignment (b) the cost of assignments.

		Job				
		1	2	3	4	5
Mechanic	A	10	3	3	2	8
	B	9	7	8	2	7
	C	7	5	6	2	4
	D	3	5	8	2	4
	E	9	10	9	6	10

### CODE:

```
function [assignment,cost] = Assign_prob(costMat)
costMat = input("enter the cost matrix")

assignment = zeros(1,size(costMat,1));
cost = 0;
validMat = costMat == costMat & costMat < Inf;
bigM = 10^(ceil(log10(sum(costMat(validMat))))+1);
costMat(~validMat) = bigM;
validCol = any(validMat,1);
validRow = any(validMat,2);

nRows = sum(validRow);
nCols = sum(validCol);
n = max(nRows,nCols);
if ~n
    return
end

maxv=10*max(costMat(validMat));
```

```

dMat = zeros(n) + maxv;
dMat(1:nRows,1:nCols) = costMat(validRow,validCol);
minR = min(dMat,[],2);
minC = min(bsxfun(@minus, dMat, minR));
zP = dMat == bsxfun(@plus, minC, minR);

```

```

starZ = zeros(n,1);
while any(zP(:))
    [r,c]=find(zP,1);
    starZ(r)=c;
    zP(r,:)=false;
    zP(:,c)=false;
end

```

```

while 1
    if all(starZ>0)
        break
    end
    coverColumn = false(1,n);
    coverColumn(starZ(starZ>0))=true;
    coverRow = false(n,1);
    primeZ = zeros(n,1);
    [rIdx, cIdx] =
find(dMat(~coverRow,~coverColumn)==bsxfun(@plus,minR(~coverRow),minC(~
coverColumn)));
    while 1

```

```

cR = find(~coverRow);
cC = find(~coverColumn);
rIdx = cR(rIdx);
cIdx = cC(cIdx);
Step = 6;
while ~isempty(cIdx)
    uZr = rIdx(1);
    uZc = cIdx(1);
    primeZ(uZr) = uZc;
    stz = starZ(uZr);
    if ~stz
        Step = 5;
        break;
    end
    coverRow(uZr) = true;
    coverColumn(stz) = false;
    z = rIdx==uZr;
    rIdx(z) = [];
    cIdx(z) = [];
    cR = find(~coverRow);
    z = dMat(~coverRow,stz) == minR(~coverRow) + minC(stz);
    rIdx = [rIdx(:);cR(z)];
    cIdx = [cIdx(:);stz(ones(sum(z),1))];

```

```

end
if Step == 6
[minval,rIdx,cIdx]=outerplus(dMat(~coverRow,~coverColumn),minR(~coverRow),minC(~coverColumn));
minC(~coverColumn) = minC(~coverColumn) + minval;
minR(coverRow) = minR(coverRow) - minval;
else
break
end
end
rowZ1 = find(starZ==uZc);
starZ(uZr)=uZc;
while rowZ1>0
starZ(rowZ1)=0;
uZc = primeZ(rowZ1);
uZr = rowZ1;
rowZ1 = find(starZ==uZc);
starZ(uZr)=uZc;
end
end

```

```

rowIdx = find(validRow);
colIdx = find(validCol);
starZ = starZ(1:nRows);
vIdx = starZ <= nCols;
assignment(rowIdx(vIdx)) = colIdx(starZ(vIdx));
pass = assignment(assignment>0);
pass(~diag(validMat(assignment>0,pass))) = 0;
assignment(assignment>0) = pass;
cost = trace(costMat(assignment>0,assignment(assignment>0)));
disp('Displaying the assignment:')
disp(assignment)
disp('Displaying total cost:')
disp(cost)
function [minval,rIdx,cIdx]=outerplus(M,x,y)
ny=size(M,2);
minval=inf;
for c=1:ny
M(:,c)=M(:,c)-(x+y(c));
minval = min(minval,min(M(:,c)));
end
[rIdx,cIdx]=find(M==minval);

```

## OUTPUT:

```
>> Assign_prob  
enter the cost matrix  
[10 3 3 2 8; 9 7 8 2 7; 7 5 6 2 4; 3 5 8 2 4; 9 10 9 6 10]
```

```
costMat =
```

10	3	3	2	8
9	7	8	2	7
7	5	6	2	4
3	5	8	2	4
9	10	9	6	10

```
Displaying the assignment:
```

2	4	5	1	3
---	---	---	---	---

```
Displaying total cost:
```

```
21
```

```
ans =
```

2	4	5	1	3
---	---	---	---	---