```
clc
clear all
format short
% LEAST COST METHOD
%% Input Phase
Cost=[11 20 7 8; 21 7 10 12; 8 12 18 9]
A=[50 40 70]
B=[30 25 35 40]
%% to check unbalanced/ balanced problem
if sum(A)==sum(B)
    fprintf ('Given Transportation Problem is Balanced')
else
    fprintf('given tansportation probelm is unbalanced \n')
    if sum(A)<sum(B)
        Cost(end+1,:)=zeros(1,size(B,2))
        A(end+1)=sum(B)-sum(A)
    elseif sum(B)<sum(A)
         Cost(:,end+1)=zeros(1,size(A,2))
        B(end+1)=sum(A)-sum(B)
    end
end
ICost= Cost
X=zeros(size(Cost)) % Initialize allocation
[m,n ]=size(Cost) %Finding No. of rows and columns
BFS=m+n-1
%% finding the cell(with minimum cost ) for the allocations
for i=1:size(Cost,1)
    for j=1:size(Cost,2)
        hh=min(Cost(:)) % finding the minimum cost value
        [Row_index,Col_index]=find(hh==Cost) %finding the position of the
min Cost cell
        x11=min(A(Row_index),B(Col_index))
        [Value,index]=max(x11) % find the max allocation
        ii=Row_index(index) %identify row position
        jj=Col_index(index) %identify row position
        y11=min(A(ii),B(jj))
        X(ii,jj)=y11
        A(ii)=A(ii)-y11
        B(jj)=B(jj)-y11
        Cost(ii,jj)=Inf
    end
end
%% Print the initial bfs
fprintf('initial bfs=\n')
%% Check the degenerate and non degenerate
TotalBFS=length(nonzeros(X))
if TotalBFS==BFS
    fprintf('Initial BFS is non degenerate \n')
else
    fprintf('Initial BFS is degenerate \n')
end
%% Compute the Initial Transportation Cost
InitialCost=sum(sum(ICost.*X))
fprintf('the initial BFS Cost is =%d ',InitialCost)
```