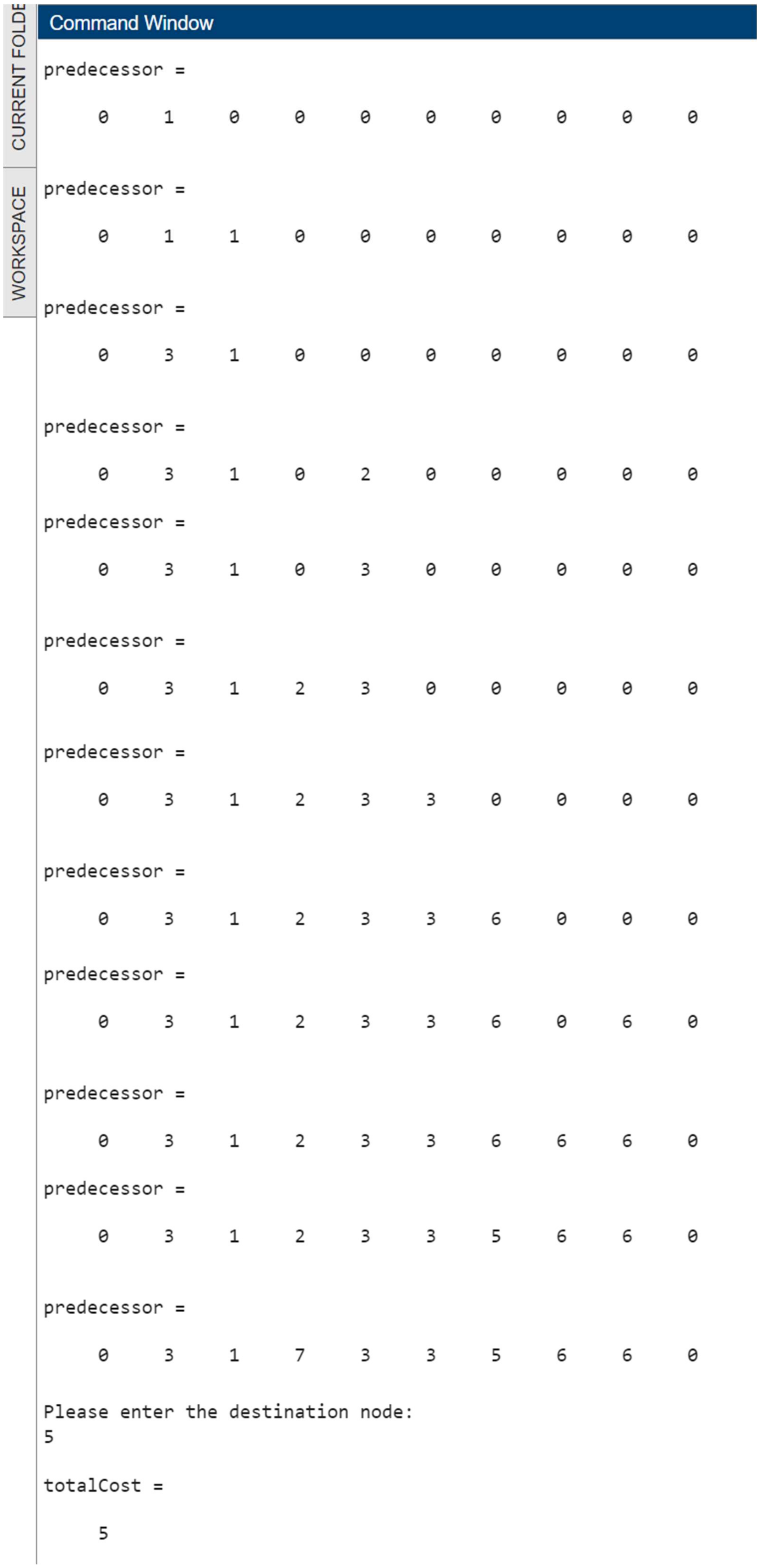
MAT2003 (Optimization Techniques) Lab-10

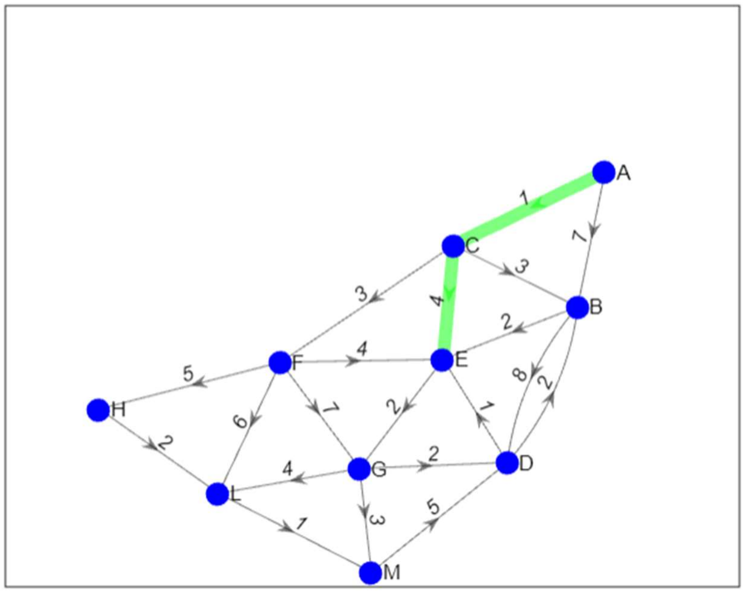
KHAN MOHD OWAIS RAZA
20BCD7138

Ford-Fulkerson Method

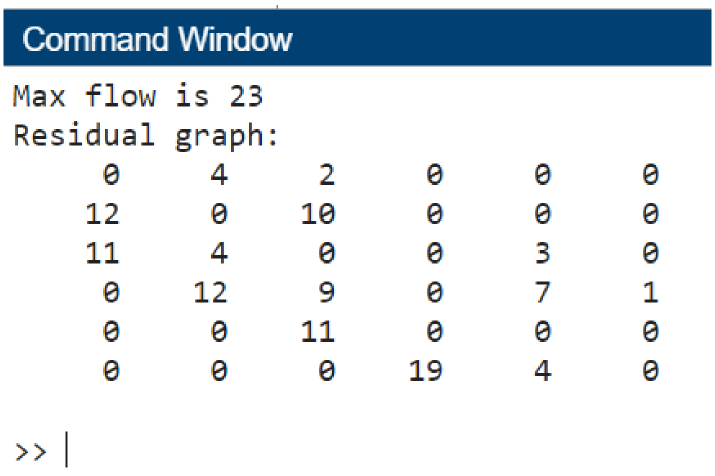
```
%% KHAN MOHD OWAIS RAZA (20BCD7138)
%% MAT2003 (OT) Lab
clc;
clear all;
close all;
N=10;       % number of nodes
e_num=20;   % number of edges in the network
s=1;        % Initialization of source node
w=[7 1 3 2 4 8 2 1 3 4 7 6 5 2 2 3 4 2 1 5];
m=[1 1 3 2 3 2 4 4 3 6 6 6 6 5 7 7 7 8 9 10];
n=[2 3 2 5 5 4 2 5 6 5 7 9 8 7 4 10 9 9 10 4];
names={'A','B','C','D','E','F','G','H','L','M'};
G=digraph(m,n,w)
h=plot(G,'EdgeLabel',G.Edges.Weight,'Nodelabel',names,'EdgeColor','k','NodeColor',
'b')
h.MarkerSize=8;
S=sparse(m',n',w');
distance(1:N)=Inf;   % distance of each node initialized to infinity
distance(s)=0;       % distance of source node intitalized to 0
predecessor(1:N)=0;
for i = 1 : N - 1
    for j = 1 : e_num
      v = n(j);
      u = m(j);
      t = distance(u) + w(j);
      if (t < distance(v) )
        distance(v) = t;
        predecessor(v) = u
      end
    end
  end
for j = 1 : e_num
    u = m(j);
    v = n(j);
    if ( distance(u) + w(j) < distance(v) )
      fprintf ( 1, '\n' );
      fprintf ( 1, 'BELLMAN_FORD - Fatal error!\n' );
      fprintf ( 1, '  Graph contains a cycle with negative weight.\n' );
      error ( 'BELLMAN_FORD - Fatal error!' );
    end
  end
for i=1:(N-1)
d=input('Please enter the destination node:');
totalCost = distance(d)
TR=shortestpathtree(G,1,d);
p=plot(G,'EdgeLabel',G.Edges.Weight,'Nodelabel',names,'EdgeColor','k','NodeColor',
'b')
p.MarkerSize=8;
highlight(p,TR,'EdgeColor','g','LineWidth',5);
end
```



```
predecessor =

     0     1     0     0     0     0     0     0     0     0


predecessor =

     0     1     1     0     0     0     0     0     0     0


predecessor =

     0     3     1     0     0     0     0     0     0     0


predecessor =

     0     3     1     0     2     0     0     0     0     0

predecessor =

     0     3     1     0     3     0     0     0     0     0


predecessor =

     0     3     1     2     3     0     0     0     0     0


predecessor =

     0     3     1     2     3     3     0     0     0     0


predecessor =

     0     3     1     2     3     3     6     0     0     0

predecessor =

     0     3     1     2     3     3     6     0     6     0


predecessor =

     0     3     1     2     3     3     6     6     6     0

predecessor =

     0     3     1     2     3     3     5     6     6     0


predecessor =

     0     3     1     7     3     3     5     6     6     0

Please enter the destination node:
5

totalCost =

     5
```


A
B
C
D
E
F
G
H
L
M
1
7
3
3
4
2
4
5
6
7
2
1
8
2
2
2
4
3
1
5

```
%% KHAN MOHD OWAIS RAZA (20BCD7138)
%% MAT2003 (OT) Lab
%% Bellmond Ford Method
function Bellmond_Ford
 clc;
 clear;
 s = 1; t = 6; f = 0;
 cap = [ 0 16 13 0 0 0;
 0 0 10 12 0 0;
 0 4 0 0 14 0;
 0 0 9 0 0 20;
 0 0 0 7 0 4;
 0 0 0 0 0 0];
 len = length(cap);
 while true
 p = findPath(cap);
 if p(1) == 0, break; end
 flow = max(max(cap));
 for j = 2:length(p)
 flow = min(flow,cap(p(j),p(j-1)));
 end
 for j = 2:length(p)
 a = p(j); b = p(j-1);
 cap(a,b) = cap(a,b) - flow;
 cap(b,a) = cap(b,a) + flow;
 end
 f = f + flow;
 end
 disp(['Max flow is ' num2str(f)]);
 disp('Residual graph:');
 disp(cap);
 function F = findPath(A)
 q = zeros(1,len);
 pred = zeros(1,len);
 front = 1; back = 2;
 pred(s) = s; q(front) = s;
 while front ~= back
 v = q(front);
 front = front + 1;
 for i = 1:len
 if pred(i) == 0 && A(v,i) > 0
 q(back) = i;
back = back + 1;
pred(i) = v;
 end
 end
 end
 path = zeros(1,len);
 if pred(t) ~= 0
 i = t; c = 1;
 while pred(i) ~= i
 path(c) = i;
 c = c + 1;
 i = pred(i);
 end
 path(c) = s;
 path(c+1:len) = [];
 end
 F = path;
 end
end
```



```
Max flow is 23
Residual graph:
     0     4     2     0     0     0
    12     0    10     0     0     0
    11     4     0     0     3     0
     0    12     9     0     7     1
     0     0    11     0     0     0
     0     0     0    19     4     0

>> 
```