MATLAB PROGRAM FOR MINIMUM WEIGHTED SPANNING TREE USING DOMINATING SET

M. Yamuna[1] and K. Karthika[2]

[1,2]School of Advanced Sciences, VIT University, Vellore, (India)

ABSTRACT

Finding a minimum weighted spanning tree is an important problem as it has many applications. Finding minimum weighted spanning tree using graph properties is not in much use. In this paper we provide a MATLAB Programme, for generating a minimum weighted spanning tree using adjacency matrix.

Keywords: Adjacency matrix, Dominating set, Graph domination, Spanning tree

I INTRODUCTION

In the 1950's, many people contributed to the minimum spanning tree problem. Among them were R. C. Prim and J. B. Kruskal, whose algorithms are very widely used today. Prims algorithm grows a spanning tree from a given tree from a given vertex of a connected weighted graph G, iteratively adding the cheapest edge from a vertex already reached to a vertex not yet reached, finishing when all the vertices of *G* have been reached [1]. Kruskal's algorithm maintain an acyclic spanning subgraph H, enlarging it by edges with low weight to form a spanning tree, by considering edges in non decreasing order of weight, breaking ties arbitrarily [2].

In [3], M. Yamuna et al have provided a procedure for generating a minimum weighted spanning tree by using minimum domination set and adjacency matrix. In this paper we provide a MATLAB Programme, for generating a minimum weighted spanning tree using adjacency matrix.

II MATERIALS AND METHODS

A spanning tree of G is a subgraph of G that is a tree containing every vertex of G. A spanning forest of a graph G is forest that contains every vertex of G such that two vertices are in the same tree of the forest when there is a path in G between these two vertices.

A graph G is said to be a weighted graph if its edges are assigned some weight. A minimum weighted spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.

An adjacency matrix of a graph G with n vertices that are assumed to be ordered from v_1 to v_n is defined by,

$$A = [\,aij\,]_{n \times n} = \begin{cases} 1 & \text{if there exist an edge between } v_i \text{ and } v_j \\ 0 & \text{otherwise} \end{cases}$$

Adjacency matrix is also used to represent weighted graphs. If $[\,a_{ij}\,] = w$, then there is an edge from vertex v_i to vertex v_j with weight w. For properties related to graph theory we refer to [4].

A set of vertices D in a graph G = (V, E) is a dominating set if every vertex of V − D is adjacent to some vertex of D. If D has the smallest possible cardinality of any dominating set of G, then D is called a minimum dominating set. The cardinality of any minimum dominating set for G is called the domination number of G and it is denoted by $\gamma (G)$. γ – set denotes a dominating set for G with minimum cardinality.The open neighborhood of vertex $v \in V (G)$ is denoted by $N (v) = \{ u \in V (G) \mid (uv) \in E (G) \}$, while its closed neighborhood is the set $N [v] = N(v) \cup \{ v \}$. The private neighborhood of $v \in D$ is denoted by $pn [v, D]$, is defined by $pn [v, D] = N (v) – N (D - \{ v \})$. For properties related to domination theory we refer to [5].

Graph Domination

In [3], M. Yamuna et al have defined a γ - set $D \in V$ as a graph domination set if D covers all the vertices and edges of G. A γ – set D of G that satisfies this property is denoted by $\gamma_G(G)$.

In all the graphs circled vertices represent a γ - set D.

Example

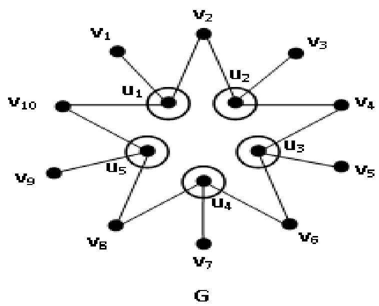


Fig. 1

The graph G in Fig. 1 is a graph domination graph. The γ – set $\{ u_1, u_2, u_3, u_4, u_5 \}$ covers all the edges and vertices of G.

In [3], they have proved the following result

R1. Let G be a weighted connected graph with n vertices and D be a γ - set for G that covers all the edges of G. Then there is a spanning tree T for G such that

1. $\gamma (G) = \gamma (T)$.
2. T is minimum weighted.

Given any weighted graph G a minimum weighted spanning tree can be obtained using the following steps.

Step 1 Consider the vertices in D, where D is a graph dominating γ - set for G.

Step 2 Include the private neighbor of every vertex in D.

Step 3 Include the 2 dominated vertices (taking care to pick the edge with smallest weight).

Step 4 Combine the spanning forest into a spanning tree by including an edge with minimum weight each time.

Spanning tree generation using the above steps is given in the following example for the graph in Fig. 2.

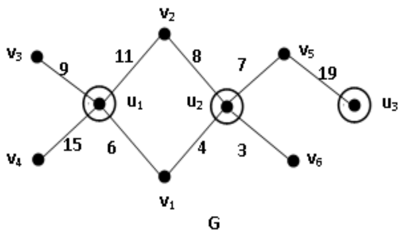


Fig. 2

By using R1 and the steps, we can generate a minimum weighted spanning tree from a weighted graph.

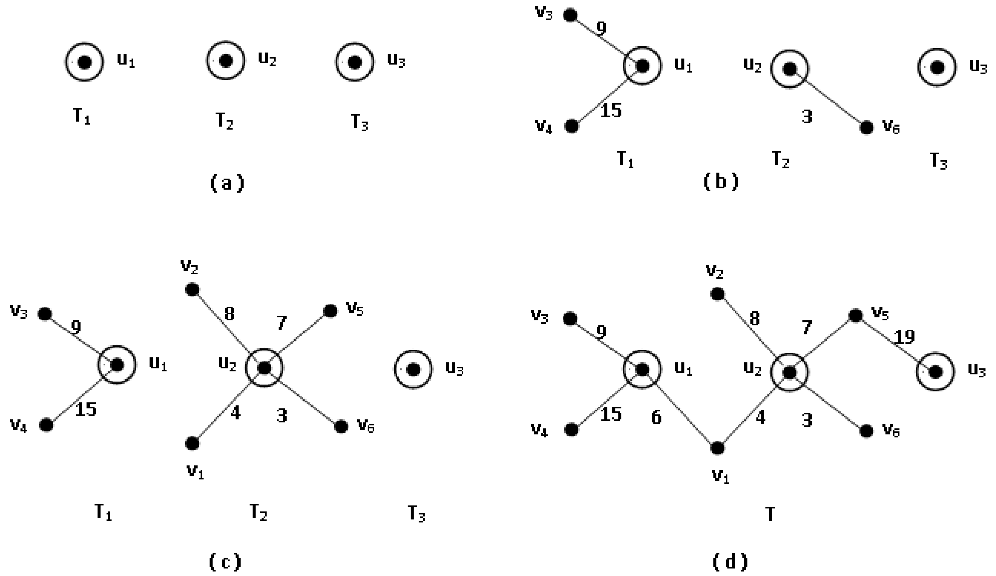


Fig. 3

In Fig. 3,

(a) The graph represents the spanning forest that contain vertices in D = { u_1, u_2, u_3 } only.

(b) The tree partially constructed by adding minimum weighted edges belonging to pn[u_1, D] and pn [u_2, D], where pn [u_1, D] = { v_3, v_4 }, pn [u_2, D] = { v_6 }.

(c) The tree partially constructed by adding minimum weighted edges belonging to 2 – dominated vertices.

(d) Minimum weighted spanning tree generated from D. The weight of this spanning tree is equal to 71.

Adjacency Matrix Using γ - set

In this procedure from [3], G is a weighted connected, graph domination graph with n vertices. Let D = {u_1, u_2, . . . , u_k }, and V − D = {v_1, v_2, . . . , v_m}, k + m = n. Let X be the adjacency matrix of G. For comfort of discussion, let us arrange the rows and columns of X as follows

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$$

where X11 represents adjacency between vertices in D, X12 and X21 between vertices in D and V – D and X22 between vertices in V − D. In a graph domination graph, since V − D is independent, X22 is a null matrix. Using these notations and procedure we have developed the MATLAB code.

MATLAB Code for Generating Minimum Weighted Spanning Tree

Using the above explained procedure and adjacency matrix X we have developed a MATLAB program for the same. The coding part of the program is given inn snapshots 1,2.

C:\Documents and Settings\Administrator\Desktop\mat sp\spexamplee1.m*

```
X=input('Enter the adjacency matrix X = ')
k=input('Enter the gamma value k = ');
n=size(X, 2);
m=n-k;
X11 = X(1:k, 1:k);
X12 = X(1:k, k+1:n);
X21=X(k+1:n, 1:k);
X22=X(k+1:end, k+1:end);
A=X12
A(~A) = inf;
[A1, A2] = min (A)
A3=[1:m];
A4=[A3;A2];
for  k1=1:m
    a2=A2(:, k1);
    a3=A3(:,k1);
    X12(a2, a3) = 0;
end
X12;
X = [X11 X12];
X2 = [X12' X22];
X=[X; X2];
B=[1:n];
B1=[1:k];
X1 = [B;X]
X2 = X1(1 ,:);
B=[];
z=[];
F=[];
    for R=randperm(k)
R= input('Enter the row number randomly: R = ')
A5 = A4(:, A4(2 ,:)==R)
if isempty(A5)
    break;
end
A6 = A5(2,:);
Ac=A6(:, 1);
A7 = A5(1, :)
```

Snapshot 1

```
n=size(A7,2);
A8 = A7+k;
A9 = [Ac A8]
z=[ z A9];
for i=A9
X1(:, X1(1 ,:)==i)=[];
end
if isempty(X1)
    break;
end
X5=X1(z+1,:)
X5(~X5) = inf;
[R1, C1]=min(X5(:))
if R1==inf
    R1=0
end
B=[B R1];
[r,c]=find(X5==min(min(X5)));
if size(c,1)==1
X6=X1(:,c);
X7=X6(1,:)
else size(c,1)~=1
c=c(c<k);
[a1 a2]=unique(c(:,1))
end
z1=z(:,r);
X1(z1+1,c)=0
A1
B
B1=[A1 B]
S1=sum(B1)
X7
if c==1
    disp('Choose any row from A:')
else if X7 > k
      disp('Choose any row from A:')
  else
      disp('Choose the row value as the value of X7:')
  end
end
end
disp('The weight of spanning tree is =')
S1
```

Snapshot 2

III RESULTS AND DISCUSSION

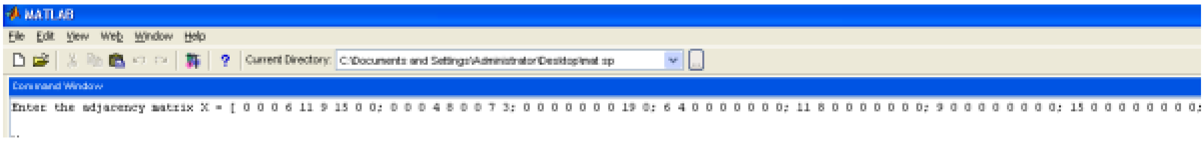
Output Verification

In this section we provide verification for few graphs using the above coding.

Example 1

We consider the adjacency matrix of Fig. 2. and verify the weight of spanning tree.

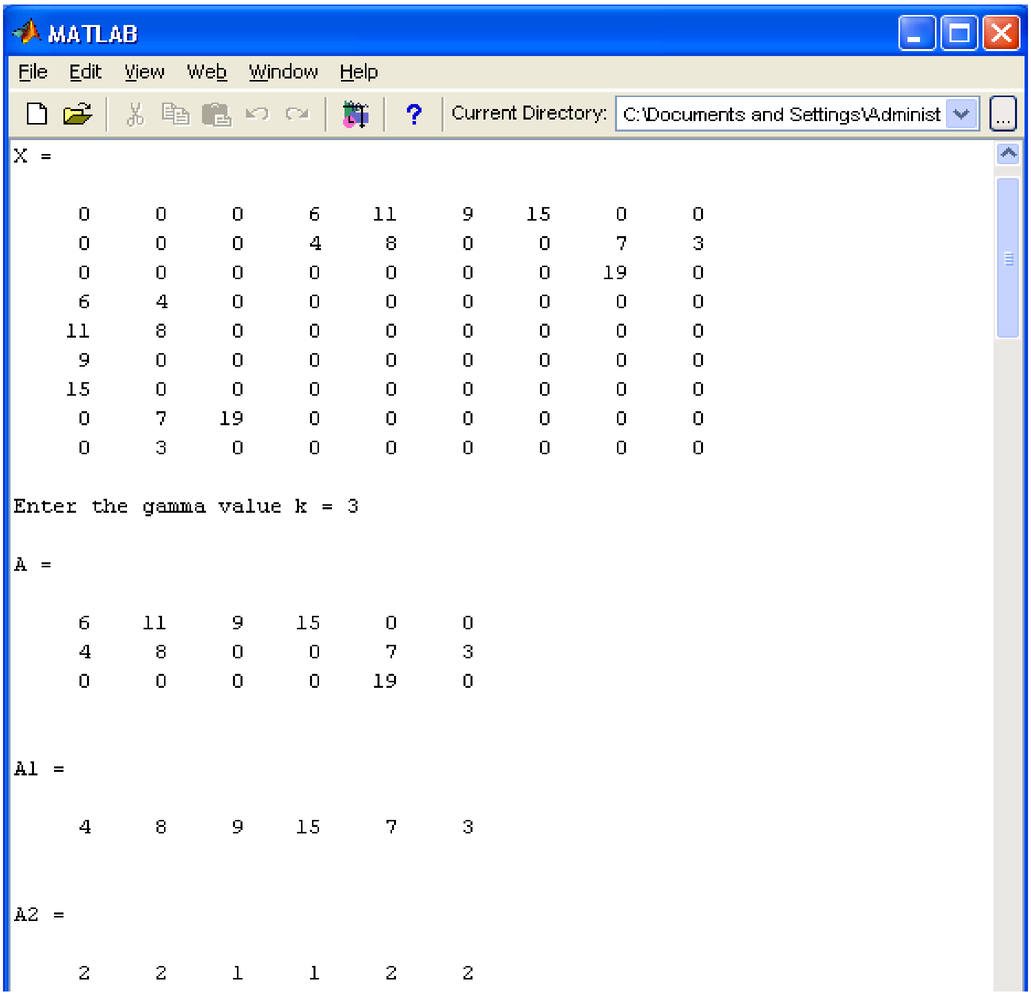
Snap shot 3 provides the input matrix for the graph in Fig. 2.



```
Enter the adjacency matrix X = [ 0 0 0 6 11 9 15 0 0; 0 0 0 4 8 0 0 7 3; 0 0 0 0 0 0 0 19 0; 6 4 0 0 0 0 0 0 0; 11 8 0 0 0 0 0 0 0; 9 0 0 0 0 0 0 0 0; 15 0 0 0 0 0 0 0 0;
```

Snapshot 3

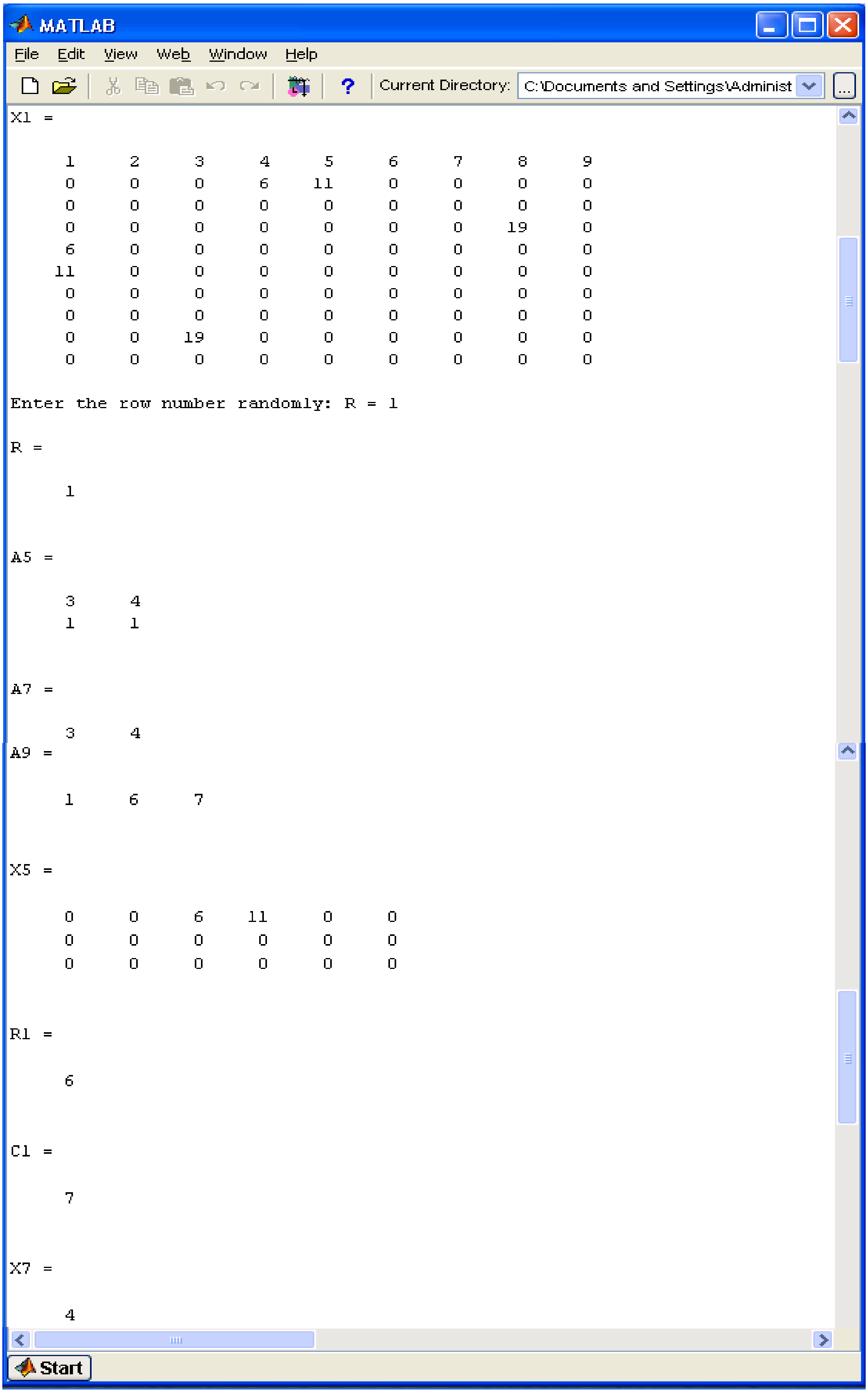
Snapshot 4 provides the out put matrix for the matrix in snapshot 3. Here the cardinality of the γ - value is 3. So when k = 3, the submatrix A generated is also seen in snapshot 4.



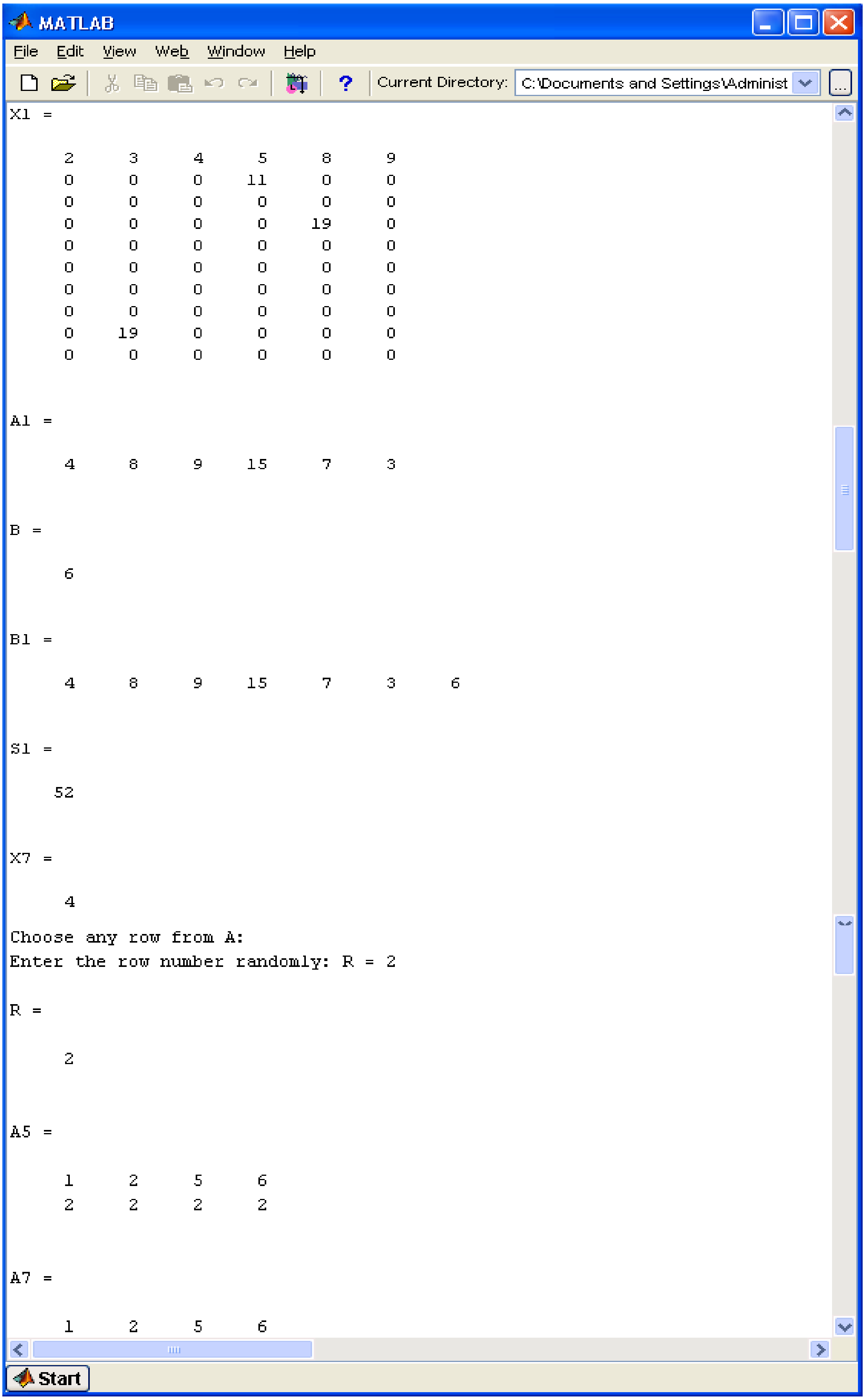
```
X =

     0     0     0     6    11     9    15     0     0
     0     0     0     4     8     0     0     7     3
     0     0     0     0     0     0     0    19     0
     6     4     0     0     0     0     0     0     0
    11     8     0     0     0     0     0     0     0
     9     0     0     0     0     0     0     0     0
    15     0     0     0     0     0     0     0     0
     0     7    19     0     0     0     0     0     0
     0     3     0     0     0     0     0     0     0

Enter the gamma value k = 3

A =

     6    11     9    15     0     0
     4     8     0     0     7     3
     0     0     0     0    19     0


A1 =

     4     8     9    15     7     3


A2 =

     2     2     1     1     2     2
```

Snapshot 4

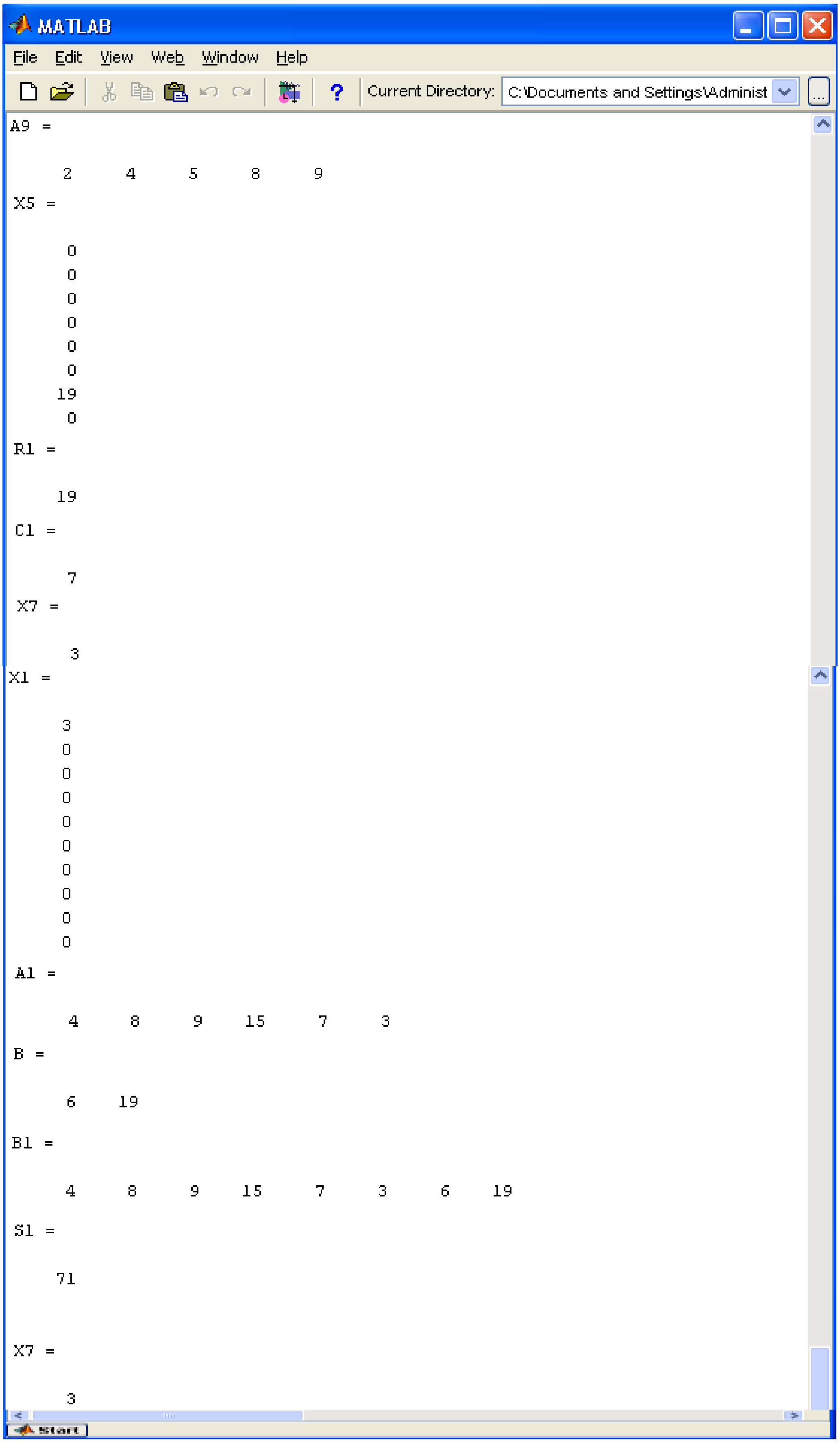
In the following snapshots 5 – 7, Xi represents the reduced matrix obtained in each stage, R represents the row selected in each execution, R1 the minimum value in each stage.


MATLAB
File Edit View Web Window Help
Current Directory: C:\Documents and Settings\Administ
X1 =
1 2 3 4 5 6 7 8 9
0 0 0 6 11 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 19 0
6 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 19 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Enter the row number randomly: R = 1
R =
1
A5 =
3 4
1 1
A7 =
3 4
A9 =
1 6 7
X5 =
0 0 6 11 0 0
0 0 0 0 0 0
0 0 0 0 0 0
R1 =
6
C1 =
7
X7 =
4
Start

Snapshot 5

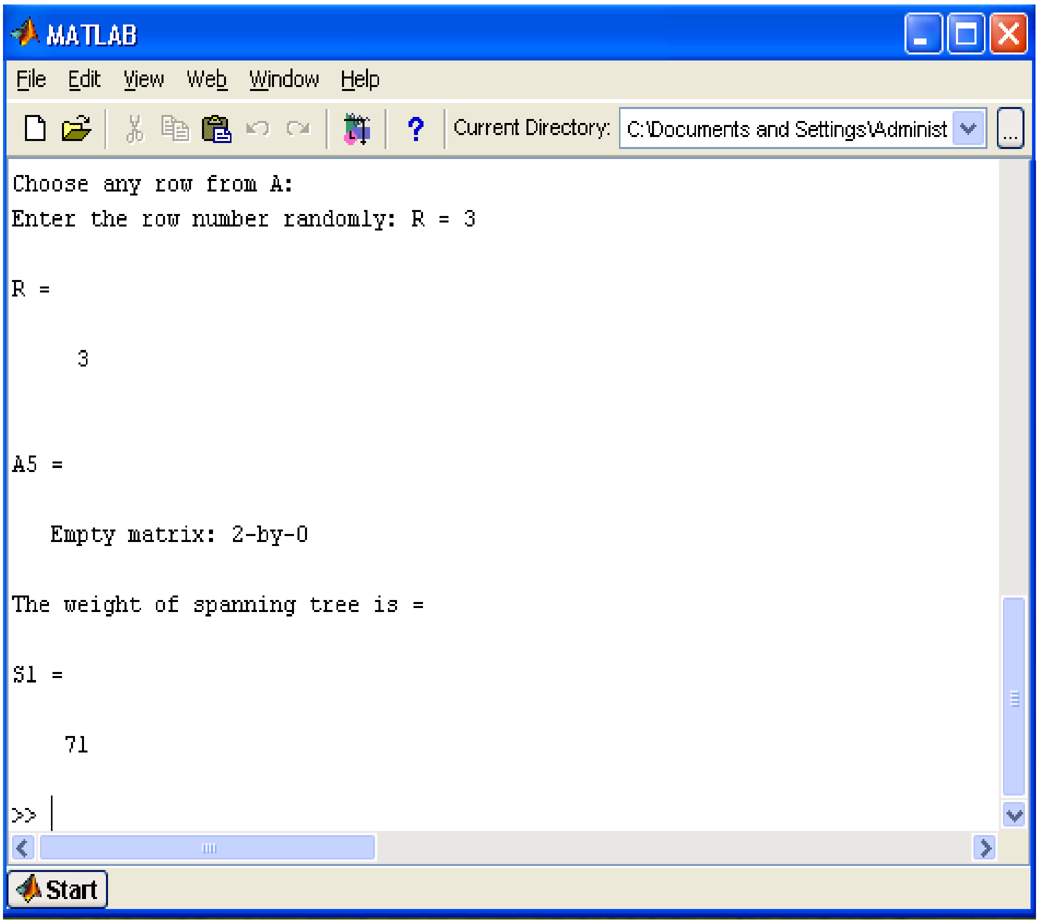

MATLAB
File Edit View Web Window Help
Current Directory: C:\Documents and Settings\Administ
X1 =
2 3 4 5 8 9
0 0 0 11 0 0
0 0 0 0 0 0
0 0 0 0 19 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 19 0 0 0 0
0 0 0 0 0 0
A1 =
4 8 9 15 7 3
B =
6
B1 =
4 8 9 15 7 3 6
S1 =
52
X7 =
4
Choose any row from A:
Enter the row number randomly: R = 2
R =
2
A5 =
1 2 5 6
2 2 2 2
A7 =
1 2 5 6
Start

Snapshot 6


MATLAB
File Edit View Web Window Help
Current Directory: C:\Documents and Settings\Administ
A9 =
2 4 5 8 9
X5 =
0
0
0
0
0
0
19
0
R1 =
19
C1 =
7
X7 =
3
X1 =
3
0
0
0
0
0
0
0
0
0
A1 =
4 8 9 15 7 3
B =
6 19
B1 =
4 8 9 15 7 3 6 19
S1 =
71
X7 =
3
Start

Snapshot 7

In snapshot 8 the final output is 71, which matches with the weight of the sapnning tree, for the graph in Fig . 2.



Snapshot 8

Example 2

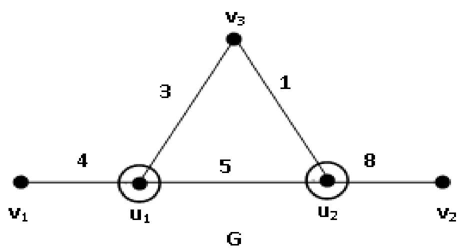
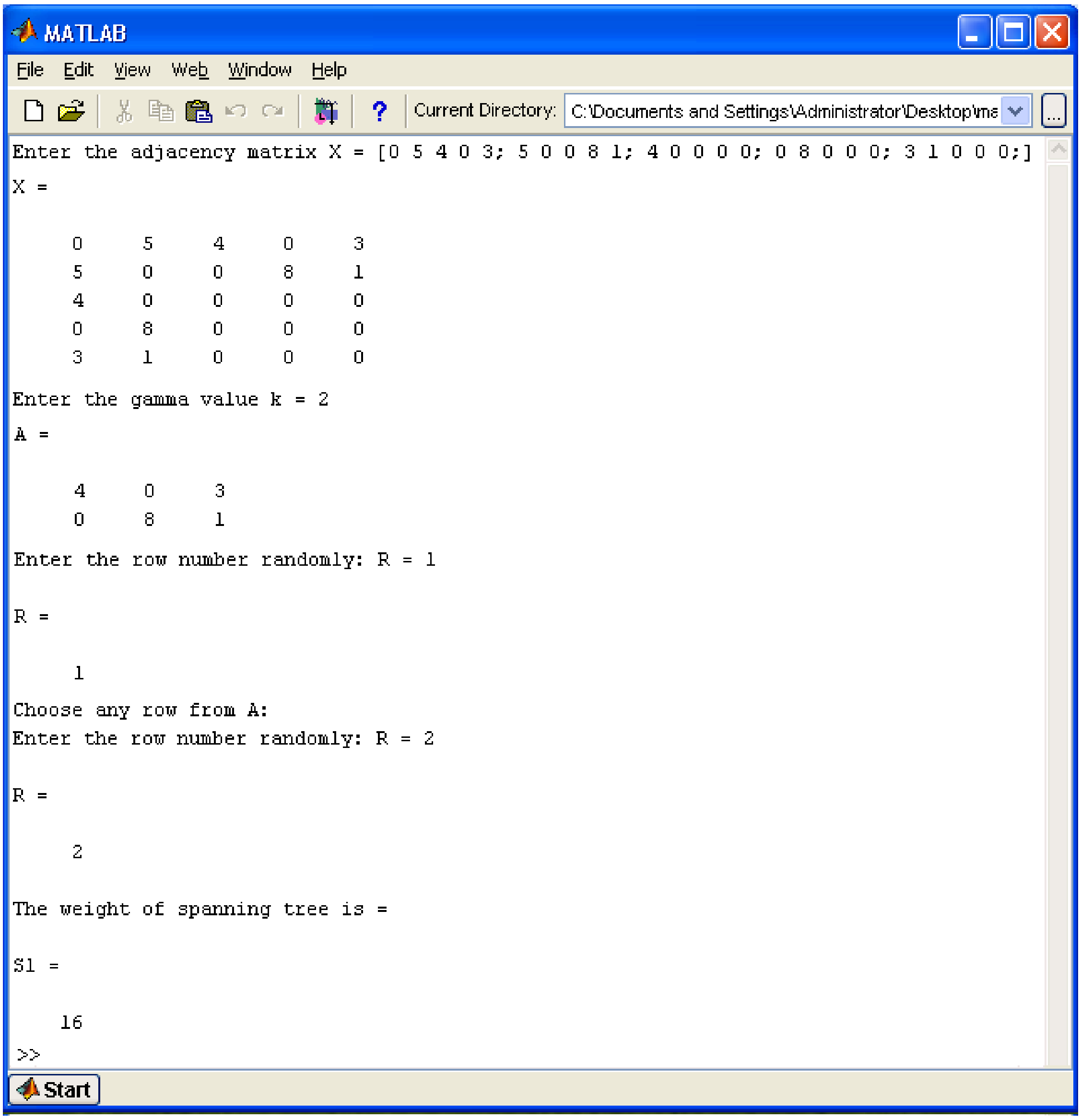


Fig. 4

OUTPUT



Snapshot 9

The minimum weight of the spanning tree for the graph in Fig. 4 is 16, which matches with the value obtained in snapshot 9.

IV CONCLUSION

The above procedure generates the minimum weight of the spanning tree using dominating set. The program developed enables us to calculate the minimum weight of the spanning tree easily since MATLAB is user friendly. We just need to input the adjacency matrix to obtain the minimum weight of the tree. So this procedure can be adopted for generation of spanning trees.

REFERENCES

[1] R. C. Prim. Shortest Connection Networks and Some Generalizations, Bell Syst. Tech. J., 36, 1957, pp.1389 – 1401.

[2] J. B. Kruskal. On the Shortest Spanning Sub tree of a Graph and the Traveling Salesman Problem, Proc. Am. Math. Soc., l.7, 1956, pp.48 – 50.

[3] M. Yamuna and K. Karthika. Minimal Spanning Tree from a Minimum Dominating Set, WSEAS Transactions on Mathematics, 12 (11), 2013, pp. 1055 – 1064.

[4] D. B. West. Introduction to Graph Theory, second ed., Prentice - Hall, New Jersey, 2007.

[5] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. Fundamentals of Domination in Graphs, Marcel Dekker, New York, 1998.