CHAPTER V
CHAPTER V

MATRIX ALGORITHMS

Introduction

There are polynomial time algorithms to find the independence number, packing number and the domination number of a tree. Even then, we make another attempt to develop algorithms to find them using matrices. For this purpose, we define self adjacency matrix of a graph which is similar to the adjacency matrix of a graph.

In this chapter, there are three sections and in each section first we prove some results which are necessary to establish the validity of the algorithms. Subsequently the algorithm and illustrations are given.

§ 5.1 Self adjacency matrix

Definition 5.1.1 Let G be a simple graph of n vertices v₁, v₂,...,vₙ. Then the n×n matrix \( A(G) = [a_{ij}] \) is called the adjacency matrix of G, where \( a_{ij} = 1 \), if \( v_i \) is adjacent to \( v_j \) and \( a_{ij} = 0 \) otherwise.
**Definition 5.1.2** Let $G$ be a simple graph of $n$ vertices $v_1, v_2, \ldots, v_n$. Then the matrix $SA(G) = [a_{ij}]$ is called the **self adjacency** matrix of $G$, where $a_{ij} = 1$, if $v_i$ is adjacent to $v_j$, $a_{ij} = 0$ otherwise for $i \neq j$ and $a_{ii} = 1$ for every $i$.

**Remark 5.1.3**

(i) $SA(G)$ is a square matrix of order $n$ and $SA(G) = A(G) + I(G)$ where $A(G)$ is the adjacency matrix of $G$.

(ii) It is a binary matrix with 0 and 1 as entries.

(iii) The primary diagonal entries of the matrix are one.

(iv) It is a symmetric matrix.

(v) Any row total or column total is the degree of the corresponding vertex plus one.

(vi) For a pendant vertex, the row sum as well as the column sum corresponding to that vertex is 2.

**Algorithm for independence number of a tree**

Before developing an algorithm to find an independence set of a tree, we prove certain results useful for the same.
Theorem 5.1.4 Let $T$ be a tree with $n > 2$. Then there exists a maximum independent set ($\beta_0$-set) containing all the pendant vertices of $T$.

Proof Let $S$ be any $\beta_0$-set of $T$. Let $u$ be any pendant vertex not in $S$. Since $S$ is a dominating set, its support $v$ must be in $S$. Clearly $u$ is a private neighbor of $v$.

We claim that $v$ has only one private neighbor $u$ other than itself. If not, let $w$ be another private neighbor of $v$. Then the set $S_1 = S \setminus \{v\} \cup \{u, w\}$ is an independent set with $|S_1| = |S| + 1$. This is a contradiction to the maximality of $S$. Thus $S$ is an independent set with $v \in S$ and so $N(v)$ contains no pendant vertex other than $u$. Hence the set $S \setminus \{v\} \cup \{u\}$ is an independent set. Continuing this way one can get a $\beta_0$-set containing all the pendant vertices.

Remark 5.1.5 Let $T$ be a tree and $u$ be any pendant vertex. Then there exists a $\beta_0$-set containing the pendant vertex $u$. In general, there exists a $\beta_0$-set containing all the pendant vertices of the tree $T$. Consequently one can start with any pendant vertex to find a $\beta_0$-set.

Remark 5.1.6 If $T_1, T_2, \ldots, T_k$ ($k > 0$) be the $k$ different trees in a forest $F$, then $\beta_0(F) = \beta_0(T_1) + \ldots + \beta_0(T_k)$. 

71
Theorem 5.1.7 Let \( u \) be a pendant vertex of a tree \( T \). Then \( \beta_0(T \setminus N[u]) = \beta_0(T) - 1 \).

Proof Let \( T_1, T_2, \ldots, T_k \) be the components of \( T \setminus N[u] \), where \( u \) is a pendant vertex of \( T \). By the Remark 5.1.6, \( \beta_0(T \setminus N[u]) = \beta_0(T_1) + \cdots + \beta_0(T_k) \). Let \( S_i \) be a \( \beta_0 \)-set of \( T_i \) for \( i = 1, 2, \ldots, k \). Then \( S_1 \cup S_2 \cup \cdots \cup S_k \cup \{u\} \) is a \( \beta_0 \)-set of \( T \). Therefore \( \beta_0(T) = \beta_0(T_1) + \cdots + \beta_0(T_k) + 1 \). Hence \( \beta_0(T \setminus N[u]) = \beta_0(T) - 1 \).

Using the above results we give an algorithm to find the independence number and an independent set of a tree having \( n \) vertices.

Algorithm 5.1.8

Given a tree \( T \) with self adjacent matrix \( S_A = (s_{aij}) \) for \( i, j = 1, 2, \ldots, n \) and \( m \) denotes the cardinality of a \( \beta_0 \)-set.

Input: The self adjacent matrix \( s_{aij} \) for \( i = 1, 2, \ldots, n \) and \( j = 1, 2, \ldots, n \).

Let \( m = 0 \).

Step 1

For \( k = 1, 2, \ldots, n \), calculate the row sum \( r_{si} = \sum_{j=1}^{n} s_{aij} \).
Step 2

2.1 Select i such that \( r_s[i] = 2 \).

2.2 Make \( m = m + 1 \).

2.3 Assign \( ind[m] = i \).

Step 3

3.1 Find \( j \) such that \( j \neq i \) and \( sa[i][j] = 1 \).

3.2 Make \( sa[i][k] = 0 \)

\[
\begin{align*}
sa[j][k] &= 0 \\
sa[k][i] &= 0 \\
sa[k][j] &= 0 \quad \text{for } k = 1, 2, \ldots, n.
\end{align*}
\]

Step 4

4.1 Perform Step 1

4.2 For \( u = 1 \) to \( n \) do

4.2.1 If \( r_s[u] = 1 \), then \( m = m + 1 \)

4.2.2 \( sa[u][l] = 0 \); for \( l = 1, 2, \ldots, n \)

4.3 Assign \( ind[m] = u \).

Step 5

5.1 Repeat Step 1

5.2 If \( r_s[k] \neq 0 \) (for \( 1 \leq k \leq n \)), then go to Step 2.

Step 6

6.1 Print \( m \)

6.2 Print \( ind[1], ind[2], \ldots, ind[m] \).
**Theorem 5.1.9** Algorithm 5.1.8 gives the independent number and an independent set of a given tree \( T \).

**Proof** Let \( u \) be a pendant vertex of \( T \). By the Remark 5.1.5, there exists a \( \beta_0 \)-set containing the pendant vertex \( u \). One of the pendant vertices \( u \) is identified by the Step 2. Step 3 removes the pendant vertex \( u \) and its adjacent vertex from the tree. After the implementation of the Step 3, the resulting matrix is the self adjacency matrix of \( T_1 = T \setminus N[u] \). Hence by the Theorem 5.1.7, \( \beta_0(T_1) = \beta_0(T) - 1 \). Since all the isolated vertices have to be included in an independent set, all the isolated vertices have been identified and included in Step 4. It can be seen that the Step 5 checks that whether there is any vertex remaining in the tree and it continues recursively by passing to Step 2.

**Example 5.1.10** Consider the following graph.

![Figure 5.2](image-url)
The self adjacent matrix corresponding to the above tree is given below.

\[
\begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
5 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
7 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
8 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
9 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
\end{pmatrix}
\]

The row sum is calculated for all values 1 to 9 from the Step 1. A pendant vertex 1 is selected using Step 2. The value of m becomes 1 and \( \text{ind}[1] = 1 \). The vertices 1 and 5 are deleted from the graph. i.e., the entries in the rows and columns corresponding to the vertices 1 and 5 are made zero. The resultant matrix is the self adjacency matrix corresponding to the graph given below.

![Graph](image)

**Figure 5.3**
Step 5: \( m = 2 \) and \( \text{ind}[2] = 2 \). The isolated vertex 2 is deleted from the above graph.

The process is repeated from the Step 2.

The pendant vertex 3 is identified and it is removed from the graph with its adjacent vertex 9. Here \( m = 3 \) and \( \text{ind}[3] = 3 \). The isolated vertex in the resultant graph is deleted with \( m = 4 \) and \( \text{ind}[4] = 4 \). The repetition from Step 2 gives \( m = 5, \text{ind}[5] = 6 \) and \( m = 6, \text{ind}[6] = 8 \).

Finally, we find that the independence number is 6 and a \( \beta_o \)-set is \{1, 2, 3, 4, 6, 8\}. These are printed using the Step 6.

**Working rule 5.11**

(i) Identify a pendant vertex \( u \) and put it in the \( \beta_o \)-set.

(ii) Remove \( N(u) \) from the tree.

(iii) If there exists isolated vertex, then add it in the \( \beta_o \)-set.

(iv) Repeat the process until possible.

**§ 5.2. ALGORITHM FOR PACKING NUMBER OF A TREE.**

First let us prove some theorems which will be useful for the algorithm.
Theorem 5.2.1 Let $T$ be the tree and $u$ be any pendant vertex of $T$. Then there exists a $\rho$-set containing the vertex $u$.

**Proof** Let $S$ be any $\rho$-set of the given tree $T$. Assume that $u \notin S$. Let $v$ be the support of $u$ in $T$. We claim that $N[v] \cap S \neq \emptyset$. If not, for a pendant vertex $x \in N(v)$, $S \cup \{x\}$ is a packing set. It is a contradiction to the fact that $S$ is a $\rho$-set. Further it is noticed that $|N(v) \cap S| = 1$.

**Case: 1** If $v \in S$, then it is easy to see that the set $S_1 = S \setminus \{v\} \cup \{u\}$ is a $\rho$-set containing $u$.

**Case: 2** Suppose $v \notin S$. Then, by the fact proved above, there exists a vertex $w \in N(v) \cap S$. Note that, by the definition of a $\rho$-set, $w$ is the only vertex in $N(v) \cap S$. Now $S_1 = S \setminus \{w\} \cup \{u\}$ is a $\rho$-set containing $u$.

**Remark 5.2.2** In view of the above theorem, we can start with any pendant vertex for finding a $\rho$-set of a tree.

**Theorem 5.2.3** Let $T$ be a tree and $u$ be a pendant vertex. Let $v$ be the support of $u$. Suppose $T \setminus N[v]$ has isolated vertices, then $N(v)$ contains some supports of $T$. Also, for each support $x \in N(v)$, $N(x) \cap S \neq \emptyset$ for any $\rho$-set $S$ containing $u$. 
Proof Let \( v \) be the support of the pendant vertex \( u \). Suppose \( T \setminus N[v] \) has isolated vertices, it is easy to see that \( N(v) \) has some supports of \( T \). By the Theorem 5.2.1, let \( S \) be a \( \rho \)-set containing \( u \). Assume that \( N(x) \cap S = \emptyset \) for a support \( x \in N(v) \). This implies that \( S_1 = S \cup \{ y \} \), for a pendant vertex \( y \in N(x) \) is a packing set of \( T \), contradicts the maximality of \( S \).

Corollary 5.2.4 Let \( v \) be any support of the tree \( T \) and \( S \) be any \( \rho \)-set. Then \( |N[v] \cap S| = 1 \).

Theorem 5.2.5 Let \( T \) be a tree and \( v \) be the support adjacent to the pendant vertex \( u \) in \( T \). Let \( S \) be a \( \rho \)-set containing \( u \). Let \( T_1 = T \setminus N[v] \). Suppose each non pendant vertex in \( N(v) \) is adjacent to exactly one vertex in \( T_1 \), then \( \rho(T_1) = \rho(T) - 1 \).

Proof It is given that no vertex in \( N(v) \) is adjacent to more than one vertex of \( T_1 \). Therefore, a component in \( T_1 \) is adjacent to only one vertex in \( N(v) \). Let \( S_1 \) be a \( \rho \)-set of \( T_1 = T \setminus N[v] \). Suppose \( \rho(T_1) > \rho(T) - 1 \), then \( S_2 = S_1 \cup \{ u \} \) is a \( \rho \)-set of \( T \) and so \( \rho(T) = |S_2| = |S_1| + 1 = \rho(T_1) + 1 > \rho(T) \). This is a contradiction. Similarly one can raise a contradiction when \( \rho(T_1) < \rho(T) - 1 \). Hence \( \rho(T_1) = \rho(T) - 1 \).
**Remark 5.2.6** By the above theorem, \( S = S_1 \cup \{u\} \) where \( S \) and \( S_1 \) are \( \rho \)-sets of \( T \) and \( T_1 \) respectively. For example consider the following tree.

![Figure 5.4](image)

In the figure 5.4, the vertex 1 is pendant and 2 is the support of 1. Let \( T_1 = T \setminus \{1, 2, 3, 7, 10\} \). Clearly \( T_1 \) consists of three components \( T \setminus \{4, 5, 6\} \), \( T \setminus \{8, 9\} \) and \( T \setminus \{11\} \). The vertices 3, 7 and 10 are the non-pendant vertices in \( N(2) \) and each of them is adjacent to exactly one vertex in \( T_1 \). It can be seen that \( \rho(T) = \rho(T_1) + 1 = 4 \) and here \( S_1 = \{5, 8, 11\} \), \( S = \{1, 5, 8, 11\} \).

**Remark 5.2.7** (i) Let \( T \) be a tree and \( v \) be any support adjacent to the pendant vertex \( u \). Let \( S \) be the \( \rho \)-set containing \( u \) and \( T_1 = T \setminus N(v) \). If \( \rho(T_1) > \rho(T) - 1 \), then there exists at least one vertex \( v_1 \in N(v) \) in \( T \) such that \( v_1 \) is adjacent to at least two vertices in \( T_1 \) and \( v_1 \) satisfies the condition that \( |N[N(v_1)] \cap S| > 1 \) (with reference to \( T \)).
(ii) Let \( N(v) = \{v_1, v_2, \ldots, v_r, v_{r+1}, \ldots, v_k\} \) where \( v_1, v_2, \ldots, v_r \) (\( r < k \)) are the pendant vertices adjacent to \( v \) in the tree \( T \) and \( T^1 = T \setminus \{v, v_1, v_2, \ldots, v_r\} \). Let \( T_{r+1}, T_{r+2}, \ldots, T_k \) be the components of \( T^1 \) such that \( v_j \in T_j \) for \( j = r + 1, r + 2, \ldots, k \) and \( S_j \) be the \( \rho \)–set of \( T_j \) for \( j = r + 1, r + 2, \ldots, k \). Let \( S = S_{r+1} \cup S_{r+2} \cup \ldots \cup S_k \). Suppose \( S \) does not contain any of the vertices \( v_{r+1}, v_{r+2}, \ldots, v_k \) (more specifically, \( v_j \notin S_j \) for all \( j = r + 1, r + 2, \ldots, k \)), then \( \rho(T) = \rho(T \setminus N[v]) + 1 \).

**Algorithm 5.2.8**

Given: A tree \( T \) with \( n \) vertices and \( V(T) = \{1, 2, \ldots, n\} \). Here \( m \) denotes the number of vertices in a \( \rho \)–set, where as the elements of a \( \rho \)–set are stored in the array \( pa[i] \).

**Input:** Self adjacent matrix \( sa[i][j] \), for \( i, j = 1, 2, \ldots, n \) of the tree \( T \).

Initialize \( m = 0 \).

**Step 1**

Calculate the row sum \( rs[i] = \sum_{j=1}^{n} sa[i][j] \) and

the column sum \( cs[i] = \sum_{i=1}^{n} sa[i][j] \) for \( i = 1, 2, \ldots, n \).
Step 2
2.1 Select $i$ such that $rs[i] = 2$ and $cs[i] = 2$.
2.2 Let $m = m + 1$.
2.3 Assign $pa[m] = i$.

Step 3
3.1 Find $j$ such that $j \neq i$ and $sa[i][j] = 1$.
3.2 Find all $k$ in $N[j]$ such that $sa[j][k] = 1$.
3.3 Make $sa[l][k] = 0$, for $l = 1, 2, \ldots, n$.

Step 4
4.1 Repeat Step 1.
4.2 For $w = 1, 2, \ldots, n$ do
   If $rs[w] = 1$ and $sa[w][w] \neq 1$, then make $sa[w][u] = 0$ for $u = 1, 2, \ldots, n$.
4.3 Repeat Step 1.

Step 5
5.1 For $s = 1, 2, \ldots, n$ do
   If $rs[s] = 1$ and $cs[s] = 1$, then $sa[s][t] = 0$ for $t = 1, 2, \ldots, n$.
5.2 Let $m = m + l$ and $pa[m] = s$.

Step 6
6.1 For $s = 1, 2, \ldots, n$ do
   If $rs[s] = 1$ and $cs[s] = 2$, then find $j$ such that $sa[j][s] = 1$ and $s \neq j$. 
6.2 Make \( sa[i][v] = 0 \) for \( v \) such that \( sa[j][v] = 1 \) for \( i = 1, 2, \ldots, n \).

6.3 Let \( m = m + 1 \).

6.4 Assign \( pa[m] = s \).

**Step 7**

7.1 Repeat Step 1.

7.2 If \( rs[u] > 0 \) for some \( u \), then go to Step 2.

**Step 8**

8.1 Print \( m \)

8.2 Print \( pa[1], pa[2], \ldots, pa[m] \).

**Remark 5.2.9** Let \( v \) be any support of the tree \( T \). Let \( M \) be the self adjacency matrix corresponding to the tree \( T \). After making all the entries of the columns of the matrix \( M \) corresponding to the vertices in \( N[v] \) as zeros, we observe the following with regard to the resultant matrix \( M_1 \).

(i) All the entries of the rows corresponding to the pendant (with respect to \( T \)) vertices and \( v \) in \( N[v] \) are zero in \( M_1 \).

(ii) For all the non-pendant vertices in \( N[v] \), the rows corresponding to the vertices will contain some non-zero entries.

(iii) The row sum corresponding to the non-pendant vertex \( x \in N[v] \) in \( M_1 \) is equal to the number of vertices in \( T_1 \) adjacent to the vertex \( x \) in \( T \).
**Theorem 5.2.10** Algorithm 5.2.8 gives the packing number of a tree.

**Proof** By the Remark 5.2.2, one can start with a pendant vertex u to find a \( \rho \)-set \( S \). Such a \( u \) is identified in the Step 2.1 using the self adjoint matrix \( M \). The support \( v \) of \( u \) is identified and all the pendant vertices and \( v \) in \( N[v] \) are deleted from \( T \) in Step 3. Further due to Step 3, we get the components \( T_{r+1}, T_{r+2}, \ldots, T_k \) of \( T^1 = T \setminus \{v_1, v_2, \ldots, v_r, v\} \) for our further consideration where \( N(v) = \{v_1, v_2, \ldots, v_{r+1}, v_{r+2}, \ldots, v_k\} \) (Remark 5.2.7(ii)).

This can be seen from the resultant matrix that the row entries corresponding to the vertices in \( N[v] \) are not all zero where as all the column entries are zero (Remark.5.2.9). Suppose one of the vertices \( u_i \), for \( i = r+1, r+2, \ldots, k \) is adjacent to only one vertex in \( T_1 = T \setminus N[v] \), then all such vertices are deleted from the tree by making the non-zero entries of rows as zeros in the matrix \( M_1 \) using Step 4. This can be done in view of Theorem 5.2.5. Let the resultant matrix be \( M_1 \). Suppose \( M_1 \neq 0 \). The rows of \( M_1 \) are the rows of the self adjacency matrix of \( T^1 \) and the columns of \( M_1 \) are the columns of the self adjacency matrix of \( T_1 \). This matrix \( M_1 \) satisfies at least one of the following properties:
(i) \( M_1 \) has a row \( i \) such that \( rs[i] = 2 = cs[i] \), i.e., there exists a pendant vertex \( x \) in \( T_1 \) which is not adjacent to any of the vertices in \( N[v] \).

(ii) \( M_1 \) has a row \( k \) such that \( rs[k] = 1 \) and \( cs[k] = 2 \), i.e., there exists an isolated vertex \( y \) in \( T_1 \) such that \( |N(N(y)) \cap V(T_1)| \geq \phi \).

(iii) \( M_1 \) has a row \( r \) such that \( rs[r] = 1, cs[r] = 0 \) and \( sa[r][r] \neq 1 \), i.e., there exists a vertex \( z \) in \( N(v) \) such that \( z \) is adjacent to exactly one vertex in \( T_1 \).

(iv) \( M_1 \) has a row \( j \) such that \( rs[j] = 1 = cs[j] \) i.e., \( T_1 \) has an isolated vertex \( x \) and no vertex in \( N(N(x)) \) with respect to \( T \) is in \( T_1 \).

Suppose (i) is true, then \( x \) is added in the \( \rho \)-set and the non zero entries in the columns corresponding to the vertices in \( N[N[v]] \) are made zero using Steps 2and 3 (Theorem 5.2.1). If (ii) is true, then the vertex \( y \) is included in the \( \rho \)-set and the non zero entries in the columns corresponding to the vertices in \( N[N[y]] \) are made zero using Step 6 (Theorem 5.2.3). If (iii) is true, then the vertex \( z \) in \( N[v] \) is removed using Step 4 (Theorem 5.2.5). i.e. the non zero entries in the row corresponding
z are made zero. Suppose (iv) is true, then the vertex x is included in the ρ-set i.e., by making the non zero entries in the row corresponding to x are zero using Step 5. Suppose (i), (ii), (iii) and (iv) are not satisfied in $M_1$. All the pendant vertices in $T_1$ are adjacent to at least one vertex $v_1$ in $N[v]$ and such $v_1$ has at least two adjacent vertices in $T_1$. Hence we see that $T$ contains a cycle.

Let the resultant matrix be $M_2$. If $M_2 \neq 0$, then the algorithm is continued by verifying any of the conditions stated above. The process is repeated until we get the zero matrix (Step 7).

**Example 5.2.11**

The corresponding self adjacency matrix is given below.

![Figure 5.5](image.png)

The corresponding self adjacency matrix is given below.

Step 2: $i = 1$; $m = 1$; $pa[1] = 1$.

Step 3: $j = 2$ and all the entries in the columns 1, 2 and 3 are made zero.

The matrix and the corresponding graph are as shown below.

\[
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{bmatrix}
\]

Figure 5.6

Step 6: By Step 6.1, $s = 4$ and $j = 3$.

According to the condition in Step 6.2, the entries in the columns 4 and 6 are made zero and so the matrix becomes.

$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}$

As per Step 6.3 and 6.4, $m = 2; pa[2] = 4$.

Step 7: The Step 2.1 is repeated.


The control goes to Step 2, and so $i = 8; m = 3; pa[3] = 8$.

Step 3: $j = 7$. 

87
All the entries of the 7th and 8th columns are made zero by Step 3.

Now the matrix and the corresponding graph are as given follow.

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\( \begin{array}{c}
o 6 \\
o 5 \\
\end{array} \)

Figure 5.8

From Step 4, the row 6 is identified and the entries in the row are made zero using Step 4.2. Now Step 5 selects the 5th row and makes \( m = 4 \), \( pa[4] = 5 \). The entries in the row 5 are made zero.

Now the matrix is a zero matrix. Thus it prints \( m = 4 \) and \( pa[1] = 1 \), \( pa[2] = 4 \) \( pa[3] = 8 \) and \( pa[4] = 5 \).

**Working rule 5.2.12**

(i) Identify a pendant vertex \( u \) and put it in the \( \rho \)-set.

(ii) Remove pendant vertices and \( v \) in \( N[v] \) where \( v \) is adjacent to \( u \) and name the remaining vertices in \( N[v] \) as dummy.

(iii) If a dummy vertex is a pendant vertex, then remove it.

(iv) Include isolated vertices in the \( \rho \)-set if any.
§ 5.3 ALGORITHM TO FIND THE DOMINATION NUMBER OF A TREE.

**Theorem 5.3.1** If $T$ is a tree, then there is a $\gamma$-set containing all the supports of $T$.

**Proof** Let $S$ be a $\gamma$-set and $u$ be any pendant vertex and $v$ be the support of $u$. Since $S$ is a dominating set, if $v \notin S$, then $u$ must be in $S$. Clearly $S \setminus \{v\} \cup \{u\}$ is a dominating set.

**Remark 5.3.2** Suppose $v_1, v_2, \ldots, v_r$ be the pendant vertices adjacent to the vertex $v$ in $T$. Let $v_{r+1}, v_{r+2}, \ldots, v_k$ be the non-pendant vertices in $N(v)$ and $T_1 = T \setminus N[v]$. If all the non-pendant vertices in $N(v)$ are adjacent to exactly one vertex in $T_1$, then $\gamma(T) = 1 + \gamma(T_1)$.

**Theorem 5.3.3** Let $v_1, v_2, \ldots, v_k$ be all the distinct supports of $T$. Consider the graph $T_1 = T \setminus N[v_1, v_2, \ldots, v_k]$. If $V(T_1) \neq \emptyset$ and $\gamma(T_1) > \gamma(T) - k$, then there exists a non-pendant vertex $v \in N(v_1, v_2, \ldots, v_k)$ such that there are at least two vertices in $T_1$ adjacent to $v$.

**Proof** By the assumption, $T_1$ exists and from the Theorem 5.3.1, there exists a $\gamma$-set $S$ containing all the supports $v_1, v_2, \ldots, v_k$. By the Remark...
5.3.2. \( |S| > k \). Assume that each non-pendant vertex \( v \in N(v_1, v_2, \ldots, v_k) \) is adjacent to exactly one vertex in \( T_1 \). Let \( G_1, G_2, \ldots, G_s \) be the components of \( T_1 \). Clearly \( \gamma(T_1) = \gamma(G_1) + \gamma(G_2) + \cdots + \gamma(G_s) \). It can be seen that the number of non-pendant vertices in \( N(v_1, v_2, \ldots, v_k) \) is \( s \). Thus \( \gamma(T) = k + \gamma(G_1) + \gamma(G_2) + \cdots + \gamma(G_s) = k + \gamma(T_1) \). It is contradiction to \( \gamma(T_1) > \gamma(T) - k \), (Hence there exists a non-pendant vertex in \( N(v_1, v_2, \ldots, v_k) \) which is adjacent to at least two vertices in \( T_1 \).

**Remark 5.3.4** Let \( v \) be a support of \( T \). Assume that \( N(v) = \{v_1, v_2, \ldots, v_r, v_{r+1}, \ldots, v_k\} \) where \( v_1, v_2, \ldots, v_r \) (\( r < k \)) are the pendant vertices and the remaining are non-pendant vertices. Let \( T_1 = T \setminus N[v] \).

(i) Suppose \( S_1 \) is a \( \gamma \)-set of \( T_1 \) and \( v_i \notin S \), for \( i = r + 1, \ldots, k \), then \( S = S_1 \cup \{v\} \) is a \( \gamma \)-set of \( T \).

(ii) On the other hand if \( v_i \in S \) for some \( i = r + 1, \ldots, k \), then \( \gamma(T_1) \geq \rho(T) + 1 \).

**5.3.5 Algorithm**

Given: A Tree \( T \) with \( n \) vertices and \( V(T) = \{1, 2, \ldots, n\} \). Here \( m \) denotes the number of vertices in the \( \gamma \)-set.
**Input:** Self adjacent matrix $sa[i][j]$, $i, j = 1, 2, \ldots, n$ of the tree $T$.

Initialize $m = 0$.

**Step 1**

1. Calculate the row sum $rs[i] = \sum_{j=1}^{n} sa[i][j]$ and
2. the column sum $cs[i] = \sum_{i=1}^{n} sa[i][j]$ for $i = 1, 2, \ldots, n$.

**Step 2**

1. Select $i$ such that $rs[i] = 2$ and $cs[i] = 2$.
2. Let $m = m + 1$.

**Step 3**

1. Find $j$ such that $j \neq i$ and $sa[i][j] = 1$.
2. Find all $k$ in $N[j]$ such that $sa[j][k] = 1$.
3. Make $sa[l][k] = 0$, for $1 = 1, 2, \ldots, n$.
4. Assign $dom[m] = j$.

**Step 4**

1. Repeat Step 1.
2. For $w = 1, 2, \ldots, n$ do
   
   If $rs[w] = 1$ and $sa[w][w] \neq 1$, then make $sa[w][u] = 0$, for $u = 1, 2, \ldots, p$.
3. Repeat Step 1.
Step 5

5.1 For \( s = 1, 2, \ldots, n \) do

   If \( rs[s] = 1 \) and \( cs[s] = 1 \), then \( sa[s][t] = 0 \) for \( t = 1, 2, \ldots, n \).

5.2 Let \( m = m + 1 \), \( \text{dom}[m] = s \).

Step 6

6.1 For \( s = 1, 2, \ldots, n \) do

   If \( rs[s] = 1 \) and \( cs[s] = 2 \), then find \( j \) such that \( sa[j][s] = 1 \) and \( s \neq j \).

6.2 Make \( sa[i][v] = 0 \), for \( v \) such that \( sa[j][v] = 1 \) for \( i = 1, 2, \ldots, n \);

6.3 Let \( m = m + 1 \);

6.4 Assign \( \text{dom}[m] = j \).

Step 7

7.1 Repeat Step 1.

7.2 If \( rs[u] > 0 \) for some \( u \), then go to Step 2.

Step 8

8.1 Print \( m \)

8.2 Print \( \text{dom}[1], \text{dom}[2], \ldots, \text{dom}[m] \).

Theorem 5.3.6 The above algorithm gives the domination number and a dominating set of a given tree.
Proof By the Theorem 5.3.1, one can start with a pendant vertex $u$ to find a $\gamma$–set $S$. Such a vertex is identified in the Step 2.1 using the self adjoint matrix $M$. The support $v$ of $u$ is identified and all the pendant vertices $v_1(=u), v_2, \ldots, v_r$ adjacent to $v$ and the vertex $v$ in $N[v]$ are deleted from $T$ in Step 3. Further due to Step 3, we get the components $T_{r+1}, T_{r+2}, \ldots, T_k$ of $T' = T \setminus \{v_1, v_2, \ldots, v_r, v\}$ for our further consideration where $N(v) = \{v_1, v_2, \ldots, v_{r+1}, v_{r+2}, \ldots, v_k\}$. This can be seen from the resultant matrix that the row entries corresponding to the vertices in $N[v]$ are not all zero where as all the column entries are zero (Remark 5.2.9). Suppose one of the vertices $v_i$, for $i = r+1, r+2, \ldots, k$ is adjacent to only one vertex in $T_1 = T \setminus N[v]$, then all such vertices are deleted from the tree, by making the non-zero entries of rows as zeros in the matrix $M_1$ using Step 4. This can be done in view of Remark 5.3.4. Let the resultant matrix be $M_1$. Suppose $M_1 \neq 0$. The rows of $M_1$ are the rows of the self adjacency matrix of $T_1$ and the columns of $M_1$ are the columns of the self adjacency matrix of $T_1$. This matrix $M_1$ satisfies at least one of the following properties:

(i) $M_1$ has a row $i$ such that $rs[i] = 2 = cs[i]$, i.e., there exists a pendant vertex $x$ in $T_1$ which is not adjacent to any of the vertices in $N[v]$. 

93
(ii) \( M_1 \) has a row \( k \) such that \( rs[k] = 1 \) and \( cs[k] = 2 \), i.e., there exists an isolated vertex \( y \) in \( T_1 \) such that \( |N(N(y)) \cap V(T_1)| \geq \phi \).

(iii) \( M_1 \) has a row \( r \) such that \( rs[r] = 1, cs[r] = 0 \) and \( sa[r][r] \neq 1 \), i.e., there exists a vertex \( z \) in \( N(v) \) such that \( z \) is adjacent to exactly one vertex in \( T_1 \).

(iv) \( M_1 \) has a row \( j \) such that \( rs[j] = 1 = cs[j] \) i.e., \( T_1 \) has an isolated vertex \( x \) and no vertex in \( N(N(x)) \) with respect to \( T \) is in \( T_1 \).

Suppose (i) is true, then the vertex adjacent to \( x \) is added in the \( \gamma \)-set and the non zero entries in the columns corresponding to the vertices in \( N[N(v)] \) are made zero using Steps 2 and 3 (Theorem 5.3.1). If (ii) is true, then the vertex adjacent to \( y \) is included in the \( \gamma \)-set and the non zero entries in the columns corresponding to the vertices in \( N[N(y)] \) are made zero using Step 6 (Theorem 5.2.3). If (iii) is true, then the vertex \( z \) in \( N[v] \) is removed using Step 4 (Theorem 5.3.2). i.e. the non zero entries in the row corresponding to \( z \) are made zero. Suppose (iv) is true, then the vertex \( x \) is included in the \( \gamma \)-set i.e., by making the non zero entries in the row corresponding to \( x \) are zero using Step 5. Suppose (i), (ii), (iii) and (iv) are not satisfied in \( M_1 \). All the pendant vertices in \( T_1 \)
are adjacent to at least one vertex \( v_1 \) in \( N[v] \) and such \( v_1 \) has at least two adjacent vertices in \( T_1 \). Hence we see that \( T \) contains a cycle.

Let the resultant matrix be \( M_2 \). If \( M_2 \neq 0 \), then the algorithm is continued by verifying any of the conditions stated above. The process is repeated until we get the zero matrix (Step 7).

**Example 5.3.7**

![Diagram of a graph](image)

**Figure 5.9**

The corresponding self adjacent matrix of the Fig. 5.9 is given below.

Step 2: $i = 1$, $m = 1$.

Step 3: The non-zero entries in the columns 1, 2 and 3 are made zero and assigns $dom[1] = 2$. The matrix and the corresponding graph are as shown below.

\[
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

---

97
Step 6: \( s = 11, m = 2, j = 3 \) and \( \text{dom}[2] = 3 \), correspondingly the column entries of the vertices 3, 11 and 4 are made zero.

The process is repeated. Step 2 gives that \( m = 3 \) and \( i = 10 \), Step 3 selects \( j \) as 9 and the column entries corresponding to the vertices 8, 9 and 10 are made zero. The assigned value of \( \text{dom}[3] = 9 \).

The resultant matrix and its graph are as follows.

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
\]
Step: 4.1 calculates the row sums and the corresponding column sums.

Step 4.2 deletes the vertex 4 and 8, i.e. the entries in the 4th and 8th rows are made zero. Again the process is repeated from Step 2 after the calculation of the row sums and the column sums. Step 2: \( m = 4 \). Step 3: \( j = 5 \) and the column corresponding to the vertices 5, 6, and 7 are made zero. At the same time, it assigns the value \( \text{dom}[4] = 6 \). Now the graph becomes

\[
\begin{array}{ccccccc}
7 & 0 & - & - & - & - & 0 & 12 \\
\end{array}
\]

Figure 5.12

Step: 4 deletes the vertex 7 and so the resultant matrix contains only one non-zero entry in the 12th row, 12th column. Step 5: \( m = 5 \) and \( \text{dom}[5] = 12 \). Finally, the values \( m = 5 \) and \( \text{dom}[1] = 2 \), \( \text{dom}[2] = 3 \), \( \text{dom}[3] = 9 \), \( \text{dom}[4] = 6 \), and \( \text{dom}[5] = 12 \) are printed using the Step 7.

**Working rule 5.3.8**

(i) Identify a pendant vertex \( u \) and put the vertex adjacent to \( u \) in the \( \gamma \)-set.

(ii) Remove the pendant vertices and \( v \) in \( N[v] \) where \( v \) is adjacent to \( u \) and name the remaining vertices in \( N[v] \) as dummy.

(iii) If a dummy vertex is a pendant vertex, then remove it.

(iv) Repeat the process until possible.
Remark 5.3.9 All the algorithms presented in this chapter are of time complexity $O(n^3)$ where $n$ is the number of vertices.

A computer program in C-language to each of the algorithms 5.1.8, 5.2.8 and 5.3.5 is given in the appendix.