***Problem Definition:-***

A tree is given, which can be divided into sub-trees. The basic condition is every subtree must have minimum 4 nodes. Count the number of subtrees, in a tree satisfying the criteria. Design efficient method to divide the tree into sub-trees.

***Detailed Description:-***

A valid tree is taken in the form of adjacency matrix and the edges is determined with the help of matrix. The Problem is implemented by the concept of breadth first search with the help of Arraylist.

The Logic used here is that count of every descendant nodes is found for the node which is going to be the root of subtree.

For every node the corresponding count of descendants nodes is calculated and if the count is greater than four than that nodes is considered as Root node of subtree and subtree is taken into consideration.

***Technology Stack: -***

The entire coding is done in Java with NetBeans IDE 8.0.1.

***Classes : -***

The program makes the use of 4 classes: -

1. Ads\_assign (Main Class)

This Main class drive the whole program where determination of tree, formation of ordered list, Node creation from class tree, count of descendant nodes and count of subtree is accomplished.

2. Tree

This Class look after the building of node at run time and returning the tree object

***Fundamental Logic:-***

Starting from the root node of the given tree, we try to do breadth search and find the nodes in the ordered list .

For every node in the ordered list we find the count of descendant nodes .

The count is searched by implementing the queue data structure with Collections class interface i.e. ArrayList.

The count of each node for the subtree is determined by removing the node, Exploring the branching node, adding up the count and again exploring the other node and continuing the process.

Finally maintaining the count in array and determining which node has count greater than or equal to four.

The node satisfying the condition becomes root node for that subtree and thus displaying the no. of subtree.