7)

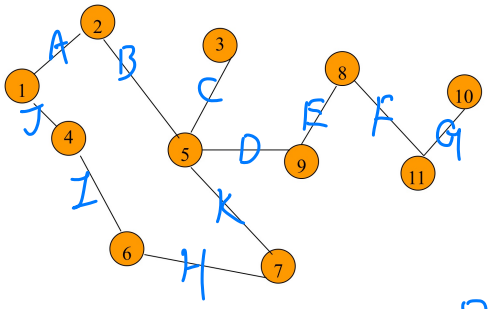
a) Prove that when function *DFS* is applied to a connected graph, the edges of *T* form a tree.

b) Prove that when function *BFS* is applied to a connected graph, the edges of *T* form a tree.

```
virtual void Graph::BFS(int v)
{// A breadth first search of the graph is carried out beginning at vertex v.
 // visited[i] is set to true when v is visited. The function uses a queue.
    visited = new bool [n];
    fill(visited, visited + n, false);
    visited[v] = true;
    Queue<int> q;
    q.Push(v);
    while (!q.IsEmpty()) {
        v = q.Front();
        q.Pop();
        for (all vertices w adjacent to v) // actual code uses an iterator
            if (!visited[w]) {
                q.Push(w);
                visited[w] = true;
            }
    } // end of while loop
    delete [] visited;
}
```

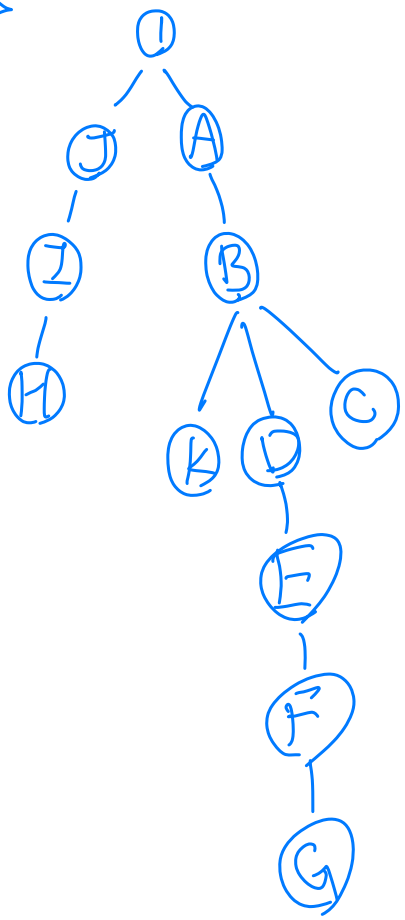
```
virtual void Graph::DFS() // Driver
{
    visited = new bool [n];
        // visited is declared as a bool* data member of Graph
    fill(visited, visited + n, false); // initialize visited to false
    DFS(0); // start search at vertex 0
    delete [] visited;
}

virtual void Graph::DFS(const int v) // Workhorse
{// Visit all previously unvisited vertices that are reachable from vertex v.
    visited[v] = true;
    for (each vertex w adjacent to v) // actual code uses an iterator
        if (!visited[w]) DFS(w);
}
```

Example:



BFS(1)

=>



DFS(1)