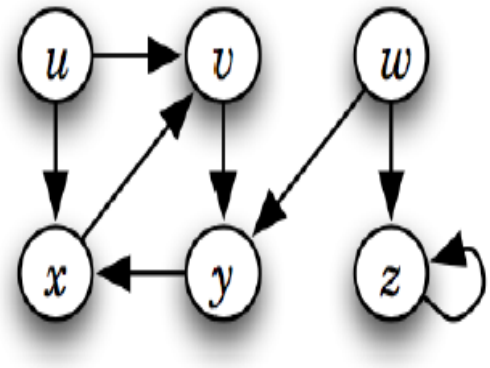
DFS Implementation

Pseudocode:

Application of DFS

```
DFS(G) ▷ G = (V, E)
1  for each vertex u ∈ V[G]
2        do color[u] ← WHITE
3           π[u] ← NIL
4  time ← 0
5  for each vertex u ∈ V[G]
6        do if color[u] = WHITE
7              then DFS-VISIT(G, u)
```

```
DFS-VISIT(G, u)
1  color[u] ← GRAY
2  time ← time + 1
3  d[u] ← time
4  for each vertex v ∈ Adj[u]
5        do if color[v] = WHITE
6              then π[v] ← u
7                   DFS-VISIT(G, v)
8  color[u] ← BLACK
9  f[u] ← time + 1
```



1, Apply DFS on a graph that you will take as a matrix.

(The template of the code is given below)

```
import java.util.Scanner;
public class DFS{

  public static int time;
  public static int[] color;
  public static int[] parent;
  public static int[] d; //starting time
  public static int[] f; //finishing time

  public static void main(String [] args){
    Scanner sc = new Scanner(System.in);
    int[][] graph = takeInputGraph(sc);

    color = new int[graph.length];
    parent = new int[graph.length];
    d = new int[graph.length];
    f = new int[graph.length];

    System.out.println("Give input of the source node");
    int s = sc.nextInt();
    dfs(graph,s);
  }
```

```
public static int[][] takeInputGraph(Scanner sc){
   System.out.println("Input the number of nodes in the graph");
   int node = sc.nextInt();
   System.out.println("Input the number of edges in the graph");
   int edge = sc.nextInt();
   int[][] mat = new int[node][node];
   for(int c=0; c<edge; c++){
      System.out.println("Enter the first node of the "+(c+1)+"th edge");
      int node1 = sc.nextInt();
      System.out.println("Enter the second node of the "+(c+1)+"th edge");
      int node2 = sc.nextInt();
      mat[node1][node2] = 1;
   }
   return mat;
}

public static void dfs(int[][] g, int s){
   time = 0;
   dfs_visit(g,s); //Running DFS from the source node

   //Checking if any nodes are still unvisited after running DFS once
   for(int c = 0; c<g.length; c++){
      if(color[c] = 0){
         dfs_visit(g,c);
```

```
        }
      }
    }

    public static void dfs_visit(int[][] g, int u){
      //DO IT YOURSELF
    }
}
```