```
#DFS (Depth-First Search) is a graph traversal algorithm that explores the vertices of a graph in a depthward motion, going as far as pos

class Graph:

  def __init__(self, edges, n):
    self.adjList = [[] for _ in range(n)]  # Initialize an empty adjacency list for each vertex

    # Add edges to the adjacency list
    for (src, dest) in edges:
      self.adjList[src].append(dest)
      self.adjList[dest].append(src)

def DFS(graph, v, discovered):
  discovered[v] = True  # Mark the current vertex as discovered
  print(v, end=' ')     # Print the current vertex

  for u in graph.adjList[v]:  # Iterate through the neighbors of the current vertex
    if not discovered[u]:   # If a neighbor is not discovered yet
      DFS(graph, u, discovered)  # Recursively call DFS on the neighbor

if __name__ == '__main__':
  edges = [
    (1, 2), (1, 7), (1, 8), (2, 3), (2, 6), (3, 4),
    (3, 5), (8, 9), (8, 12), (9, 10), (9, 11)
  ]

  n = 13

  graph = Graph(edges, n)  # Create a graph object

  discovered = [False] * n  # Initialize a list to track discovered vertices

  for i in range(n):
    if not discovered[i]:  # If a vertex is not discovered yet
      DFS(graph, i, discovered)  # Call DFS starting from that vertex
```

```
0 1 2 3 4 5 6 7 8 9 10 11 12
```