**DFS of Graph:**

Given a connected undirected graph. Perform a Depth First Traversal of the graph.  
**Note:** Use recursive approach.

**Input:**  
The first line of the input contains an integer **'T'** denoting the number of test cases. Then **'T'** test cases follow. Each test case consists of two lines. Description of testcases is as follows: The First line of each test case contains two integers **'N' and 'E'** which denotes the no of vertices and no of edges respectively. The Second line of each test case contains **'E'** space separated pairs **u** and **v** denoting that there is a edge from **u** to **v** .

**Output:**  
For each testcase, print the nodes while doing DFSstarting from node 0.

**Your task:**  
You don't need to read input or print anything. Your task is to complete the function **dfs**() which takes the Graph and the number of vertices as inputs and returns a list containing the DFS Traversal of the graph starting from the 0th node.

**Expected Time Complexity:**O(V + E).  
**Expected Auxiliary Space:**O(V).

**Constraints:**  
1 <= T <= 100  
2 <= N <= 104  
1 <= E <= (N\*(N-1))/2  
Graph doesn't contain multiple edges and self loops.

**Example:  
Input:**  
2  
5 4  
0 1 0 2 0 3 2 4  
4 3  
0 1 1 2 0 3

**Output:**  
0 1 2 4 3     
0 1 2 3

**Explanation:  
Testcase 1:**  
0 is connected to 1 , 2 , 3  
1 is connected to 0  
2 is connected to 0 and 4  
3 is connected to 0  
4 is connected to 2  
so starting from 0 , dfs will be 0 1 2 4 3.