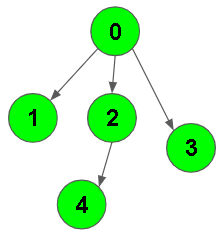
**BFS of graph**

**Easy**

Given a directed graph. The task is to do Breadth First Traversal of this graph starting from 0.  
**Note:**One can move from node u to node v only if there's an edge from u to v. Find the BFS traversal of the graph starting from the 0th vertex, from left to right according to the input graph. Also, you should only take nodes directly or indirectly connected from Node 0 in consideration.

**Example 1:**

**Input:**



**Output:** 0 1 2 3 4

**Explanation**:

0 is connected to 1 , 2 , 3.

2 is connected to 4.

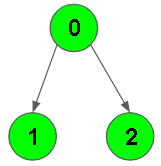
so starting from 0, it will go to 1 then 2

then 3. After this 2 to 4, thus bfs will be

0 1 2 3 4.

**Example 2:**

**Input:**



**Output:** 0 1 2

**Explanation**:

0 is connected to 1 , 2.

so starting from 0, it will go to 1 then 2,

thus bfs will be 0 1 2.

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

**Constraints:**  
1 ≤ V, E ≤ 104

**Company Tags**

[**Flipkart**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Flipkart) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Samsung**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Samsung) [**Ola Cabs**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Ola%20Cabs) [**Adobe**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Adobe) [**SAP Labs**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=SAP%20Labs)

//{ Driver Code Starts

// Initial Template for Java

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class CodingMaxima {

public static void main(String[] args) throws IOException {

BufferedReader br =

new BufferedReader(new InputStreamReader(System.in));

int T = Integer.parseInt(br.readLine().trim());

while (T-- > 0) {

String[] s = br.readLine().trim().split(" ");

int V = Integer.parseInt(s[0]);

int E = Integer.parseInt(s[1]);

ArrayList<ArrayList<Integer>> adj = new ArrayList<>();

for (int i = 0; i < V; i++) adj.add(i, new ArrayList<Integer>());

for (int i = 0; i < E; i++) {

String[] S = br.readLine().trim().split(" ");

int u = Integer.parseInt(S[0]);

int v = Integer.parseInt(S[1]);

adj.get(u).add(v);

// adj.get(v).add(u);

}

Solution obj = new Solution();

ArrayList<Integer> ans = obj.bfsOfGraph(V, adj);

for (int i = 0; i < ans.size(); i++)

System.out.print(ans.get(i) + " ");

System.out.println();

}

}

}

// } Driver Code Ends

class Solution {

// Function to return Breadth First Traversal of given graph.

public ArrayList<Integer> bfsOfGraph(int V, ArrayList<ArrayList<Integer>> adj) {

ArrayList<Integer> ar=new ArrayList<Integer>();

Queue<Integer> q=new LinkedList<>();

boolean[] visited=new boolean[V];

q.add(0);

visited[0]=true;

while(!q.isEmpty()){

int temp=q.poll();

ar.add(temp);

for(int nebour: adj.get(temp)){

if(!visited[nebour])

{

visited[nebour]=true;

q.add(nebour);

}

}

}

return ar;

}

}