# BFS of graph

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 and find the BFS traversal of the graph starting from the 0th vertex, from left to right according to the graph. Also, you should only take nodes directly or indirectly connected from Node 0 in consideration.
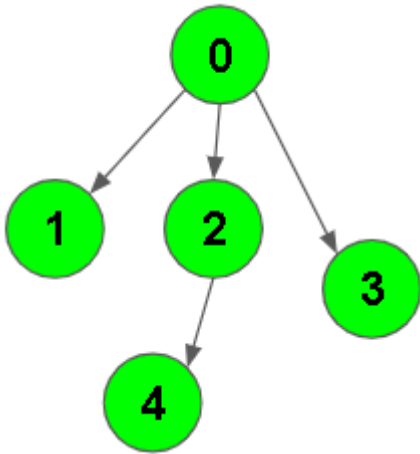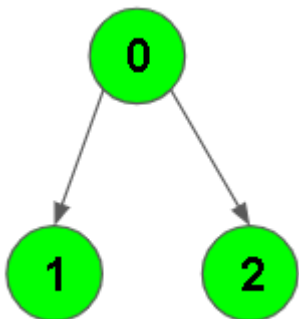


```
Output: 0 1 2 3 4
Explanation:
0 is connected to 1 , 2 , 3.
2 is connected to 4.
so starting from 0, it will go to 1 then 2
then 3.After this 2 to 4, thus bfs will be
0 1 2 3 4.
```

Example 2:


```
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 3 4.
```

```python
class Solution:

    #Function to return Breadth First Traversal of given graph.
    def bfsOfGraph(self, V, adj):
        # code here
        visited = [False]*V
        res = []
        self.helper(visited,adj,V,res,0)
        return res




    def helper(self,visited,adj,V,res,src):
        queue = []
        queue.append(src)
        while queue:
            temp = queue.pop(0)

            if visited[temp]==True:
                continue
            res.append(temp)
            visited[temp] = True
            for neigh in adj[temp]:
                if visited[neigh]==False:
                    queue.append(neigh)
        return res
```