

**“AZƏRBAYCAN HAVA YOLLARI” CJSC NATIONAL AVIATION ACADEMY”**

**Individual Work №: 6**

**Topic: Counts. BFS and DFS algorithms.**

**Subject: OOP**

**Teacher: Mamed Shahmaliyev**

**Group: 1459i Student:Ilaha Aliyeva**

**Baku 2022**

**Count the number of nodes at given level in a tree using BFS.**

* Difficulty Level : [Easy](https://www.geeksforgeeks.org/easy/)
* Last Updated : 27 Sep, 2021

Given a tree represented as an undirected graph. Count the number of nodes at a given level l. It may be assumed that vertex 0 is the root of the tree.

**Examples:**

Input : 7

0 1

0 2

1 3

1 4

1 5

2 6

2

Output : 4

Input : 6

0 1

0 2

1 3

2 4

2 5

2

Output : 3

[BFS](https://www.geeksforgeeks.org/breadth-first-traversal-for-a-graph/) is a traversing algorithm that starts traversing from a selected node (source or starting node) and traverses the graph layer-wise thus exploring the neighbour nodes (nodes that are directly connected to the source node). Then, move towards the next-level neighbor nodes.   
As the name BFS suggests, traverse the graph breadth wise as follows:  
**1.** First move horizontally and visit all the nodes of the current layer.   
**2.** Move to the next layer.

In this code, while visiting each node, the level of that node is set with an increment in the level of its parent node i.e., level[child] = level[parent] + 1. This is how the level of each node is determined. The root node lies at level zero in the tree.

**Explanation :**

0 Level 0

/ \

1 2 Level 1

/ |\ |

3 4 5 6 Level 2

Given a tree with 7 nodes and 6 edges in which node 0 lies at 0 level. Level of 1 can be updated as : level[1] = level[0] +1 as 0 is the parent node of 1. Similarly, the level of other nodes can be updated by adding 1 to the level of their parent.   
level[2] = level[0] + 1, i.e level[2] = 0 + 1 = 1.   
level[3] = level[1] + 1, i.e level[3] = 1 + 1 = 2.   
level[4] = level[1] + 1, i.e level[4] = 1 + 1 = 2.   
level[5] = level[1] + 1, i.e level[5] = 1 + 1 = 2.   
level[6] = level[2] + 1, i.e level[6] = 1 + 1 = 2.  
Then, count of number of nodes which are at level l(i.e, l=2) is 4 (node:- 3, 4, 5, 6)

# Python3 program to print

# count of nodes at given level.

from collections import deque

adj = [[] for i in range(1001)]

def addEdge(v, w):

    # Add w to v’s list.

    adj[v].append(w)

    # Add v to w's list.

    adj[w].append(v)

def BFS(s, l):

    V = 100

    # Mark all the vertices

    # as not visited

    visited = [False] \* V

    level = [0] \* V

    for i in range(V):

        visited[i] = False

        level[i] = 0

    # Create a queue for BFS

    queue = deque()

    # Mark the current node as

    # visited and enqueue it

    visited[s] = True

    queue.append(s)

    level[s] = 0

    while (len(queue) > 0):

        # Dequeue a vertex from

        # queue and print

        s = queue.popleft()

        #queue.pop\_front()

        # Get all adjacent vertices

        # of the dequeued vertex s.

        # If a adjacent has not been

        # visited, then mark it

        # visited and enqueue it

        for i in adj[s]:

            if (not visited[i]):

                # Setting the level

                # of each node with

                # an increment in the

                # level of parent node

                level[i] = level[s] + 1

                visited[i] = True

                queue.append(i)

    count = 0

    for i in range(V):

        if (level[i] == l):

            count += 1

    return count

# Driver code

if \_\_name\_\_ == '\_\_main\_\_':

    # Create a graph given

    # in the above diagram

    addEdge(0, 1)

    addEdge(0, 2)

    addEdge(1, 3)

    addEdge(2, 4)

    addEdge(2, 5)

    level = 2

    print(BFS(0, level))

# This code is contributed by mohit kumar 29

**Output:**

3