



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

← (<https://www.educba.com/circular-linked-lists-in-c/>)

→ (<https://www.educba.com/dfs-algorithm-in-c/>)



## Introduction to BFS algorithm in C

BFS is a traversal algorithm that is applied mostly on the graphs to travel from one node to another node for various manipulation and usage. The visit on each node using the BFS algorithm makes the entire algorithm treated as an efficient and useful algorithm. There are certain ways to traverse from one node to another node in a graph using the BFS algorithm in the sense that the selected node will be encountered first, followed by the next node to be reached either breadthwise or moving to the node directly. In this topic, we are going to learn





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

**Start Your Free Software Development Course**

Web development, programming languages, Software testing & others

There is no particular syntax for BFS, but some algorithmic steps are followed to traverse from root to the base. Thus the algorithm to be used is as follows:

Define the structure as represented below:

```
Struct node {  
    char lbl;  
    boolean vtrue;  
}
```

- Followed by the variables to be defined
- Then form the variables with the matrix defined
- Count each of the nodes.
- Display all the vertex
- Check for the adjacent vertexes
- Check for the final deployments and
- Construct the logic for BFS traversal.
- Run the main function
- Drive the entire program

More will clear in the Example section, where the implementation will make use of the algorithm with proper representation.



## How does BFS algorithm work in C?



[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

- Breadth-First Search (BFS) is an algorithm used for traversing and is also considered a searching tree with the data structure.
- A node which is the initial node or any arbitrary node from there the actual traversal will start then moving forward; it will be used for exploring the rest of the other nodes at neighbor which will be used for the current depth of the node before moving to the next node at the depth level.
- BFS works completely opposite to the DFS algorithm, which is also used for the same purpose of vertex of graph traversal using some rules and regulations which instead of nodes even adjoining nodes apply for backtracking and expansion of other nodes within it.
- BFS was already developed and implemented, but it got its final approval when used as an algorithm for traversal by C.Y Lee to find the shortest path by wire routing.
- The BFS is a non-recursive implementation that is similar to the non-recursive implementation of depth-first search but differs in ways like it makes use of queue, not stack, and also checks whether or not the vertex explored recently is the actual one to have a check on or already that vertex is explored. It has to be kept in mind that the same vertex should not repeatedly be traversed; then, only dequeuing from the queue is possible.
- I suppose for  $G_0$  is any tree, then in that tree replacement of queue from BFS algorithm is used with a stack for DFS yields.
- Queue comprises nodes and vertexes, which have labels stored for tracking and backtracking from the destination node.
- BFS search end result is nothing but a breadth-first tree that is optimized and will almost all the information for reference at any time.
- There is sometimes talk on the completeness of the graph when used BFS algorithm





(<https://www.educba.com/software-development/>)

release and output as predicted and traversed using DFS algo.

- BFS is the only algorithm that is considered as one of the finite and complete algorithms that makes the infinite graph represent implicitly which have some goals that are useful and relevant, not as DFS algorithm where the completeness and relevant information is not upto the mark.
- BFS algorithm has a pattern of a time complexity measure which, according to the implementation, comes out to be  $O(V+E)$  where  $V$  represents the number of vertexes present in the tree and  $E$  stands for the number of edges present in the tree. This complexity is considered finite only if all the nodes are covered at the time of traversal.
- BFS is used for solving many of the graph theory that exists like garbage collection copying, Testing of Bipartite graph, Serialization or deserialization of binary tree that allows a lot of reconstruction of tree in an efficient and effective manner.

## Example of BFS algorithm in C

This program demonstrates the implementation of the BFS algorithm in C language, which is used for various graph traversal ways by adjusting the neighbor nodes and adjacent nodes for manipulation, as shown below in the output.

```
#include<stdio.h>
#include<conio.h>
int a_0[30]
[20],q_1[30],visited_nodes[30],n_8,i_0,j_1,f_5=0,r_2=-1;
void bfs_logic(int v_8) {
for (i_0=1;i_0<=n_8;i_0++)
if(a_0[v_8][i_0] && !visited_nodes[i_0])
q_1[++r_2]=i_0;
```





(<https://www.educba.com/software-development/>)

```

,
void main() {
int v_8;
printf("\n Enter Vertices_to_represent:");
scanf("%d",&n_8);
for (i_0=1;i_0<=n_8;i_0++) {
q_1[i_0]=0;
visited_nodes[i_0]=0;
}
printf("\n Enter graph_data especially_in_matrix_format:\n");
for (i_0=1;i_0<=n_8;i_0++)
for (j_1=1;j_1<=n_8;j_1++)
scanf("%d",&a_0[i_0][j_1]);
printf("\n Enter Starting_vertex_for_traversal:");
scanf("%d",&v_8);
bfs_logic(v_8);
printf("\n Reachable_nodes_are:\n");
for (i_0=1;i_0<=n_8;i_0++)
if(visited_nodes[i_0])
printf("%d\t",i_0); else
printf("\n Bfs not_possible_if_not_proper.");
getch();
}

```



⚡ Popular Course in this category



(<https://www.educba.com/software-development/>)

Lifetime Access

★★★★★ 4.5 (8,635 ratings)

Course Price

**\$79** ~~\$399~~

[View Course](#)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

### Related Courses

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)

### Output:

**Explanation:** The program expects an input with the number of vertices that will be present in the tree. Then formulate the tree with the adjoining nodes represented as vertexes in the matrix format starting from root to base reachable nodes as shown.





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

implementation of trees with a lot of vertexes and nodes for traversal. BFS makes the implemented tree as finite, which also helps in providing relevant and required information.

## Recommended Articles

This is a guide to the BFS algorithm in C. Here we discuss How does BFS algorithm work in C along with the example and output. You may also have a look at the following articles to learn more –

1. [BFS Algorithm \(https://www.educba.com/bfs-algorithm/\)](https://www.educba.com/bfs-algorithm/)
2. [Patterns in C Programming \(https://www.educba.com/patterns-in-c-programming/\)](https://www.educba.com/patterns-in-c-programming/)
3. [Arrays in C Programming \(https://www.educba.com/arrays-in-c-programming/\)](https://www.educba.com/arrays-in-c-programming/)
4. [Socket Programming in C++ \(https://www.educba.com/socket-programming-in-c-plus-plus/\)](https://www.educba.com/socket-programming-in-c-plus-plus/)

## ALL IN ONE SOFTWARE DEVELOPMENT BUNDLE (600+ COURSES, 50+ PROJECTS)

- ☒ 600+ Online Courses
- ☒ 50+ projects
- ☒ 3000+ Hours
- ☒ Verifiable Certificates
- ☒ Lifetime Access

**Learn More**

<https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3>





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

---

## About Us

Blog (<https://www.educba.com/blog/?source=footer>)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)

Sign Up (<https://www.educba.com/software-development/signup/?source=footer>)

Corporate Training (<https://www.educba.com/corporate/?source=footer>)

Certificate from Top Institutions (<https://www.educba.com/educbalive/?source=footer>)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

## Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)







[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

## Certification Courses

All Courses (<https://www.educba.com/software-development/courses/?source=footer>)

Software Development Course - All in One Bundle (<https://www.educba.com/software-development/courses/software-development-course/?source=footer>)

Become a Python Developer (<https://www.educba.com/software-development/courses/python-certification-course/?source=footer>)

Java Course (<https://www.educba.com/software-development/courses/java-course/?source=footer>)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.



