

AI 1: Depth First Search

DFS algorithm in Python

DFS algorithm

```
def dfs(graph, start, visited=None):
```

```
    if visited is None:
```

```
        visited = set()
```

```
    visited.add(start)
```

```
    print(start)
```

```
    for next in graph[start] - visited:
```

```
        dfs(graph, next, visited)
```

```
    return visited
```

```
graph = {'0': set(['1', '2']),
```

```
        '1': set(['0', '3', '4']),
```

```
        '2': set(['0']),
```

```
        '3': set(['1']),
```

```
        '4': set(['2', '3'])}
```

```
dfs(graph, '0')
```

output:

0

2

1

3

4

What's NewNew TabChatGPTonline compiler - Yahoo IndiaOnline Python Compiler (Interp...

programiz.com/python-programming/online-compiler/

ProgramizPython Online Compiler

CLIENT

Quick edits. Seamless workflows.

Learn more

Adobe Acrobat

Programiz PRO

main.py

Run

Output

Clear

```
1 # Online Python compiler (interpreter) to run Python online.
2 # Write Python 3 code in this online editor and run it.
3
4 # DFS algorithm
5 def dfs(graph, start, visited=None):
6     if visited is None:
7         visited = set()
8     visited.add(start)
9
10    print(start)
11
12    for next in graph[start] - visited:
13        dfs(graph, next, visited)
14    return visited
15
16
17 graph = {'0': set(['1', '2']),
18         '1': set(['0', '3', '4']),
19         '2': set(['0']),
20         '3': set(['1']),
21         '4': set(['2', '3'])}
22
23 dfs(graph, '0')
24
```

```
0
1
4
3
2
3
2
=== Code Execution Successful ===
```

33°C

Search

ENG

20:55

```

# BFS algorithm in Python

import collections

# BFS algorithm
def bfs(graph, root):

    visited, queue = set(), collections.deque([root])
    visited.add(root)

    while queue:

        # Dequeue a vertex from queue
        vertex = queue.popleft()
        print(str(vertex) + " ", end="")

        # If not visited, mark it as visited, and
        # enqueue it
        for neighbour in graph[vertex]:
            if neighbour not in visited:
                visited.add(neighbour)
                queue.append(neighbour)

if __name__ == '__main__':
    graph = {0: [1, 2], 1: [2], 2: [3], 3: [1, 2]}
    print("Following is Breadth First Traversal: ")
    bfs(graph, 0)

```

OUTPUT

Following is Breadth First Traversal:

0 1 2 3



What's New x New Tab x ChatGPT x online compiler - Yahoo India x Online Python Compiler (Interp: x

programiz.com/python-programming/online-compiler/

Gmail YouTube Maps Gmail YouTube Maps Customize Links plot iris dataset Suggested Sites Imported From IE T132 T210 T186 Embedded All Bookmarks

Programiz
Python Online Compiler

Quick edits. Seamless workflows. [Learn more](#) **Adobe Acrobat** [Programiz PRO >](#)

main.py   [Share](#) [Run](#) **Output** [Clear](#)

```
1 /
2 # BFS algorithm
3 def bfs(graph, root):
4
5     visited, queue = set(), collections.deque([root])
6     visited.add(root)
7
8     while queue:
9
10         # Dequeue a vertex from queue
11         vertex = queue.popleft()
12         print(str(vertex) + " ", end="")
13
14         # If not visited, mark it as visited, and
15         # enqueue it
16         for neighbour in graph[vertex]:
17             if neighbour not in visited:
18                 visited.add(neighbour)
19                 queue.append(neighbour)
20
21
22 if __name__ == '__main__':
23     graph = {0: [1, 2], 1: [2], 2: [3], 3: [1, 2]}
24     print("Following is Breadth First Traversal: ")
25     bfs(graph, 0)
```

Following is Breadth First Traversal:
0 1 2 3
=== Code Execution Successful ===

