Exp No-2
Date: 14-9-24

Breadth First Search

AIM

To traverse or search through a graph in a level order manner

Algorithm

* Create a set visited to keep track
* Create queue
* Traverse - do
  when queue is not empty
  dequeue & print vertex
* for each neighbour if not visited add to visited set & enqueue to queue
* then algo stops when all node visited

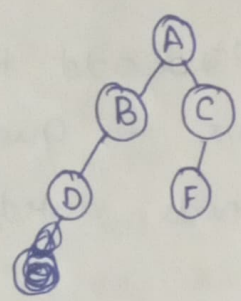
CODE

```
from collections import deque
def bfs(graph, start):
    visited = set()
    queue = deque([start])
    visited.add(start)
    while queue:
        vertex = queue.popleft()
        print(vertex, end = " ")
        for neighbour in graph[vertex]:
            if neighbour not in visited:
                visited.add(neighbour)
                queue.append(neighbour)
```

```
graph = {}
n = int(input("Enter the number of node"))
for i in range(n):
    node = input(f"Enter node {i+1}: ")
    neighbours = input(f"Enter the neighbour of
                {node} seperated by space ").split()
    graph[node] = neighbor.
start_node = input("enter the starting node :")
bfs(graph, start_node)
```

Output

→ for graph like



Enter number of nodes : 5
Enter Node 1 : a
Enter neighbor of a : b c
Enter Node 2 : b
Enter neighbor of b : d
Enter node 3 : c
Enter neighbor of c : f
Enter node 4 : d
Enter neighbor of d :
Enter nod 5 : f
Enter neighbor of f :
Enter starting node : a

a b c d f

RESULT

→ Thus BFS is successfully executed & O/p is verified