



Experiment-1.3

Student Name: Utkarsh Joshi

Branch: BE-CSE

Semester: 5th

Subject Name: AI & ML Lab

UID: 21BCS9158

Section/Group: ST_802_A

Date of Performance: 23-08-2023

Subject Code: 21CSH-316

1. **Aim:** Implement the BFS algorithm and analyze its performance and characteristics
2. **Objective:** The objective of this experiment is to implement the BreathFirst Search (BFS) algorithm and analyze its performance and characteristics.
3. **Theory:** Breadth-First Search (BFS) is a graph traversal algorithm used to explore and visit all nodes of a graph or tree in a systematic manner. The key characteristic of BFS is that it explores nodes level by level, starting from the initial node (often called the "source" node) and moving outward in a breadth-wise fashion. BFS can be implemented using a queue data structure to keep track of the nodes to visit. It's important to mark nodes as visited to avoid revisiting them and to prevent infinite loops in the case of cyclic graphs.

4. Script:

```
graph = {  
  1: [2, 3],  
  2: [1, 4, 5],  
  3: [1, 5],  
  4: [2, 6],
```

```
        5: [2, 3],
        6: [3]
    }

    def bfs(node):        visited = [False] *
        (len(graph) + 1)    queue = []
    visited[node] = True
    queue.append(node)
        while
    queue:
        v = queue.pop(0)
    print(v, end=" ")
        for neigh in graph[v]:
    if not visited[neigh]:
        visited[neigh] = True
        queue.append(neigh)

    print("Breadth-First Search:") if
    __name__ == "__main__":
        bfs(1)
```

5. Output:

```
Breadth-First Search:
1 2 3 4 5 6
```

6. Learning Outcomes:

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



Discover. Learn. Empower.

1. Understood the order in which nodes are traversed during BFS.
2. Learned the BFS algorithm, a graph traversal method that explores nodes level by level.
3. Learned how to use a queue data structure to keep track of the nodes to visit in BFS. Nodes are enqueued as they are visited, and dequeued when their neighbors are explored.