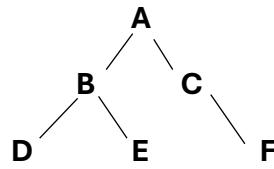


WEEK – 3

BFS and DFS Algorithm



Breadth First Search:

Start from A and visit all nodes level by level (left to right)

A -> B -> C -> D-> E-> F

CODE:

```
#include <stdio.h>
#define MAX 10
int graph[MAX][MAX], visited[MAX];
int queue[MAX], front = -1, rear = -1;

void bfs(int start, int n)
{
    int i;
    queue[++rear] = start;
    visited[start] = 1;
    while (front != rear)
    {
        int current = queue[++front];
        printf("%c ", current + 'A');
        for (i = 0; i < n; i++)
        {
            if (graph[current][i] == 1 && visited[i] == 0)
            {
                queue[++rear] = i;
                visited[i] = 1;
            }
        }
    }
}
```

```
    }
}

int main()
{
    int n, i, j, start;
    printf("Enter number of nodes: ");
    scanf("%d", &n);
    printf("Enter adjacency matrix:\n");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%d", &graph[i][j]);
        }
        visited[i] = 0;
    }
    printf("Enter starting node (0 for A, 1 for B, ...): ");
    scanf("%d", &start);
    printf("BFS Traversal: ");
    bfs(start, n);
    return 0;
}
```

Output:

```
Enter number of nodes: 6
Enter adjacency matrix:
0 1 1 0 0 0
1 0 0 1 1 0
1 0 0 0 0 1
0 1 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0
Enter starting node (0 for A, 1 for B, ...): 0
BFS Traversal: A B C D E F
```

Depth First Search:

Start from A and go to depth of the tree (left to right)

A -> B -> D -> E -> C -> F

CODE:

```
#include <stdio.h>
#define MAX 10
int graph[MAX][MAX], visited[MAX];
void dfs(int node, int n)
{
    int i;
    visited[node] = 1;
    printf("%c ", node + 'A'); // Print node
    for (i = 0; i < n; i++)
    {
        if (graph[node][i] == 1 && visited[i] == 0)
        {
            dfs(i, n);
        }
    }
}
int main()
{
    int n, i, j, start;
    printf("Enter number of nodes: ");
    scanf("%d", &n);
    printf("Enter adjacency matrix:\n");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%d", &graph[i][j]);
        }
        visited[i] = 0;
    }
    printf("Enter starting node (0 for A, 1 for B, ...): ");
    scanf("%d", &start);
```

```
    printf("DFS Traversal: ");
    dfs(start, n);
    return 0;
}
```

Output:

```
Enter number of nodes: 6
Enter adjacency matrix:
0 1 1 0 0 0
1 0 0 1 1 0
1 0 0 0 0 1
0 1 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0
Enter starting node (0 for A, 1 for B, ...): 0
DFS Traversal: A B D E C F
```