

---

**Department of Computer science and Engineering**

**PES UNIVERSITY**

**UE19CS202: Data Structures and its Applications (4-0-0-4-4)**

**GRAPHS**

**Abstract**

**Implementation of Graph using Adjacency Matrix**

**Dr.Sandesh and Saritha**

**Sandesh\_bj@pes.edu**

**Saritha.k@pes.edu**

## Implementation of graph using adjacency matrix

### C representation of graph:

A graph with 25 nodes can be declared as

```
#define MAX 25
```

```
struct node
```

```
{
```

```
    //information associated with each node
```

```
};
```

```
struct arc
```

```
{
```

```
    int adj; //information associated with each arc
```

```
};
```

```
struct graph
```

```
{
```

```
    struct node nodes[MAX];
```

```
    struct arc arcs[MAX][MAX];
```

```
};
```

```
struct graph g;
```

Each node in a graph is represented by an integer number from zero to MAX-1, and the array field nodes represent the appropriate information assigned to each node. The array field an arc is a two dimensional array representing every possible ordered pair of nodes.

---

## The different operations performed on adjacency matrix are

### 1. To add an arc from node1 to node2

```
void join(int adj[][MAX],int node1,int node2)
{
    adj[node1][node2]=TRUE;
}
```

### 2. To delete an arc from node1 to node2 if it exists

```
void remv(int adj[][MAX],int node1,int node2)
{
    adj[node1][node2]=FALSE;
}
```

### 3. To check whether arc exists between node1 and node2

```
int adjacent(int adj[][MAX],int node1,int node2)
{
    return((adj[node1][node2]==TRUE)?TRUE:FALSE);
}
```