

## ADA LAB-TEST 2

USN: IBM19CS168

Name: Swetha Patil

Date: 05/07/2021

4<sup>th</sup> sem 'D' section

### Floyd's algorithm:-

```
#include <stdio.h>
#define infinity 1000000
#define max 100
int n, adj[max][max], D[max][max],
    pred[max][max];

void creategraph();
void Floyd's();
void findpath(int s, int d);
void display(int matrix[max][max], int n);

int main()
{
    int s, d;
    Floyd's();
    while(1)
    {
        printf("Enter source vertex");
        scanf("%d", &s);
        printf("Enter destination vertex");
        scanf("%d", &d);
        printf("Shortest path is");
        findpath(s, d);
    }
}

void Floyd's()
{
    int i, j, k;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
        {
            if (adj[i][j] == 0)
            {
                D[i][j] = infinity;
                pred[i][j] = -1;
            }
            else
            {
                D[i][j] = adj[i][j];
                pred[i][j] = i;
            }
        }
}
```

```

for (k=0; k<n; k++)
{
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            if (D[i][j] > D[i][k] + D[k][j])
                D[i][j] = D[i][k] + D[k][j];
}
}

```

```

printf("Shortest path matrix is: \n");
display(D, n);
}

```

```

display findPath(int s, int d)
{
    int i, path[max], count;
    if (D[s][d] == infinity)
    {
        printf("no path");
        return;
    }
    count = -1;
    do {
        path[++count] = d;
        d = pred[s][d];
    } while (d != s);
    path[++count] = s;
    for (i=count; i>=0; i--)
        printf("%d ", path[i]);
}

```

```

void display (int matrix[max][max], int n)
{
    int i, j;
    for (i=0; i<n; i++)
    {
        for (j=0; j<n; j++)
            printf("%d ", matrix[i][j]);
    }
}

```

```

createGraph() {
    printf("Enter the no. of vertices);
    scanf("%d", &n);
    printf("Enter adjacency matrix)
}

```

```
for (int i = 0; i < n; i++)  
    for (i int j = 0; j < n; j++)  
        scanf("%f", &adj[i][j]);  
}
```