

Name : Bijay Regmi
Roll Number : 210913032

Week 4[Dynamic Source Routing (DSR)]

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define inf 999
#define max 10
static int count = 0;
struct rreq
{
    int uniq_id;
    char nodeList[20];
    int src;
    int dest;
};
void dsr(int graph[max][max], int s, int d, int n, struct rreq *rqpack)
{
    if (s == d)
    {
        if (count != 1)
        {
            char node_str[5];
            sprintf(node_str, "%d", d);
            strcat(rqpack->nodeList, node_str);
            printf("\n path to destination %s", rqpack->nodeList);
            count = 1;
            return;
        }
    }
    for (int i = 0; i < n; i++)
    {
        if (graph[s][i] != 999 && graph[s][i] != 0)
        {
            for (int j = 0; j < n; j++)
            {
```

```

        if (graph[i][j] != 999 && j != s && graph[i][j] != 0)
        {
            char node_str[5];
            sprintf(node_str, "%d", i);
            strcat(rqpack->nodeList, node_str);
            printf("\n inter path %s", rqpack->nodeList);
            dsr(graph, j, d, n, rqpack);
        }
    }
}
return;
}

```

```

int main()
{
    int n, graph[max][max], i, j, s, d, choice;

    printf("Enter the number of vertices: \n");
    scanf("%d", &n);

    printf("Enter the adjacent matrix");

    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &graph[i][j]);

    printf("The initial cost matrix is: \n");

    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            printf("%d\t\t", graph[i][j]);
        }
        printf("\n");
    }

    printf("Enter the Source node : \n");
    scanf("%d", &s);
}

```

```
printf("Enter the destination node : \n");
scanf("%d", &d);
// dijkstras(graph,s,n);
struct rreq *rqpack;
rqpack = (struct rreq *)malloc(sizeof(struct rreq));
rqpack->uniq_id = 20;
rqpack->nodeList[20];
rqpack->src = s;
rqpack->dest = d;
char src_str[5];
sprintf(src_str, "%d", s);
strcat(rqpack->nodeList, src_str);
printf("uniq_id=%d\n", rqpack->uniq_id);
printf("nodeList=%s\n", rqpack->nodeList);
printf("src=%d\n", rqpack->src);
printf("dest=%d\n", rqpack->dest);
dsr(graph, s, d, n, rqpack);

return 0;
}
```

OUTPUT

```
Enter the number of vertices:
3
Enter the adjacent matrix0
1
999
1
0
1
999
1
0
The initial cost matrix is:
0          1          999
1          0          1
999        1          0
Enter the Source node :
0
Enter the destination node :
2
uniq_id=20
nodeList=0
src=0
dest=2

inter path 01
[ path to destination 012%
```