Lab 9

Bellmann Ford Alogrithm

Aim: Find suitable path for transmission using Bellmann ford algorithm.

```
#include <stdio.h>
#include <stdlib.h>

int Bellmon Ford (int G[20][20], int V, int E, int edge
 [20][20]) {
int i, u, v, k, distance [20], parent [20], s, flag=1;
for (i=0; i<V; i++) {
    distance [i] = 1000;
    parent [i] = -1; }
    printf ("Enter size ");
    scanf ("%d", &s);
    distance [s-1] = 0;
    for (i=0; i<V-1; i++) {
        for (k=0; k<E; k++) {
            u=edge [k][0];
            v = edge [k][1];
            if (distance [u] + G[u][v] < distance [v])
                distance [v] = distance [u] + G[u][v];
                parent [v] = u;
        }
}
```

```
for (k=0; k<E; k++ ){

    u = edge [k][0]
    v = edge [k][1]
   if (distance[u] + G[u][v] < distance[v] )

       flag = 0
   }

   if (flag)
     for(i=0; i<v; i++)
       printf ( " Vector %d → cost = %d parent = %d \n "
       i+1, distance[i], parent[i]+1 );
       return flag ;
   }

int main (){

   int v, edge [20][2], G[20][20], i, j, k = 0;
   printf ( " Enter no. of vertices " );
   scanf ( " %d ", &v );
   printf ( " Enter graph in matrix form : \n " );
   for (i=0; i<v ; i++)
       for (j=0; j<v; j++ ) {
           scanf ( " %d ", & G[i][j]);
           if ( G[i][j] != 0)
           edge [k][0] = i ;
           edge [k+1][1] = j ;

   }
```

```
if (Bellman_ford(G, V, E, edge))
    printf ("\n No negative weight cycle \n");
else
    printf ("\n Negative weight cycle exists \n");
    return 0;
}
```

Output

```
Enter no. of vertices : 5
Enter graph in matrix:
0  6  0  7  0
0  0  5  8  -4
0  -2 0  0  0
0  0  -3
```

FINAL OUTPUT

```
PS C:\Users\Manoj\OneDrive\Desktop> cd "c:\Users\Manoj\OneDrive\Desktop\" ; if ($?) { g++ Unti
tled-1.cpp -o Untitled-1 } ; if ($?) { .\Untitled-1 }
Enter the number the routers(<10): 5
Enter 1 if the corresponding router is adjacent to routerA else enter 99:
 B C D E
Enter matrix:1 1 99 99

Enter 1 if the corresponding router is adjacent to routerB else enter 99:
 A C D E
Enter matrix:99 99 1 99

Enter 1 if the corresponding router is adjacent to routerE else enter 99:
 A B C D
Enter matrix:99 99 1 99

Router Table entries for router A:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 0 1 1 99 99

Router Table entries for router B:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 1 0 99 99 99

Router Table entries for router C:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 1 99 0 1 1

Router Table entries for router D:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 99 99 1 0 99

Router Table entries for router E:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 99 99 1 99 0
```