**Ravi sajjanar**

**1BM19CS127**

**PROGRAM 13: Floyd’s algorithm**

Implement All Pair Shortest paths problem using Floyd’s algorithm.

**ALGORITHM:** floyds(a[1….n,1….n])

//Implements Floyd’s algorithm for all-pairs shortest path problem

//Input: cost matrix a[1….n,1….n] of size nXn

//Output: Shortest distance matrix a[1….n,1….n] of size nXn

**for** k🡨1 to n **do**

**for** i🡨1 to n **do**

**for** j🡨1 to n **do**

            a[i,j]🡨min(a[i,j],a[i,k]+a[k,j])

**end for**

**end for**

**end for**

write ‘all pair shortest path matrix is’

**for** i🡨1 to n **do**

**for** j🡨1 to n **do**

        write a[i,j]

**end for**

**end for**

**Program:**

#include<stdio.h>

#include<conio.h>

int a[10][10],n;

void floyds();

int min(int,int);

void main()

{

 int i,j;

 clrscr();

 printf("\nenter the no. of vertices:\t");

 scanf("%d",&n);

 printf("\nenter the cost matrix:\n");

 for(i=1;i<=n;i++)

 {

  for(j=1;j<=n;j++)

  {

   scanf("%d",&a[i][j]);

  }

 }

 floyds();

 getch();

}

void floyds()

{

 int i,j,k;

 for(k=1;k<=n;k++)

 {

  for(i=1;i<=n;i++)

  {

   for(j=1;j<=n;j++)

   {

    a[i][j]=min(a[i][j],a[i][k]+a[k][j]);

   }

  }

 }

 printf("\nall pair shortest path matrix is:\n");

 for(i=1;i<=n;i++)

 {

  for(j=1;j<=n;j++)

  {

   printf("%d\t",a[i][j]);

  }

  printf("\n\n");

 }

}

int min(int x,int y)

{

 if(x<y)

 {

  return x;

 }

 else

 {

  return y;

 }

}

**Output**

Enter the no. of vertices:  4

Enter the cost matrix:

9999   9999         3   9999

      2   9999   9999   9999

9999         7   9999         1

      6   9999   9999   9999

All pair shortest path matrix is:

10     10      3      4

  2     12      5      6

  7       7    10      1

  6     16      9    10



