23/11/20

Divyank Gupta
1BM18CS030

Write-Up

④ Code for Distance-Vector Algorithm

```
import java.util.*;
import java.io.*;

class DistanceVector
{
    static int graph [][];
    Static int via [][];
    static int rt [][];
    static int hop-count [][];
    Static int v, e ;

    public static void main (String args[]) throws IOException
    {
      BufferedReader br = new BufferedReader (new InputStreamReader (System.in))
      Scanner sc= new Scanner (System.in);
      System.out.println (" Please enter the number of Routers: ");
      v = Integer.parseInt (br.readLine ());

      System.out.println (" Please enter total no. of connections");
      e = Integer.parseInt (br.readLine());

      graph = new int [v][v];
      hop.count = new int [v][v];
      via = new int [v][v];
      rt = new int [v][v];
```

[illegible]

Divyank Gupta

1BM18CS079

Write-Up

Continued...

```
System.out.println("Please Enter cost to fill the matrix");
for(int i=0; i<v; i++)
{
   for(int j=0; j<v; j++)
   {
      int c= sc.nextInt();
      graph[i][j] = c;
   }
}
dvr_calc_disp ("The Initial Routing Tables are: ");
int choice = 0;
while (choice != -1)
{
   System.out.print ("Please enter source node whose cost has changed");
   int s= sc.nextInt();
   s--;
   System.out.print ("Please enter dest node whose cost has changed");
   int d= sc.nextInt();
   d--;
   System.out.print ("Please enter the new cost ");
   int c = sc.nextInt();
   graph [s][d] = c;
   graph [d][s] = c;
   dvr_calc_disp ("The new Routing Tables are: ");
   System.out.println ("Enter -1 to exit or any other number to continue")
   choice = sc.nextInt();
}
```

[illegible]

Divyank Gupta
1BM18CS030

Write-Up

Continued ...

```
static void dvr_calc_disp (String message)
{
    System.out.println();
    init_tables();
    update_tables();
    System.out.println(message);
    print_tables();
    System.out.println();
}

static void update_table (int source)
{
  for (int i=0; i<v ; i++)
  {
    if (graph[source][i] != 9999)
    {
      int dist = graph [Source][i];
      for (int j=0; j<v; j++)
      {
        int inter_dist = rt[i][j];
        if (via [i][j] == source)
          inter_dist = 9999;
        if (dist + inter_dist < rt [source][j])
        {
          rt [source][j] = dist + inter_dist;
          via [source][j] = i;
          hop_count [source][j] ++;
        }
      }
    }
  }
}
```

[illegible]

23/11/20

Divyansh Gupta
1BM18CS030

Write-Up

Continued...

```
static void update_tables()
{
  int k=0;
  for(int i=0; i<4*v; i++)
  {
     update_table(k);
     k++;
     if(k==v)
        k=0;
  }
}

static void init_tables()
{
  for(int i=0; i<v; i++)
  {
    for(int j=0; j<v; j++)
    {
      if(i==j)
      {
        rt[i][j]=0;
        via[i][j]=i;
      }
      else
      {
        rt[i][j]=9999;
        via[i][j]=100;
      }
    }
}}
```

[illegible]

23/11/20

Divyank Gupta
1BM18CS030

Write-up

Continued...

```
static void print_tables()
{
    for(int i=0; i<v; i++)
    {
        for (int j=0; j<v; j++)
        {
            System.out.print((i+1)+"to"+(j+1)+"    "+"Cost "+rt[i][j]+"");
            system.out.print("Hop Count: "+ hop_count[i][j]+"    ");
        }
        system.out.println();
    }
}
```

[illegible]