```java
package Group8.MainClass;
import java.util.*;
public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Class c = new Class();
        System.out.print("Input number of node : ");
        int n = sc.nextInt();
        int[][] weight = new int[n][n];
        String[] temp = new String[n];
        String[] name = new String[n];
        name[0] = "u0";
        for(int y=1;y<n;y++){
            name[y] = "v"+y;
        }
        for(int y=0;y<n;y++){
            System.out.print("Node "+name[y]+" : ");
            String node = sc.next();
            temp = node.split(",");
            for(int z=0;z<n;z++){
                weight[y][z] = Integer.valueOf(temp[z]);
            }
            temp = null;
        }
        System.out.println("\n
  ----------------------------------------------------------------------------------------");
        int visit[] = new int[n];
        visit[0] = 1;

        int stack[] = new int[n];
        stack[0] = 0;
        int stackw[] = new int[n];
        stackw[0] = 0;
        int minw = -1;
        int minad = 0;
        int minadwint = 0;
        int minadw[] = new int[n];
        String path[] = new String[n];

        System.out.printf("%8s", "l(u0)");
        for(int y=1;y<name.length;y++){
            System.out.printf("%15s", name[y]);
        }
        System.out.printf("%15s\n","Add to S");
        System.out.printf("%8s", "0");
        for(int y=1;y<name.length;y++){
            System.out.printf("%15s", "(INFI,-)");
        }
        System.out.printf("%15s\n","u0");

        for (int i = 0; i < n - 1; i++) {
            System.out.printf("%8s", "");
            int w[] = new int[n];
            int wnode[] = new int[n];

            for (int j = 1; j < n; j++) {
                if (visit[j] == 1 && j != 0) {
                    System.out.printf("%15s", "");
                } else if (visit[j] == 0) {
                    for (int k = 0; k < stack.length; k++) {
                        if(i>0){
                            if (weight[j][stack[k]] != -1) {
                                w[k] = weight[j][stack[k]]+stackw[stack[k]];
                            } else if (weight[j][stack[k]] == -1) {
                                w[k] = -1;
                            }
                        }
                    }
                    if(i==0){
                        for (int k = 0; k < stack.length; k++) {
                            w[k] = weight[j][stack[k]];
                        }
                    }
                    minw = -1;
                    boolean chk = false;
                    for (int l=0;l<w.length;l++) {
                        if (w[l] > -1 && visit[j] == 0) {
                            chk = true;
                            minw = w[l];
                        }
                    }


```

```
                    if (chk) {
                        for (int l : w) {
                            if (visit[j] == 0) {
                                if(l<minw && l!=-1){
                                    minw = l;
                                }
                            }
                        }
                    }

                    wnode[j] = minw;
                    minadw[j] = c.findindex(minw, w);
                    if (wnode[j] == -1) {
                        System.out.printf("%15s", "(INFI,-)");
                    } else {
                        System.out.printf("%15s",("("+wnode[j]+","+name[stack[minadw[j]]]+")"));
                    }
                }
            }

            for (int q = 0; q < wnode.length; q++) {
                if (wnode[q] != -1 && visit[q] == 0) {
                    c.minn(wnode[q]);
                }
            }
            int minn = c.getminn();
            for (int q = 0; q < wnode.length; q++) {
                if (wnode[q] < c.getminn() && wnode[q] != -1 && visit[q] == 0) {
                    c.minn(wnode[q]);
                }
            }

            minad = c.findindex(c.getminn(), wnode);
            minadwint = minadw[minad];
            visit[minad] = 1;
            stack[i + 1] = minad;
            System.out.printf("%15s  ", name[minad]);

            stackw[minad] = stackw[stack[minadwint]]+weight[minad][stack[minadwint]];
            path[i] = name[stack[minadwint]]+"->"+name[minad];
            System.out.println("");
        }

        System.out.print("\nPath : ");
        for(int t=0;t<path.length-1;t++){
            System.out.print(path[t]+" , ");
        }
        System.out.println("\b\b\b");
        System.out.println(
    "--------------------------------------------------------------------------------------------");
    }
}

```

```java
package Group8.MainClass;

public class Class {
    private int minn;
    private int minad;
    private int min;

    public void min(int min){

        this.min = min;
    }

    public int getmin(){

        return min;
    }

    public void minn(int minn){

        this.minn = minn;
    }

    public int getminn(){

        return minn;
    }

    public void minad(int minad){

        this.minad = minad;
    }

    public int getminad(){

        return minn;
    }

    public int findindex(int m,int[] n){
        int index = 0;
        int i = 0;
        boolean chk = false;
        for(i=0;i<n.length;i++){
            if(n[i] == m) {
                index = i;
                chk = true;
            }
            if(chk) break;
        }
        return index;
    }
}

```