prims_algorithm

Problem Submissions Leaderboard Discussions

Find Minimum Cost Spanning Tree of a given connected undirected graph using Find Minimum Cost Spanning Tree using Prim's algorithm.

Input Format

7 0 28 999 999 999 10 999 28 0 16 999 999 999 14 999 16 0 12 999 999 999 999 999 12 0 22 999 18 999 999 999 22 0 25 24 10 999 999 999 25 999 999 999 14 999 18 24 999 999

Constraints

No Constraints

Output Format

1edge(1,6)=10 2edge(6,5)=25 3edge(5,4)=22 4edge(4,3)=12 5edge(3,2)=16 6edge(2,7)=14 The minimum cost of spanning tree is 99

Sample Input 0

```
7
0 28 999 999 999 10 999
28 0 16 999 999 999 14
999 16 0 12 999 999 999
999 999 12 0 22 999 18
999 999 999 22 0 25 24
10 999 999 999 25 999 999
999 14 999 18 24 999 999
```

Sample Output 0

```
1edge(1,6)=10
2edge(6,5)=25
3edge(5,4)=22
4edge(4,3)=12
5edge(3,2)=16
6edge(2,7)=14
The minimum cost of spanning tree is 99
```

f t in

Contest ends in 9 days

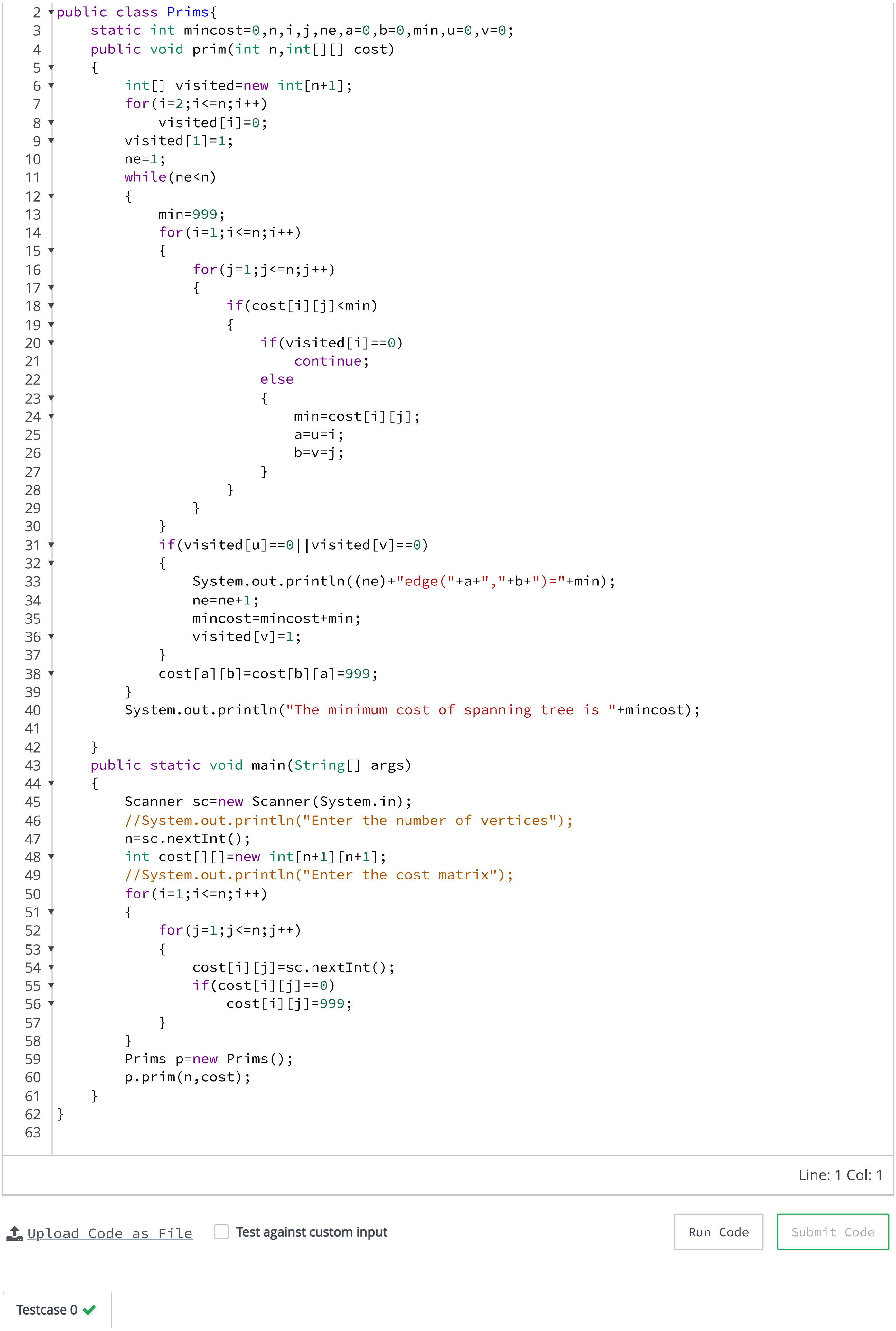
Submissions: 96
Max Score: 10
Difficulty: Medium

Rate This Challenge:
☆☆☆☆☆

More

Java 7

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



```java
public class Prims{
    static int mincost=0,n,i,j,ne,a=0,b=0,min,u=0,v=0;
    public void prim(int n,int[][] cost)
    {
        int[] visited=new int[n+1];
        for(i=2;i<=n;i++)
            visited[i]=0;
        visited[1]=1;
        ne=1;
        while(ne<n)
        {
            min=999;
            for(i=1;i<=n;i++)
            {
                for(j=1;j<=n;j++)
                {
                    if(cost[i][j]<min)
                    {
                        if(visited[i]==0)
                            continue;
                        else
                        {
                            min=cost[i][j];
                            a=u=i;
                            b=v=j;
                        }
                    }
                }
            }
            if(visited[u]==0||visited[v]==0)
            {
                System.out.println((ne)+"edge("+a+","+b+")="+min);
                ne=ne+1;
                mincost=mincost+min;
                visited[v]=1;
            }
            cost[a][b]=cost[b][a]=999;
        }
        System.out.println("The minimum cost of spanning tree is "+mincost);

    }
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        //System.out.println("Enter the number of vertices");
        n=sc.nextInt();
        int cost[][]=new int[n+1][n+1];
        //System.out.println("Enter the cost matrix");
        for(i=1;i<=n;i++)
        {
            for(j=1;j<=n;j++)
            {
                cost[i][j]=sc.nextInt();
                if(cost[i][j]==0)
                    cost[i][j]=999;
            }
        }
        Prims p=new Prims();
        p.prim(n,cost);
    }
}
```

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
7
0 28 999 999 999 10 999
28 0 16 999 999 999 14
999 16 0 12 999 999 999
999 999 12 0 22 999 18
999 999 999 22 0 25 24
10 999 999 999 25 999 999
999 14 999 18 24 999 999
```

Your Output (stdout)

```
1edge(1,6)=10
2edge(6,5)=25
3edge(5,4)=22
4edge(4,3)=12
5edge(3,2)=16
6edge(2,7)=14
The minimum cost of spanning tree is 99
```

Expected Output

```
1edge(1,6)=10
2edge(6,5)=25
3edge(5,4)=22
4edge(4,3)=12
5edge(3,2)=16
6edge(2,7)=14
The minimum cost of spanning tree is 99
```