

[All Contests](#) > [DAA_LAB](#) > [prims_algorithm](#)

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 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 2 monthsSubmissions: [76](#)

Max Score: 10

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

Java 7



```
1 import java.util.*;
2 public class Prims{
3     static int mincost=0,n,i,j,ne,a=0,b=0,min,u=0,v=0;
4     public void prim(int n,int[][] cost){
5         int[] visited=new int[n+1];
6         for(i=2;i<=n;i++)
7             visited[i]=0;
8         visited[1]=1;
9         ne=1;
10        while(ne<n){
11            min=999;
12            for(i=1;i<=n;i++){
13                for(j=1;j<=n;j++){
14                    if(cost[i][j]<min){
15                        if(visited[i]==0)
16                            continue;
17                        else{
18                            min=cost[i][j];
19                            a=u=i;
20                            b=v=j;
21                        }
22                    }
23                }
24            }
25            if(visited[u]==0||visited[v]==0){
26                System.out.println((ne)+"edge("+a+","+b+")="+min);
27                ne=ne+1;
28                mincost=mincost+min;
29                visited[v]=1;
30            }
31            cost[a][b]=cost[b][a]=999;
32        }
33        System.out.println("The minimum cost of spanning tree is "+mincost);
34    }
35 }
36 public static void main(String[] args){
37     Scanner sc=new Scanner(System.in);
38     //System.out.println("Enter the number of vertices");
39     n=sc.nextInt();
40     int cost[][]=new int[n+1][n+1];
41     //System.out.println("Enter the cost matrix");
42     for(i=1;i<=n;i++){
43         for(j=1;j<=n;j++){
44             cost[i][j]=sc.nextInt();
45             if(cost[i][j]==0)
46                 cost[i][j]=999;
47         }
48     }
49     Prims p=new Prims();
50     p.prim(n,cost);
51 }
52 }
```

Line: 52 Col: 2

 Upload Code as File ☐ Test against custom input

Run Code

Submit Code

Testcase 0 

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
```