Dijkstra's Algorithm 4

From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm. Write the program in Java.

Input Format

4 0 15 10 9999 9999 0 15 9999 20 9999 0 20 9999 10 9999 0 3

Constraints

--

Output Format

Shortest path from 3 to all other vertices To 0 is 45 To 1 is 10 To 2 is 25 To 3 is 0

Sample Input 0

```
4
0 15 10 9999
9999 0 15 9999
20 9999 0 20
9999 10 9999 0
3
```

Sample Output 0

```
Shortest path from 3 to all other vertices
To 0 is 45
To 1 is 10
To 2 is 25
To 3 is 0
```

```java
//224G1A0553
import java.util.Arrays;
import java.util.Scanner;
public class Dijkstra {
 static int n,cost[][],i,j,u,dist[],src;
 void dij(int src,int cost[][],int dist[],int n) {
 int visited[],min;
 visited=new int[n];
 for(i=0;i<n;i++) {
 visited[i]=0;
 dist[i]=cost[src][i];
 }
 visited[src]=1;
 dist[src]=0;
 for(i=0;i<n;i++) {
 if(i==src) continue;
 min=999;
 for(j=0;j<n;j++)
 if((visited[j]==0)&&(min>dist[j])) {
 min=dist[j];
 u=j;
 }
 visited[u]=1;
 for(j=0;j<n;j++)
 if(visited[j]==0) {
 if(dist[j]>dist[u]+cost[u][j])
 dist[j]=dist[u]+cost[u][j];
 } } }
 public static void main(String[] args) {
 Scanner sc=new Scanner(System.in);
 //System.out.println("Enter the number of vertices");
 n=sc.nextInt();
 //System.out.println("Enter the matrix");
 cost=new int[n][n];
 dist=new int[n];
 Arrays.fill(dist,0);
 for(i=0;i<n;i++)
 for(j=0;j<n;j++)
 cost[i][j]=sc.nextInt();
 //System.out.println("Enter the source vertex");
 src=sc.nextInt();
 new Dijkstra().dij(src, cost, dist, n);
 System.out.println("Shortest path from "+src+" to all other vertices");
 for(i=0;i<n;i++)
 System.out.println("To " +i+" is "+dist[i]);
 } }
```

Line: 46 Col: 4

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)

```
4
0 15 10 9999
9999 0 15 9999
20 9999 0 20
```

```
9999 10 9999 0
3
```

Your Output (stdout)

```
Shortest path from 3 to all other vertices
To 0 is 45
To 1 is 10
To 2 is 25
To 3 is 0
```

Expected Output

```
Shortest path from 3 to all other vertices
To 0 is 45
To 1 is 10
To 2 is 25
To 3 is 0
```