```java
    private static int INF= (int) Math.pow(10,4)+1;
    public int findTheCity(int n, int[][] edges, int distanceThreshold) {
        int[][] graph= new int[n][n];
        int res=0, smallest= n;
        for(int[] row: graph)
            Arrays.fill(row, INF);

        for(int[] edge: edges)
            graph[edge[0]][edge[1]]= graph[edge[1]][edge[0]]= edge[2];

        for (int i = 0; i < n; ++i)
            graph[i][i] = 0;

        for(int k=0;k<n;k++){
            for(int i=0;i<graph.length;i++){
                for(int j=0;j<graph[i].length;j++){
                    if(graph[i][k]==INF || graph[k][j]==INF) continue;
                    graph[i][j]= Math.min(graph[i][j], graph[i][k]+graph[k][j]);
                }
            }
        }

        for(int i=0;i<n;i++){
            int count= 0;
            for(int j=0;j<graph[i].length;j++)
                if(graph[i][j]<=distanceThreshold)
                    count++;
            if(count<=smallest){
                res= i;
                smallest= count;
            }
        }

        return res;
```

Testcase | Run Code Result | Debugger

Accepted Runtime: 0 ms

Your input	4 [[0,1,3],[1,2,1],[1,3,4],[2,3,1]]
Output	3
Expected	3