计算至少需要多少个快递主站点

时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

快递业务范围有N个站点，A站点与B站点可以中转快递，则认为A-B站可达，如果A-B可达，B-C可达，则A-C可达。现在给N个站点编号0、1、...n-1，用s[i][j]表示i-j是否可达，s[i][j]=1表示i-j可达，s[i][j]=0表示i-j不可达。

现用二维数组给定N个站点的可达关系，请计算至少选择从几个主站点出发，才能可达所有站点（覆盖所有站点业务）。

说明：s[i][j]与s[j][i]取值相同。

输入描述：

第一行输入为N，N表示站点个数。
之后N行表示站点之间的可达关系，第i行第j个数值表示编号为i和j之间是否可达。

输出描述：

输出站点个数，表示至少需要多少个主站点。
补充说明：
1<N<10000

示例1

输入：

```
4
1 1 1 1
1 1 1 0
1 1 1 0
1 0 0 1
```

输出：

```
1
```

说明：

选择0号站点作为主站点，0站点可达其他所有站点，所以至少选择1个站点作为主站才能覆盖所有站点业务。

示例2

输入：

```
4
1 1 0 0
1 1 0 0
0 0 1 0
0 0 0 1
```

输出：

```
3
```

说明：

选择0号站点可以覆盖0、1站点，选择2号站点可以覆盖2号站点，选择3号站点可以覆盖3号站点，所以至少选择3个站点作为主站才能覆盖所有站点业务。

解题思路：

1. **通过回溯法将互通的快递站点放在一个集合中。**
2. **步骤1执行的次数即快递主站点的个数。**

```
public class Main{

    public static int[][] s;

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int N = sc.nextInt();

        s = new int[N][N];
        for(int i=0; i<N; i++){
            for(int j=0; j<N; j++){
                s[i][j] = sc.nextInt();
            }
        }

        Set<Integer> set = new HashSet<>();         //已经有连通的站点
        int res = 0;        //主站点数
        for(int i=0; i<N; i++){
            if(set.contains(i)){       //站点已经有连通，无需再处理
                continue;
            }
            Set<Integer> temp = new HashSet<>();       //相互连通的站点集合
            temp.add(i);
            handle( temp, i);
            set.addAll(temp);      //已经连通的站点放在 set 里面用来判断后续是否需要处
理
            res ++;
        }

        System.out.println(res);
    }

    /**
     * 寻找跟 n 站点相连通的站点
     * @param set      相互连通的站点集合
     * @param n         站点编号
     */
    public static void handle(Set<Integer> set, int n){

        for(int i=0; i<s.length; i++){
            if(set.contains(i)){       //已经连通的站点无需再判断
                continue;
```

```
                }
                if(n != i && s[n][i] == 1){        //值等于 1 说明两站点连通（n==i 就是本站点）
                    set.add(i);
                    handle( set, i);
                }
            }

        }
    }
```