

资源限制

内存限制：256.0MB

C/C++ 时间限制：1.0s

Java 时间限制：3.0s

Python 时间限制：5.0s

问题描述

一个 8×8 的棋盘上有一个马初始位置为 (a, b) ，他想跳到 (c, d) ，问是否可以？如果可以，最少要跳几步？

输入格式

一行四个数字 a, b, c, d 。

输出格式

如果跳不到，输出 -1 ；否则输出最少跳到的步数。

样例输入

```
1 1 2 3
```

样例输出

```
1
```

数据规模与约定

$0 < a, b, c, d \leq 8$ 且都是整数。

解题思路

很简单的一道BFS的题目，只需要从起点开始，每次向可以走的地方走一步，放入队列里即可。

代码如下：

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

public class Main{
    static int[][] map = new int[9][9];
    static boolean[][] st = new boolean[9][9];
    static int[] dx = {-2, -1, 1, 2, 2, 1, -1, -2};
    static int[] dy = {1, 2, 2, 1, -1, -2, -2, -1};
```

```

static int bfs(int a, int b, int c, int d) {
    Deque<int[]> q = new ArrayDeque<>();
    q.offerFirst(new int[]{a, b, 0});
    while (!q.isEmpty()) {
        int[] t = q.pollFirst();
        if (st[t[0]][t[1]]) continue;
        st[t[0]][t[1]] = true;
        if (t[0] == c && t[1] == d) return t[2];
        for (int i = 0; i < 8; i++) {
            int x = t[0] + dx[i];
            int y = t[1] + dy[i];
            if (x <= 0 || y <= 0 || x > 8 || y > 8) continue;
            if (st[x][y]) continue;
            q.offerLast(new int[]{x, y, t[2]+1});
        }
    }
    return -1;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int a = sc.nextInt(), b = sc.nextInt();
    int c = sc.nextInt(), d = sc.nextInt();
    int res = bfs(a, b, c, d);
    System.out.println(res);
    sc.close();
}
}

```