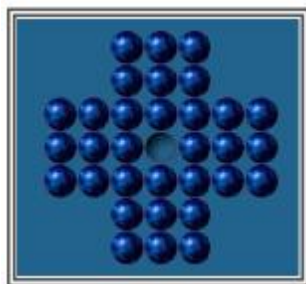


算法实现题 5-14 独立钻石跳棋问题（习题 5-23）

★问题描述：

独立钻石跳棋的棋盘上有 33 个方格，每个方格中可放 1 枚棋子。棋盘中最多可摆放 32 枚棋子。下棋的规则是任一棋子可以沿水平或垂直方向跳过与其相邻的棋子进入空着的方格并吃掉被跳过的棋子。试设计一个算法，对于任意给定的棋盘布局，找出一种下棋方法，使得最终棋盘上只剩下一个棋子。



★编程任务：

对于给定的独立钻石跳棋的棋盘初始布局，和棋盘上最终剩下的棋子所在的位置 (x,y) ，编程计算一种遵循下棋的规则下棋方法，使最终棋盘上仅在位置 (x,y) 处有一枚棋子。当 $(x,y)=(0,0)$ 时，表示不指定棋子的最终位置。棋子位置的坐标定义如下。

		(2,0)	(3,0)	(4,0)		
		(2,1)	(3,1)	(4,1)		
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)	(6,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)	(6,3)
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)	(5,4)	(6,4)
		(2,5)	(3,5)	(4,5)		
		(2,6)	(3,6)	(4,6)		

★数据输入：

由文件 input.txt 提供输入数据，文件的第一行中有 1 个正整数 n ，表示棋盘的初始布局中有 n 个棋子。第 2 行起每行 2 个数，分别是 n 个棋子的位置。最后 1 行是棋盘上最终剩下的棋子所在的位置。

★结果输出:

程序运行结束时，将计算出的下棋步法依次输出到文件 output.txt 中。每行有 2 对方格坐标(a,b)和(c,d)，表示从方格(a,b)跳到方格(c,d)。问题无解时输出 “No” solution!。

输入文件示例

input.txt

8

4 1

5 2

5 3

6 3

3 4

4 4

4 5

4 6

0 0

输出文件示例

output.txt

(3,4) (5,4)

(4,6) (4,4)

(4,4) (6,4)

(6,4) (6,2)

(6,2) (4,2)

(4,1) (4,3)

(5,3) (3,3)

