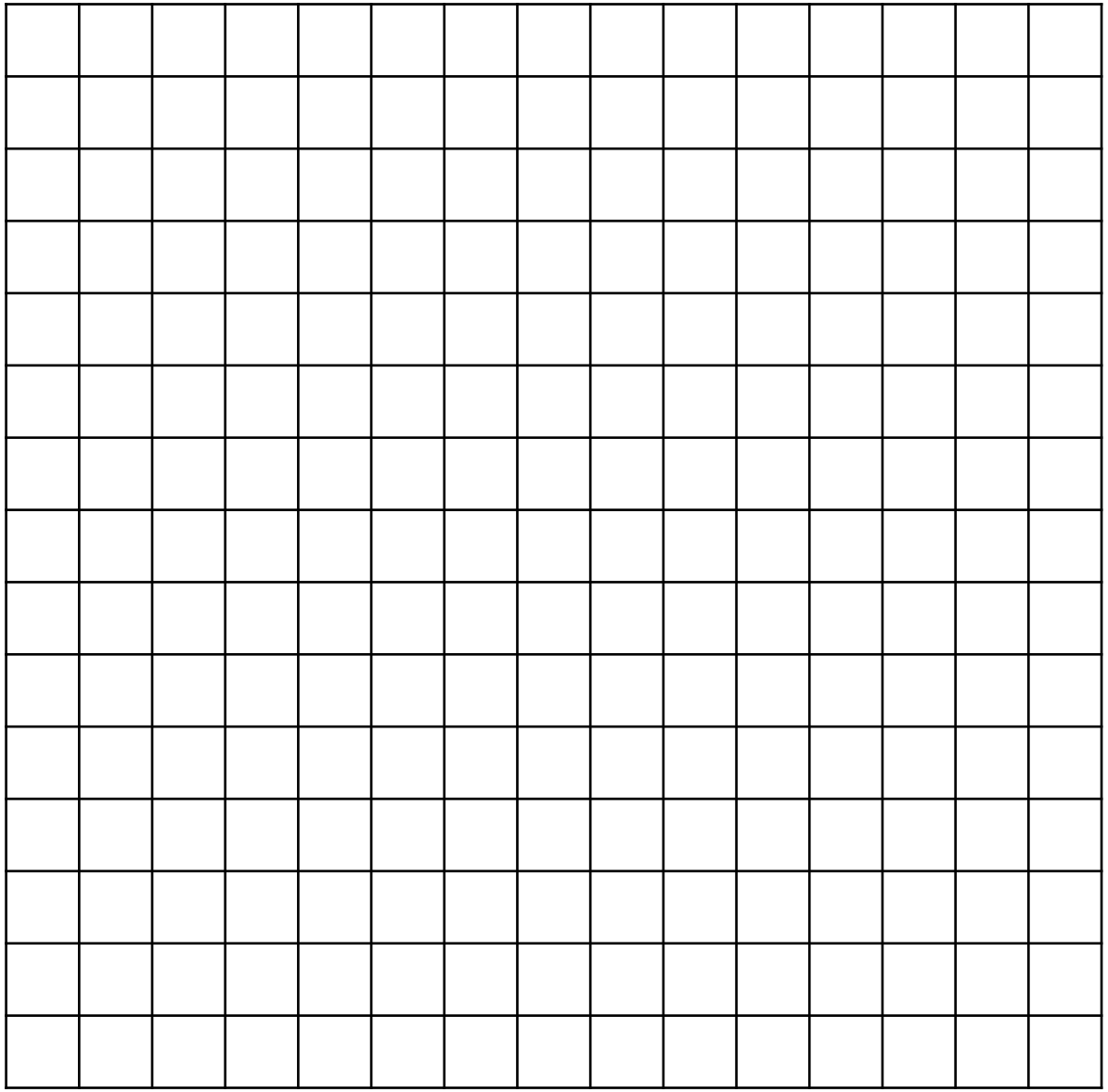
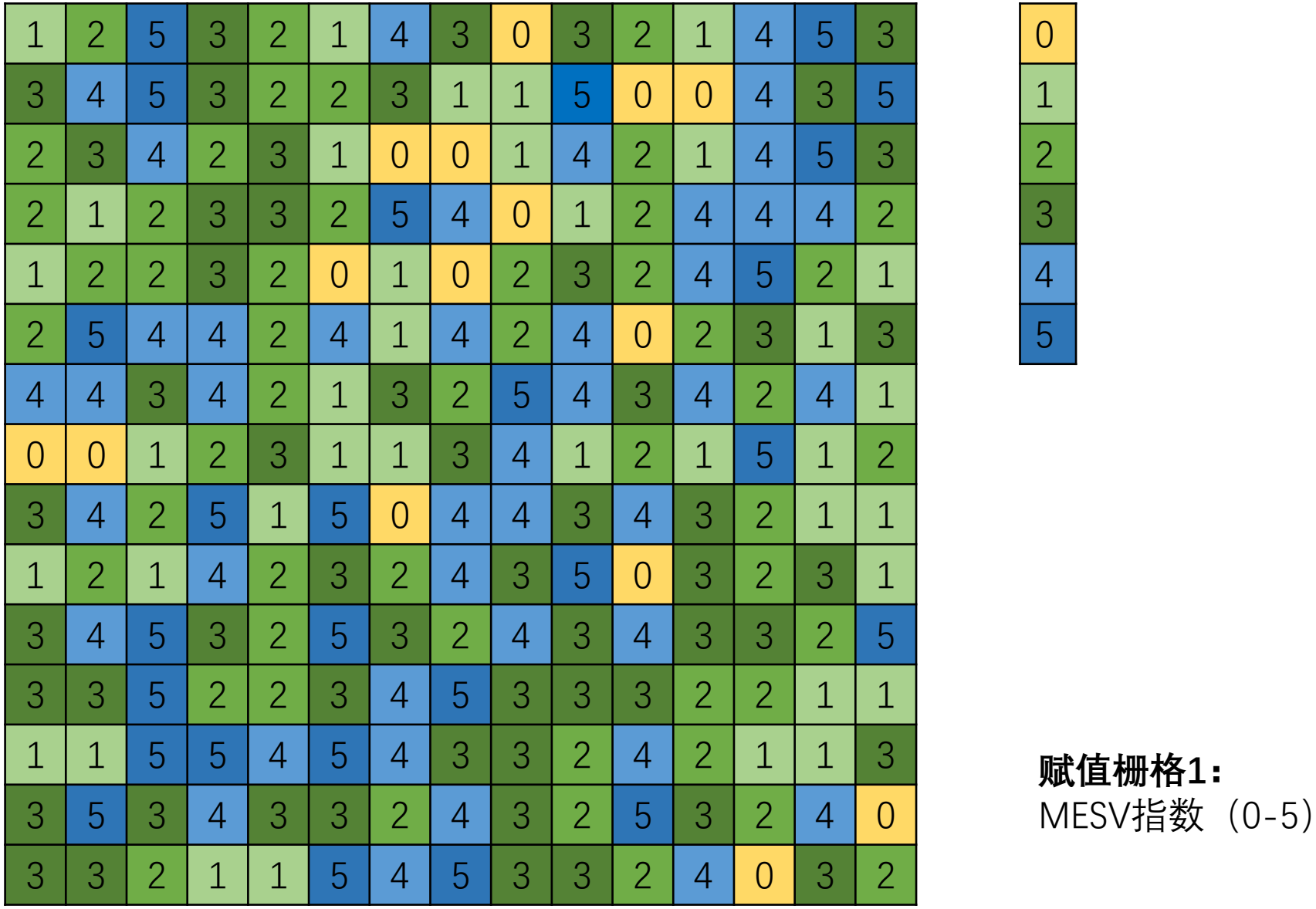


输入数据都是30m×30m的栅格数据

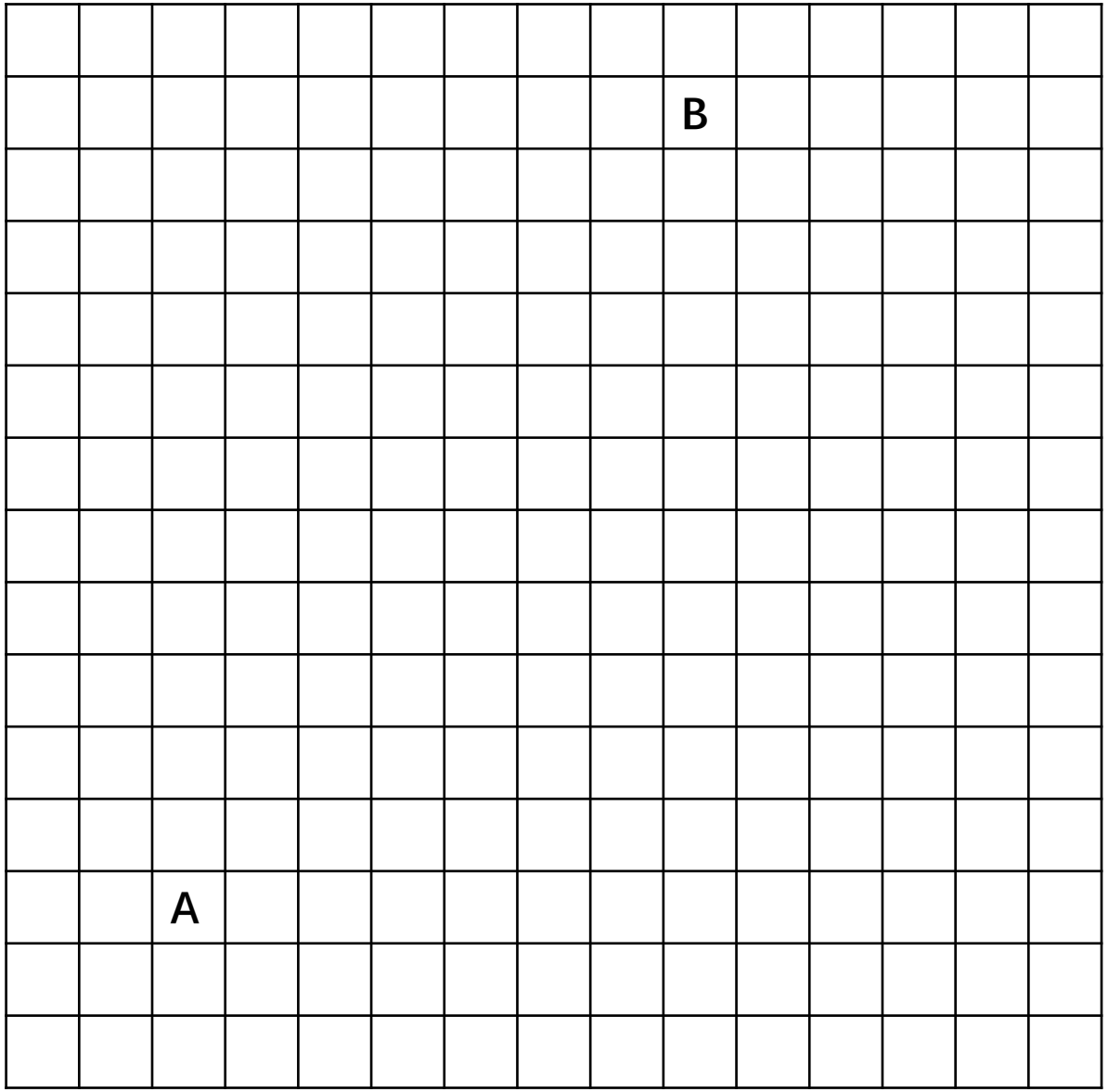


赋值栅格1：
MESV指数（0-5）

1	2	5	4	8	9	1	3	2	4	10	9	8	5	6
2	4	8	9	1	5	4	3	7	3	5	9	8	5	5
5	4	7	10	2	1	5	6	7	8	8	7	5	8	9
10	8	5	4	6	5	7	1	4	10	2	5	7	6	6
5	8	9	9	7	6	4	2	5	10	3	7	8	9	5
1	1	4	5	2	3	6	7	8	7	9	10	5	9	7
5	8	7	9	10	5	8	9	10	7	8	5	6	5	8
9	7	4	8	5	2	1	3	6	6	5	7	8	9	10
10	2	2	4	5	7	8	8	9	5	5	4	2	2	4
1	8	8	7	5	4	2	1	1	2	3	6	9	9	8
2	10	4	4	2	2	1	3	6	5	7	8	9	10	8
2	4	3	5	6	4	2	1	3	7	4	8	5	2	1
1	2	2	3	4	5	4	4	1	5	7	8	9	10	8
2	5	4	1	7	8	7	8	5	2	4	5	4	7	5
9	6	5	10	2	4	5	7	8	2	9	10	2	5	4

1
2
3
4
5
6
7
8
9
10

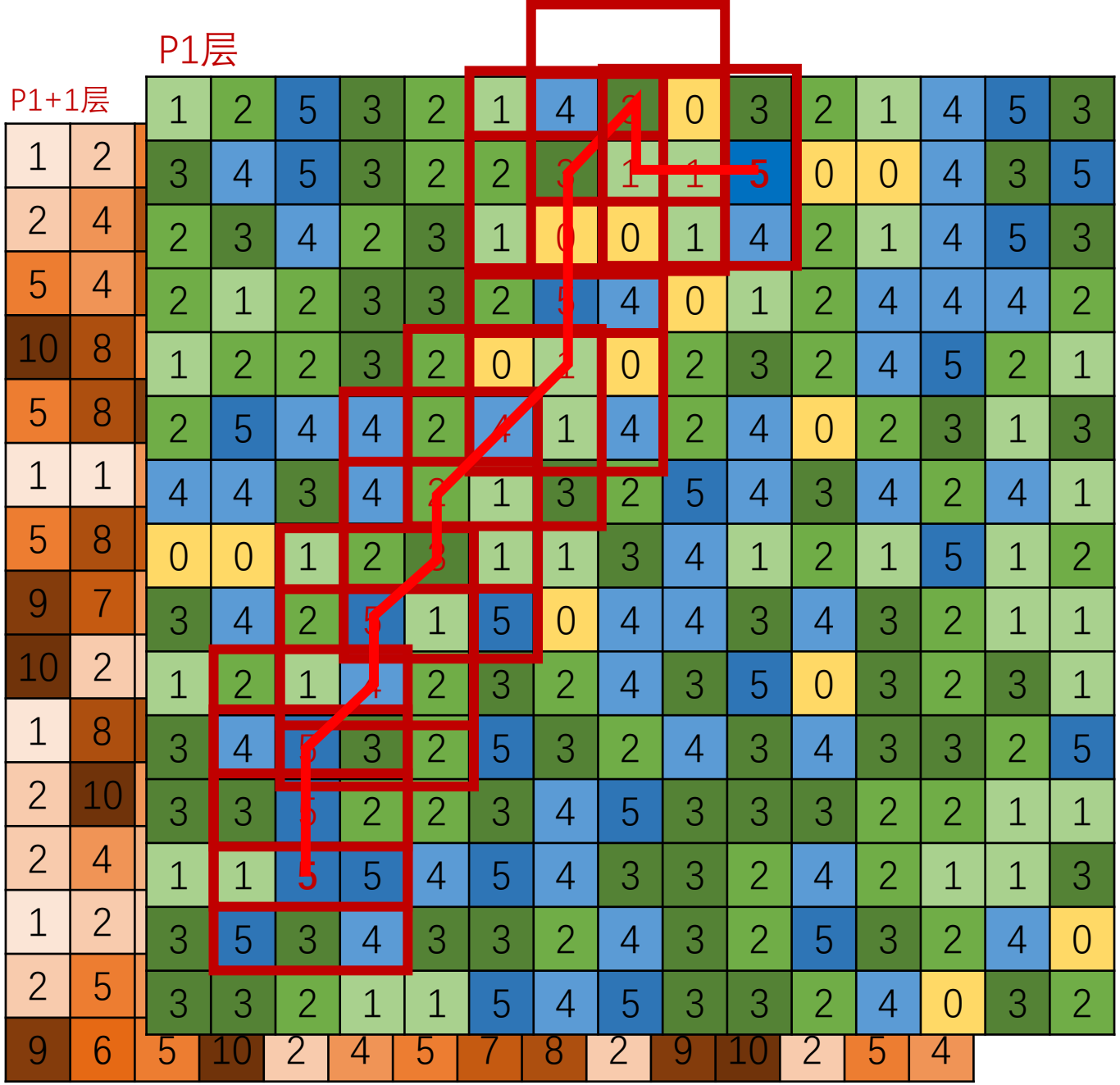
赋值栅格2：
ECR值（1-10）



案例1

从起点A到终点B的路径搜寻

（左下→右上）



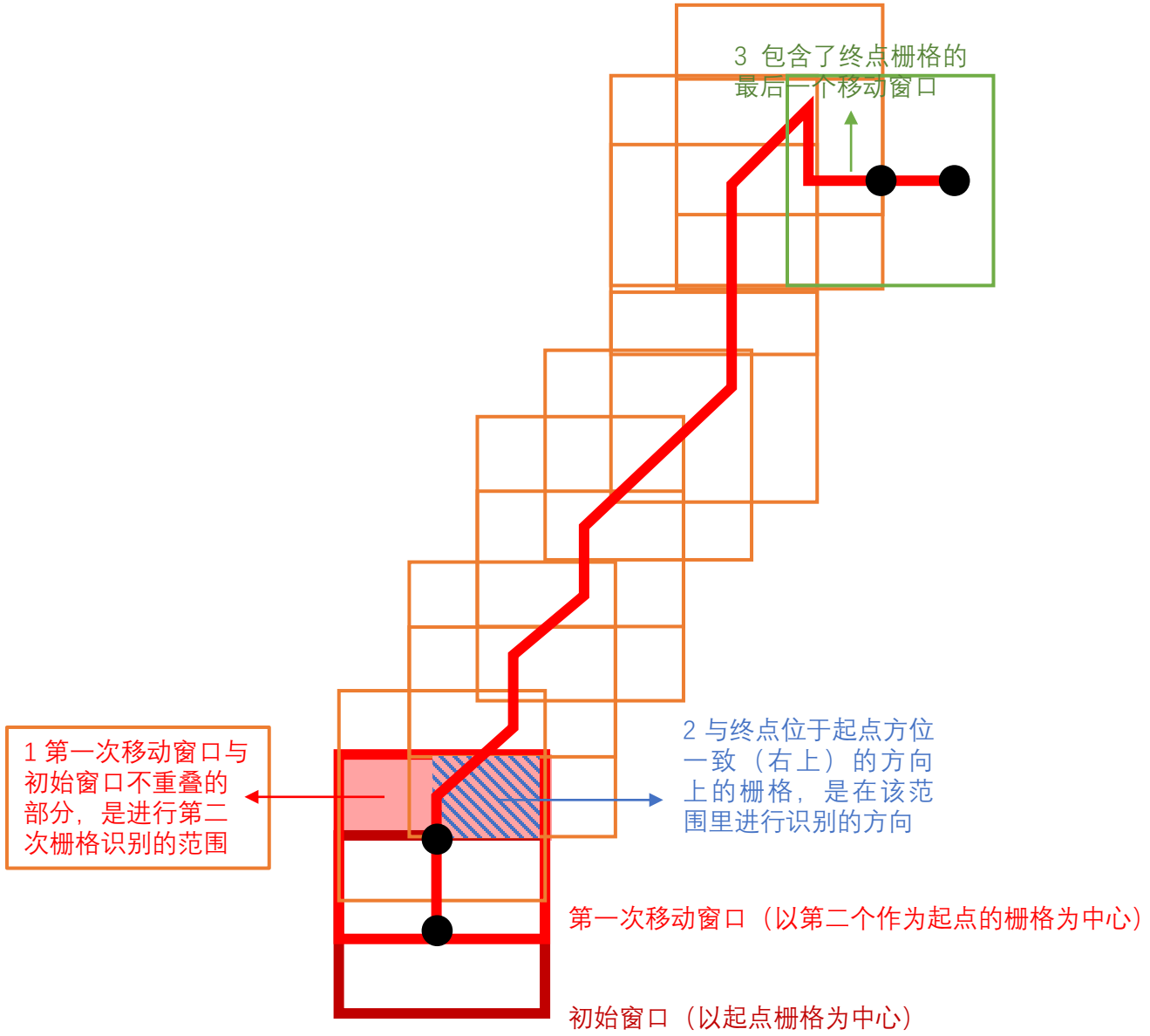
0
1
2
3
4
5

方法规定：1）只有在高层目标已满足但未得到唯一结果的情况下，才考虑继续满足底层目标，直到得到唯一结果；2）在考虑底层目标时，不允许违背已满足的高层目标；3）若满足高层目标后已得到唯一结果，便不再继续考虑底层目标的满足，从而在保证最重要目标被优先满足的同时，简化计算流程、节省分析时间。这种不同层次目标间的优先性可用优先因子P1来表示，其关系为P1≫P1+1，即 P1层目标的满足优先于P1＋1层目标。

具体执行规则：

- P1层目标为选择**MESV指数**最大的栅格；
- P1+1层目标为选择**ECR值**最小的栅格；
- P1+2层目标为选择与目的地栅格**欧氏距离**最近的栅格。

疑问：执行完上述三个规则是否能保证栅格是唯一的？



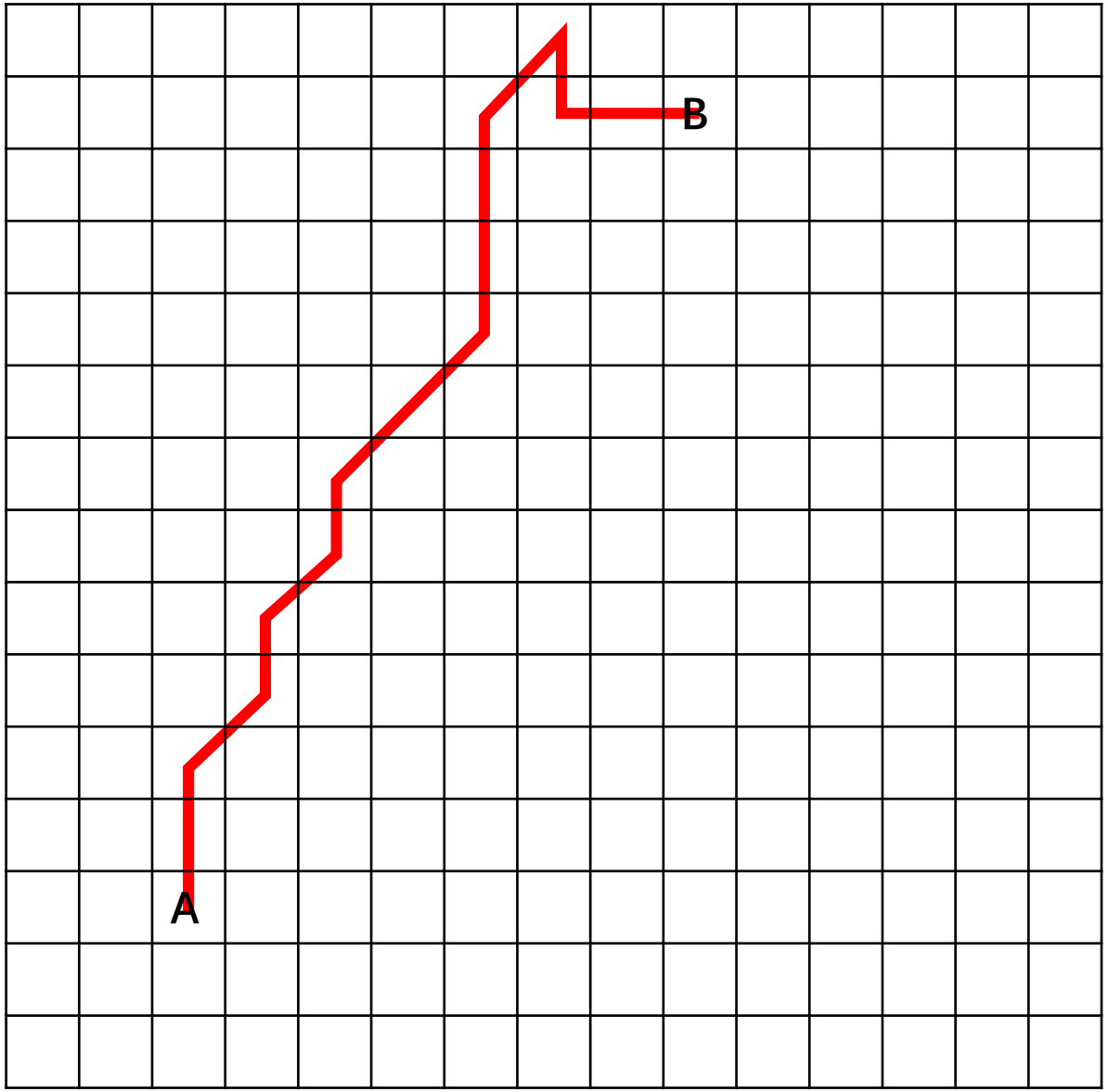
执行三个规则的前提基础：

1）每一次进行贪婪选择的栅格范围：
以每个起始点栅格为中心，用一个3×3的移动窗口对周边栅格进行覆盖；在新的窗口与上一个窗口（若存在）非重叠的部分（左图浅红色区域），执行多层次目标规则下的贪婪选择，从而防止识别的路径走回头路。（即：前一个窗口中包含过的栅格不参与下一次规则中的比较和识别，从而保证每一次移动后执行规则的栅格都是新的。）

2）每一次进行贪婪选择的前进方向：
选择位于与终点相对于每一个新的起点方位一致（左上/左下/右上/右下/左侧/右侧/上侧/下侧）的方向上的栅格进行贪婪选择。

3）进行贪婪选择的结束标志：
当3×3的移动窗口中包含了属于终点的栅格时，将该终点栅格设置为下一步也就是最后一步前进的栅格，并终止搜索。

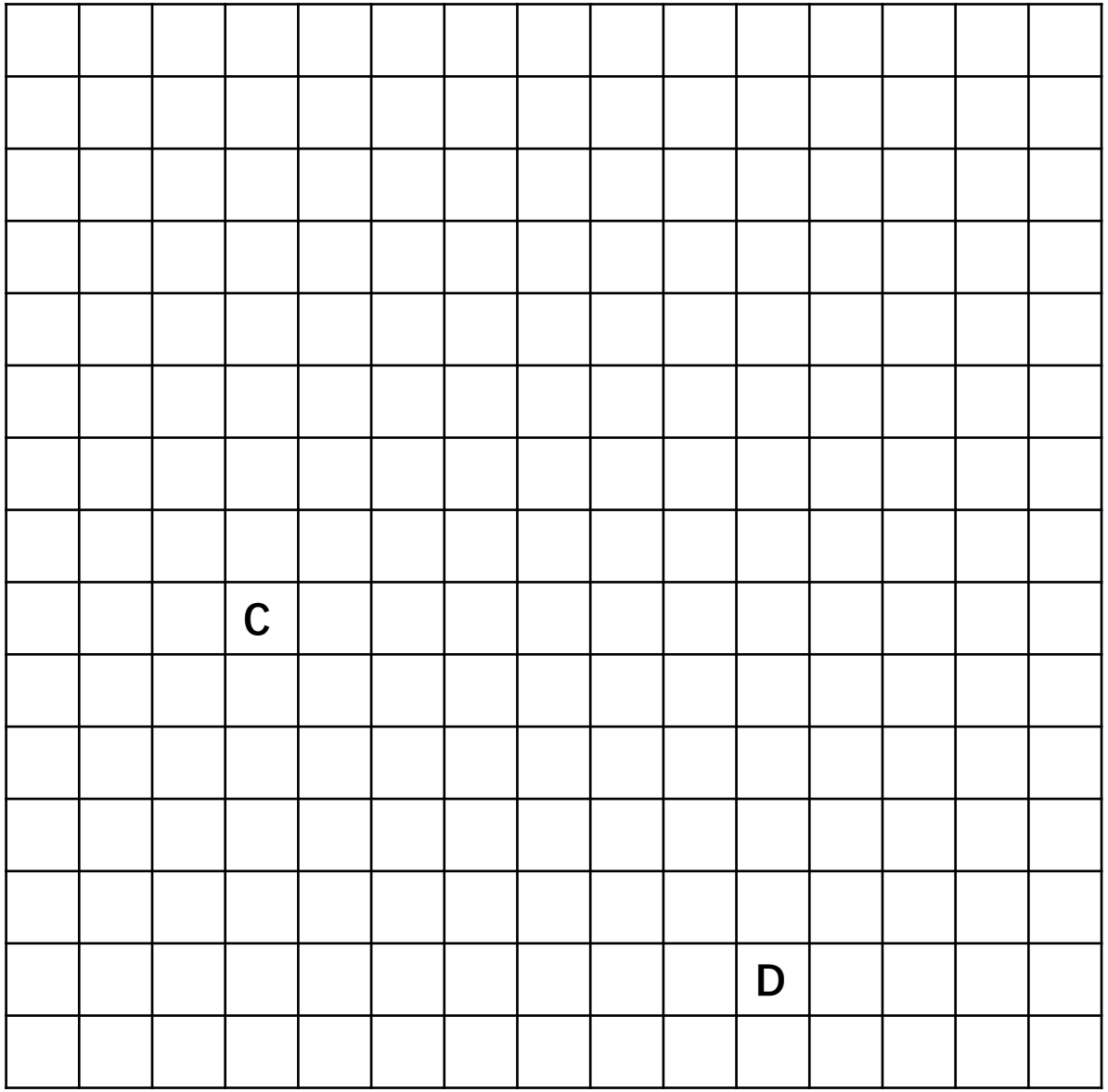
疑问：以上规则如何更好地在编程/规则中表述？



案例1

从起点A到终点B的路径搜寻结果

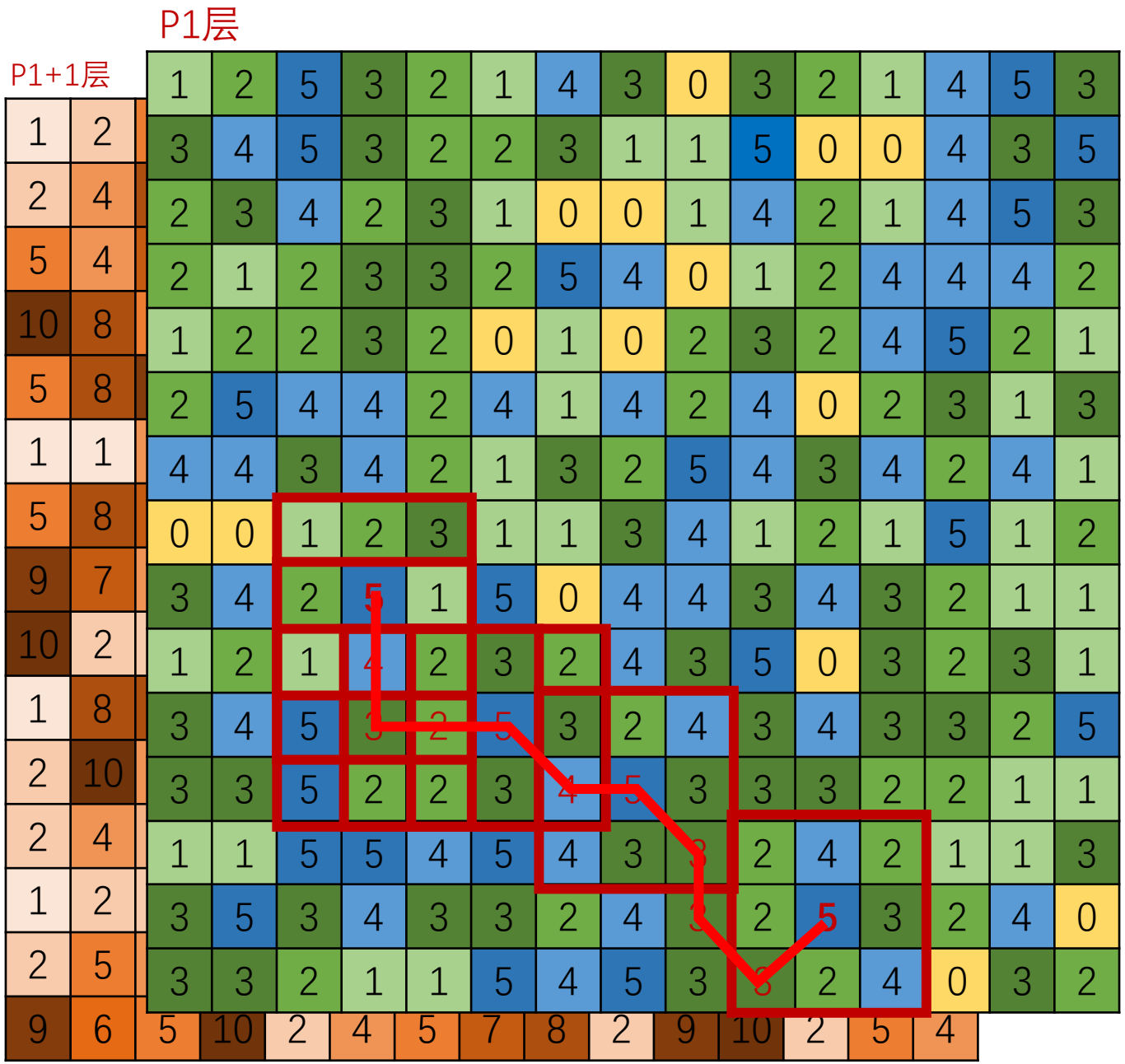
（左下→右上）



案例2

从起点C到终点D的路径搜寻

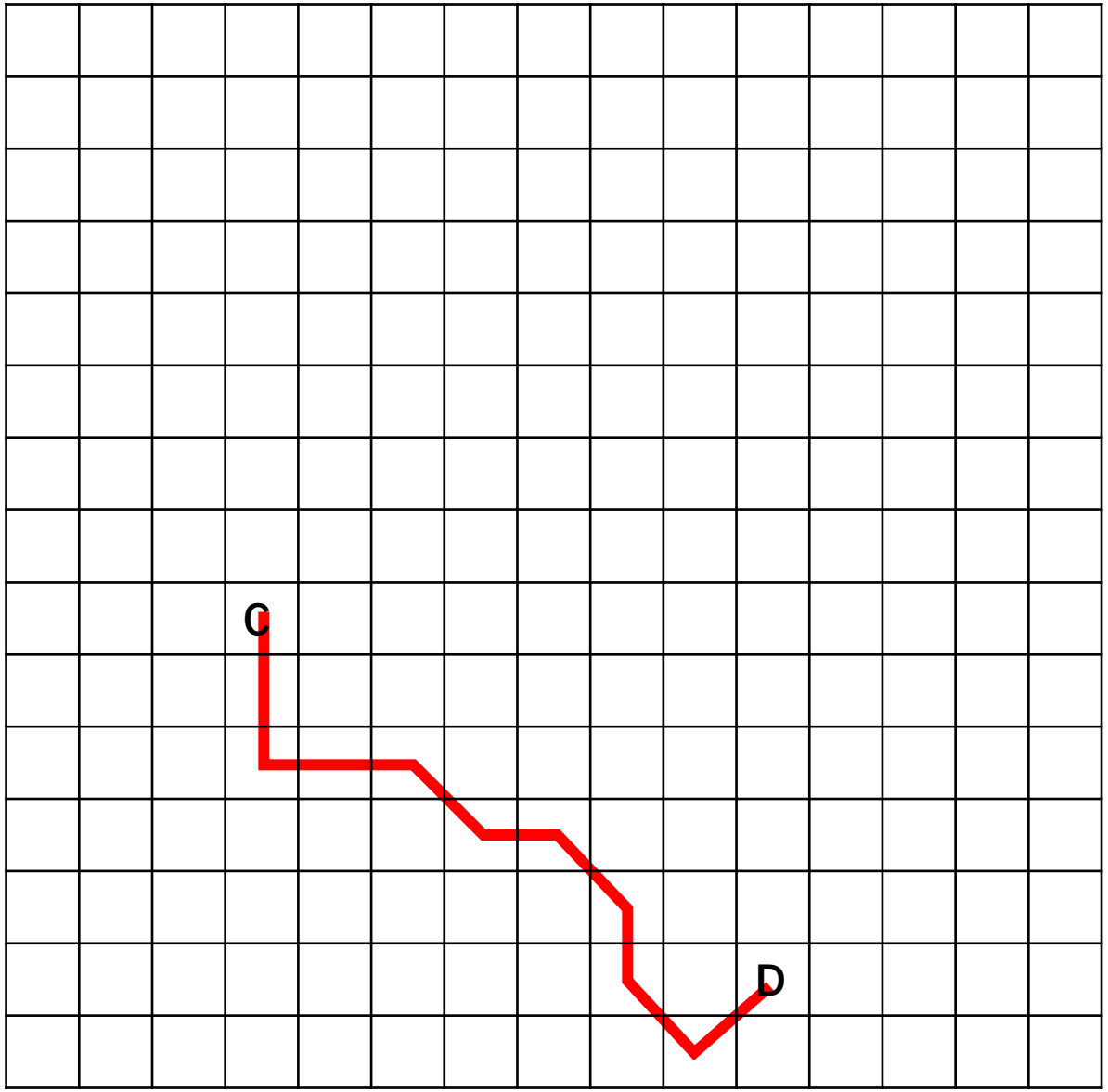
（左上→右下）



案例2

从起点C到终点D的路径搜寻结果

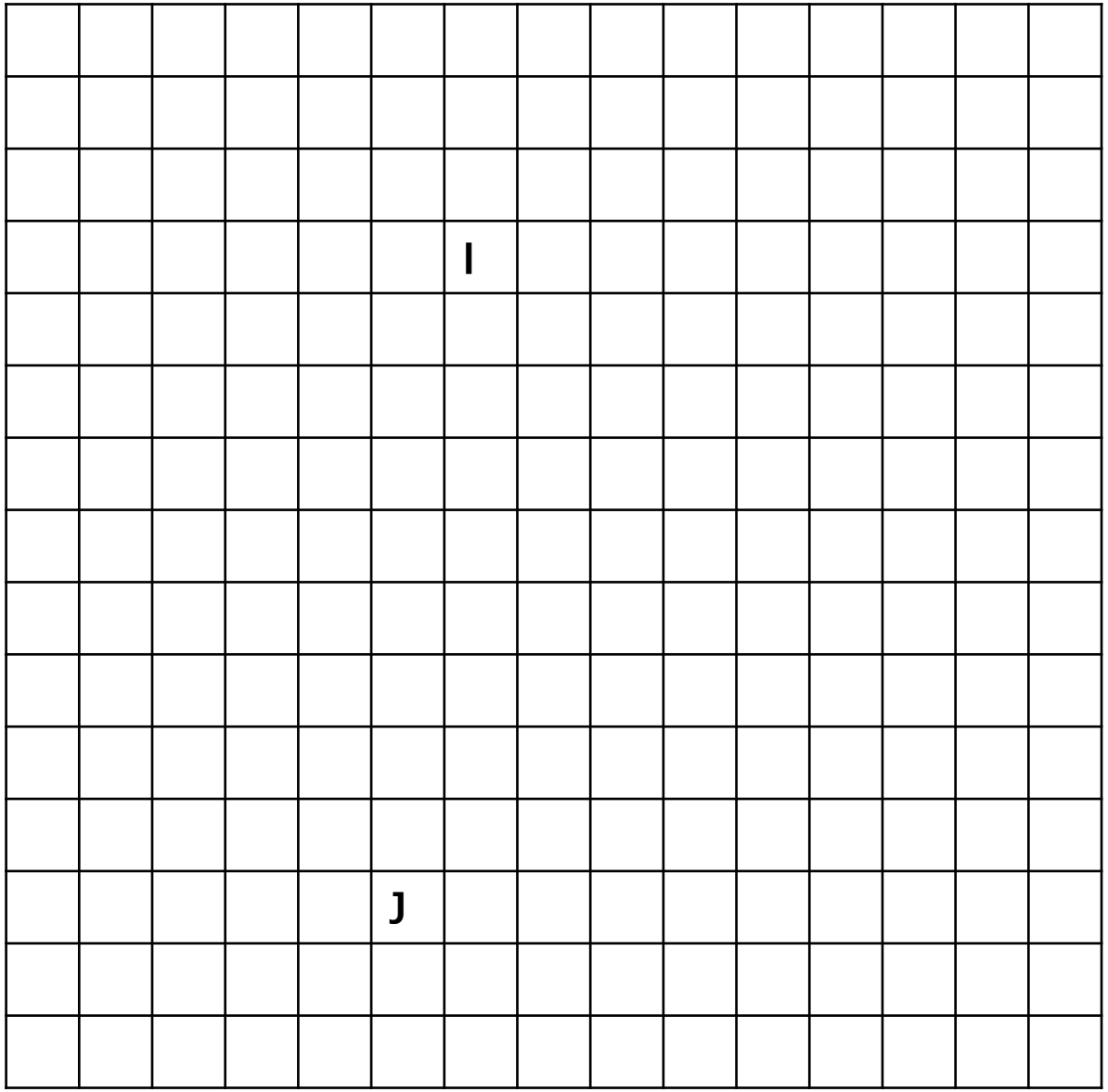
（左上→右下）



案例2

从起点C到终点D的路径搜寻结果

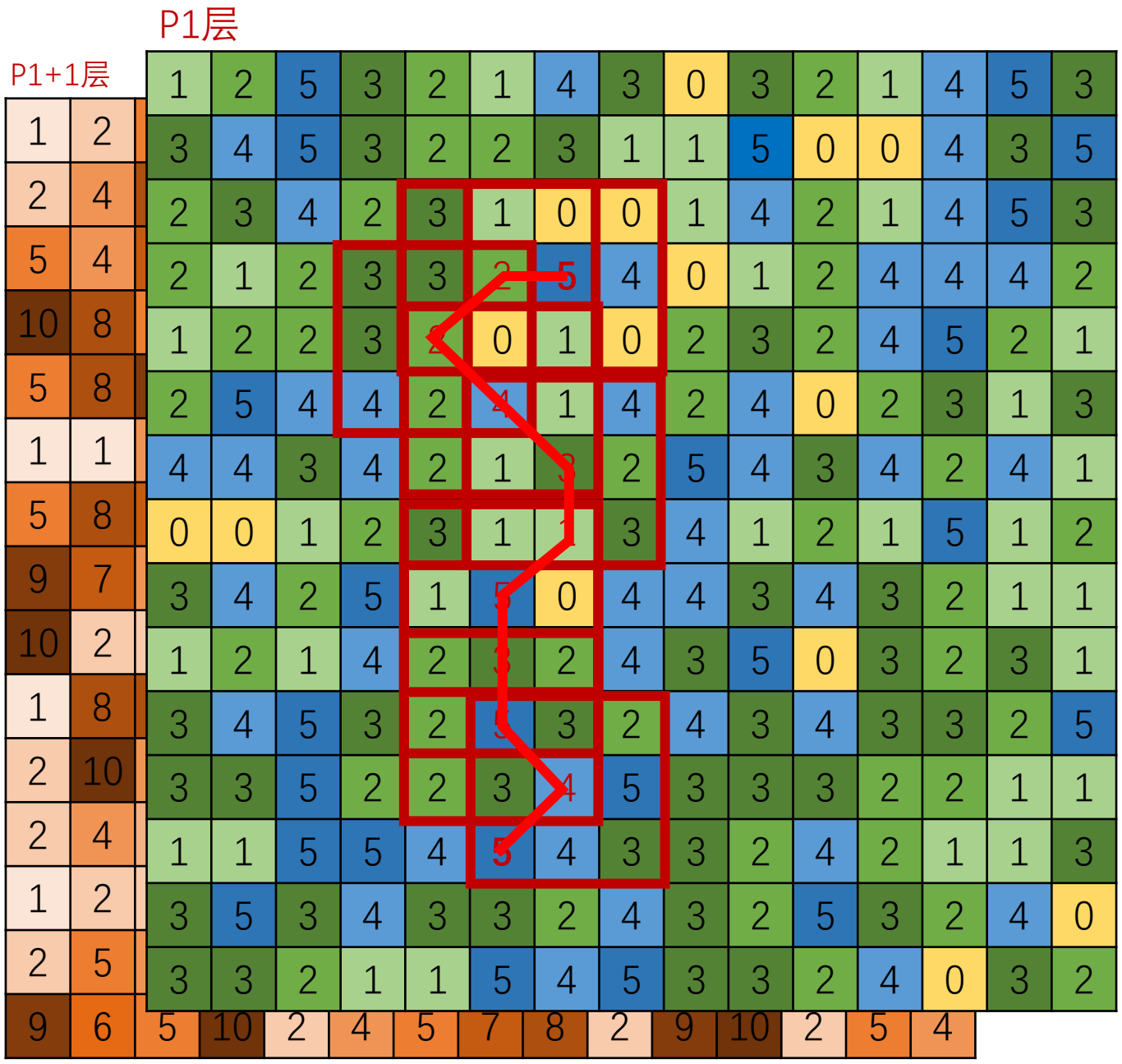
（左上→右下）



案例3

从起点I到终点J的路径搜寻

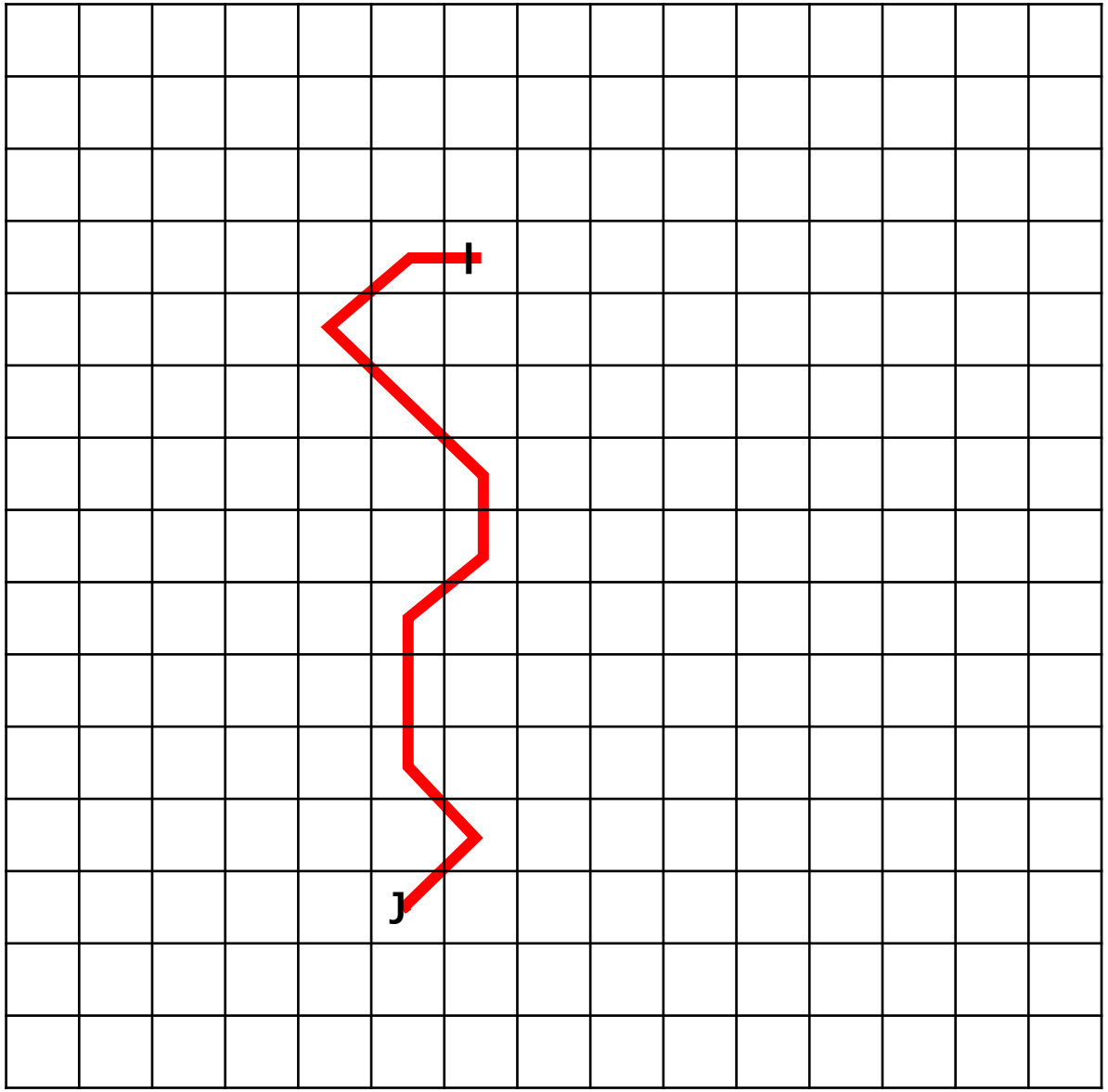
（右上→左下，几乎在一列）



案例3

从起点I到终点J的路径搜寻结果

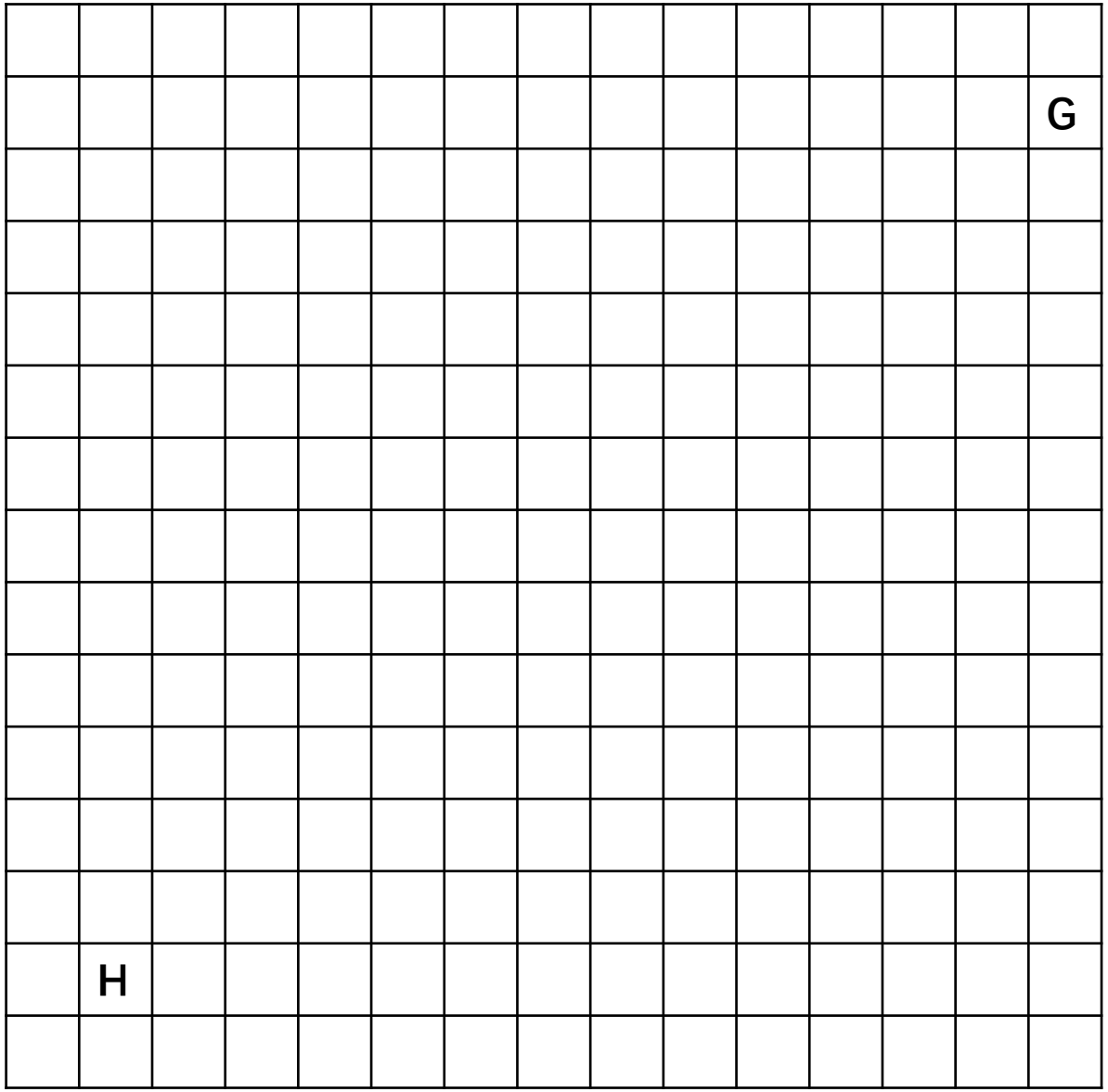
（右上→左下，几乎在一列）



案例3

从起点I到终点J的路径搜寻结果

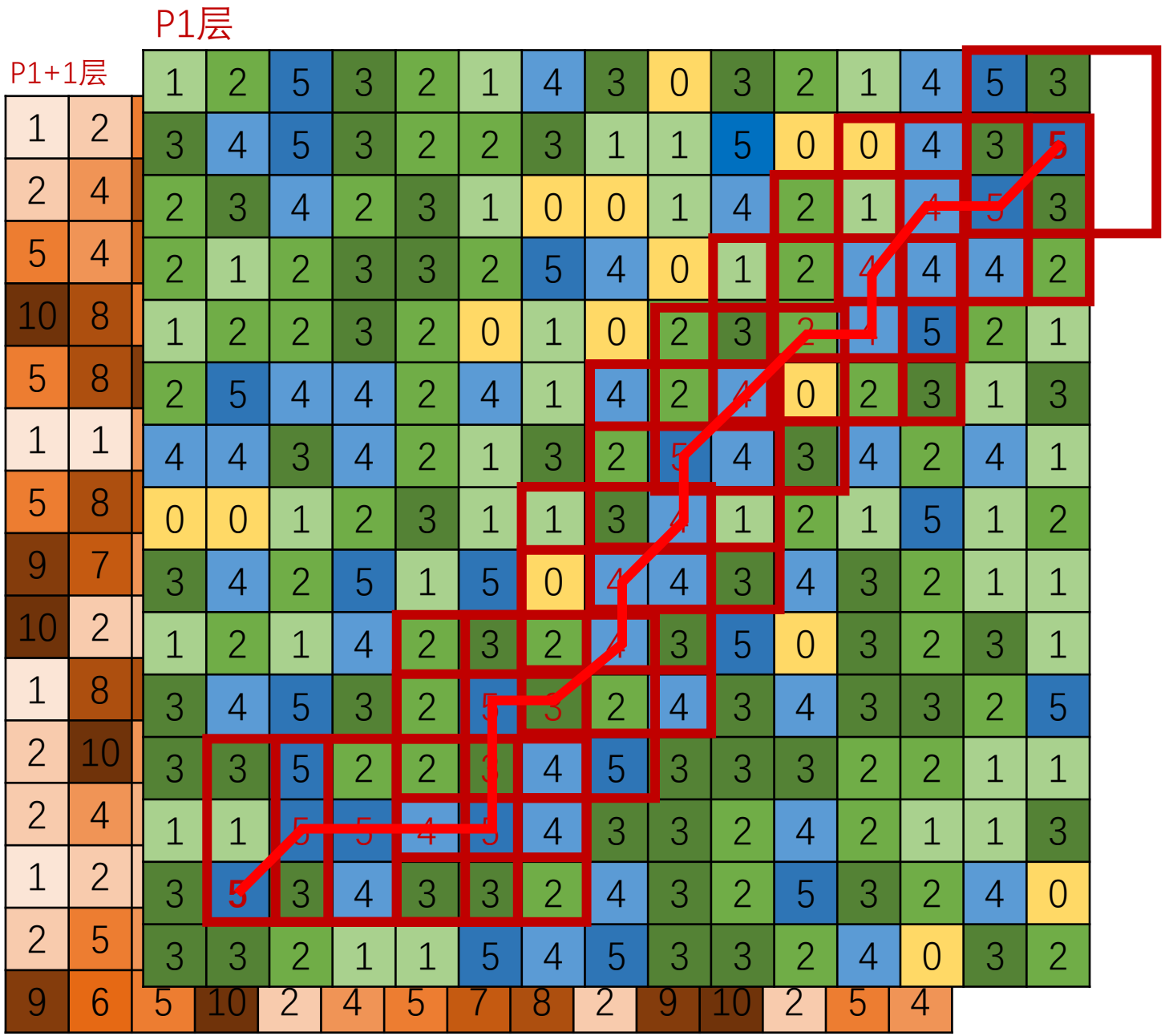
（右上→左下，几乎在一列）



案例4

从起点G到终点H的路径搜寻

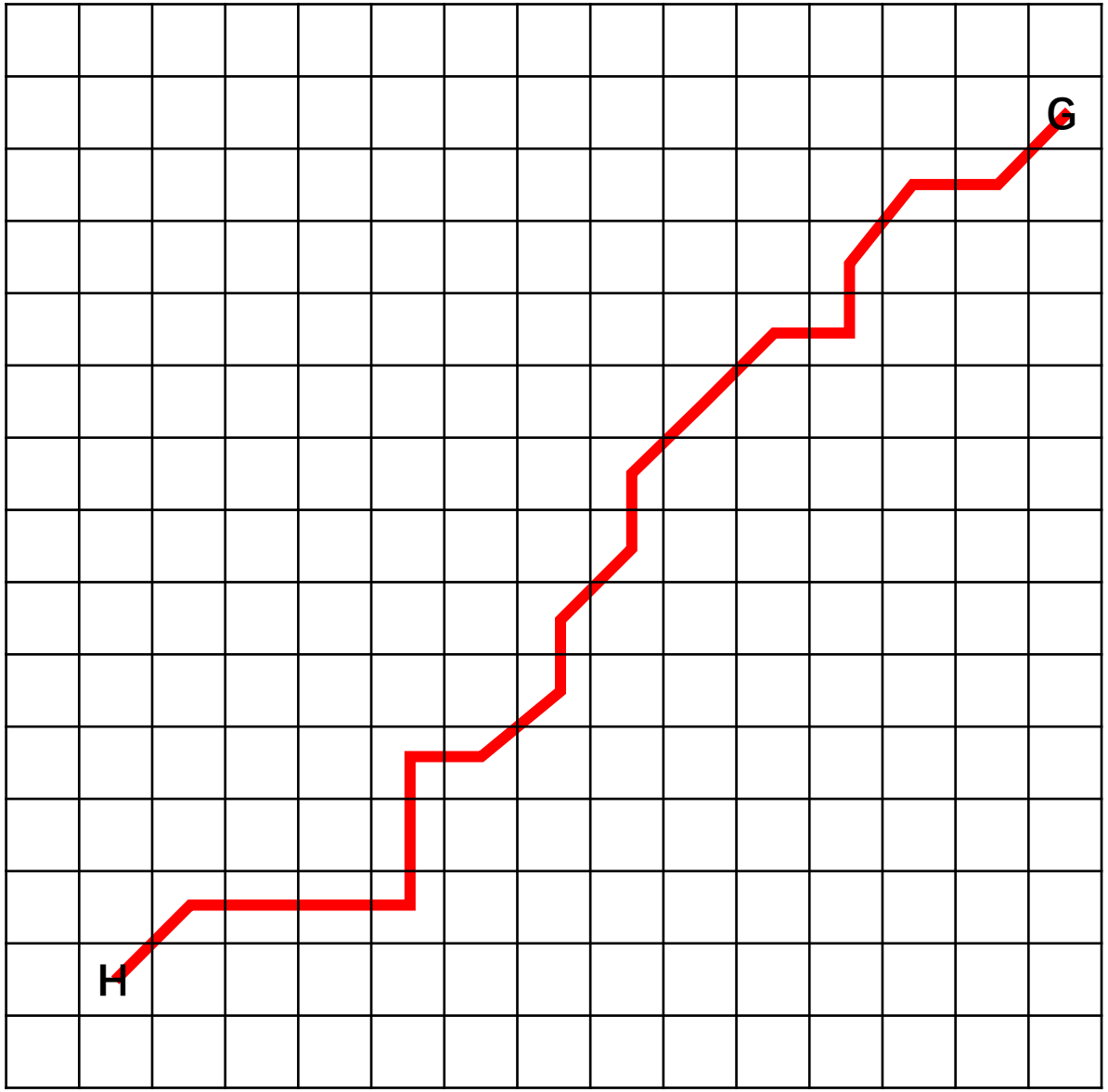
（右上→左下，长距离）



案例4

从起点G到终点H的路径搜寻结果

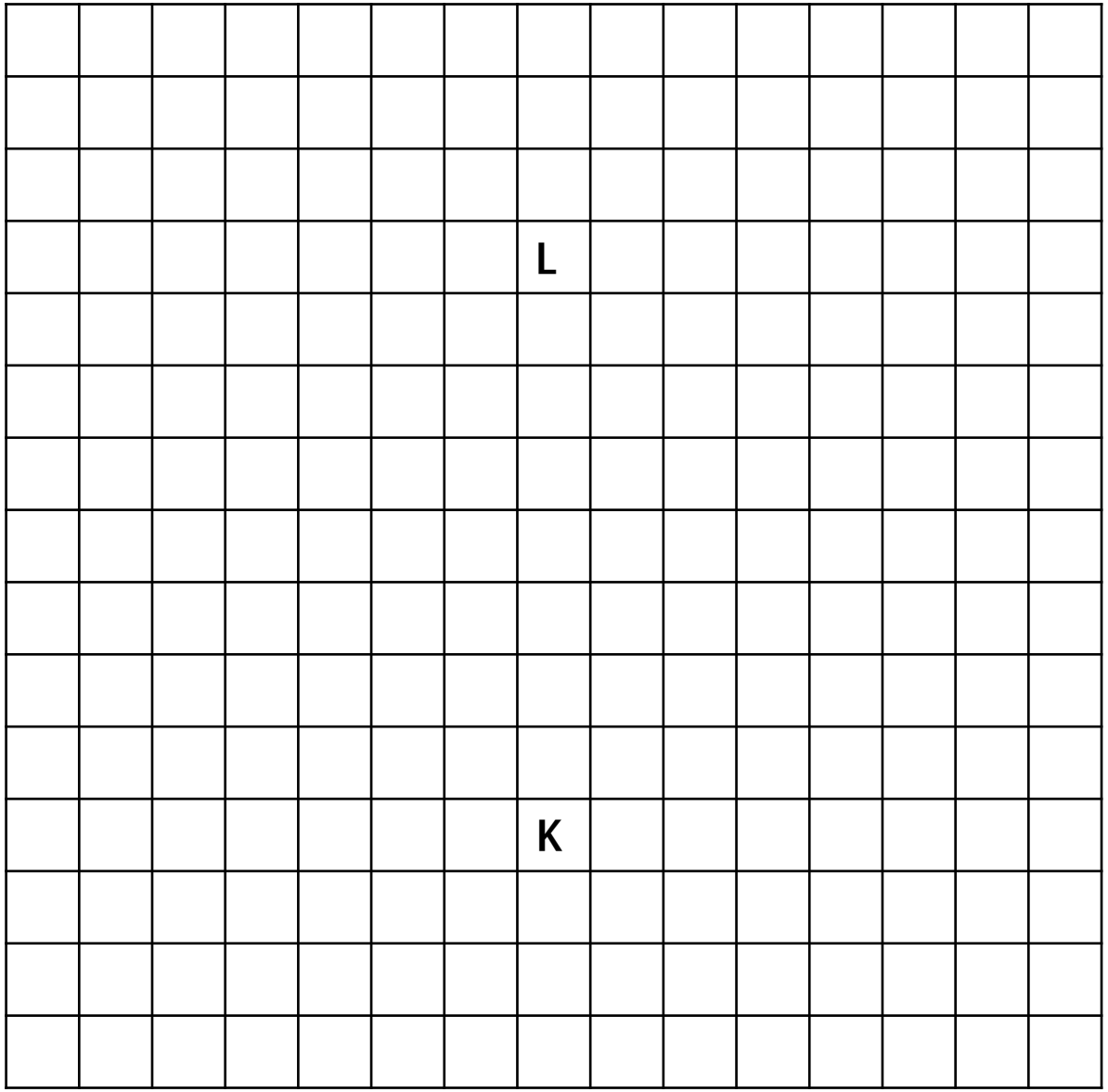
（右上→左下，长距离）



案例4

从起点G到终点H的路径搜寻结果

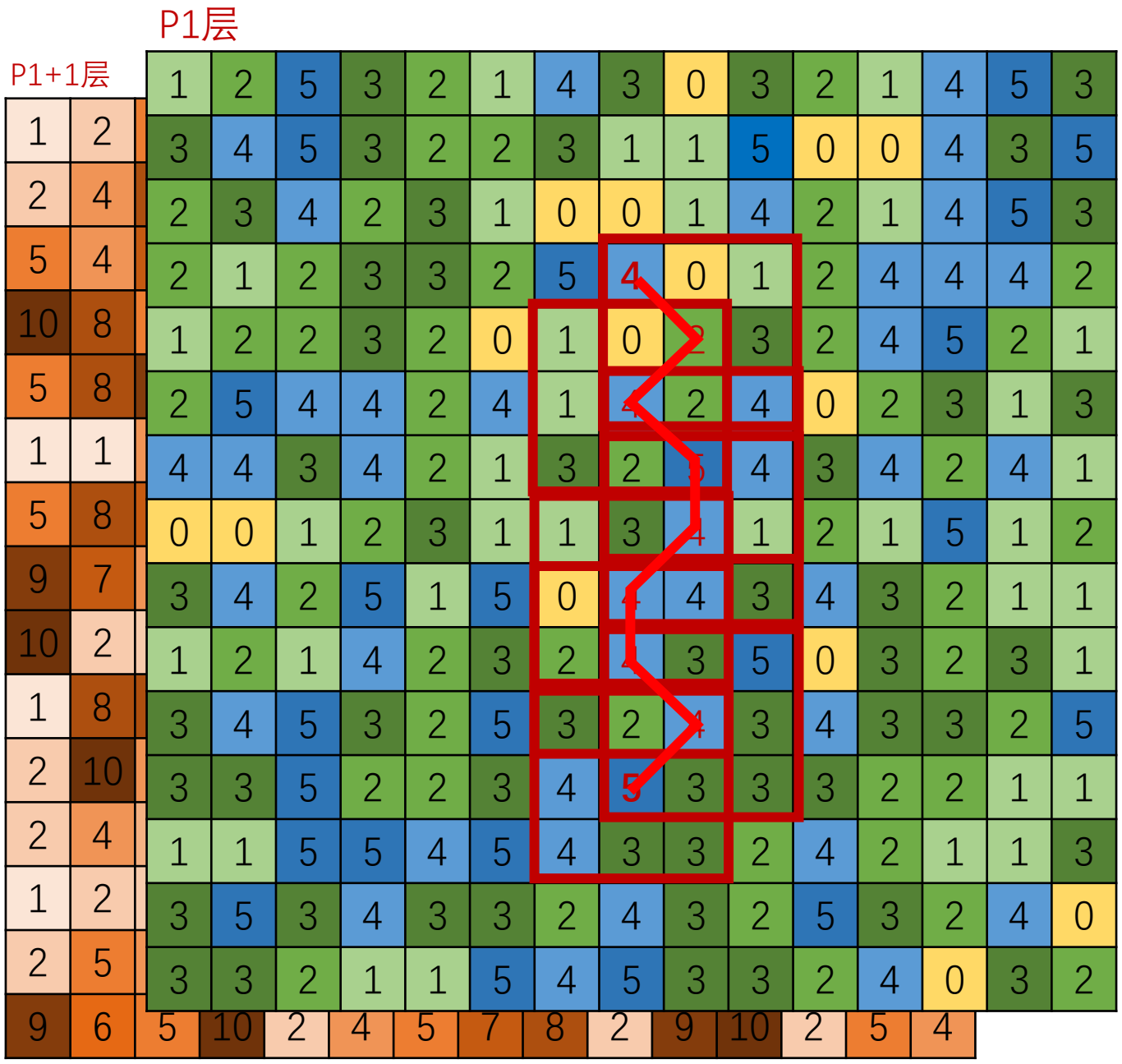
（右上→左下，长距离）



案例5

从起点K到终点L的路径搜寻

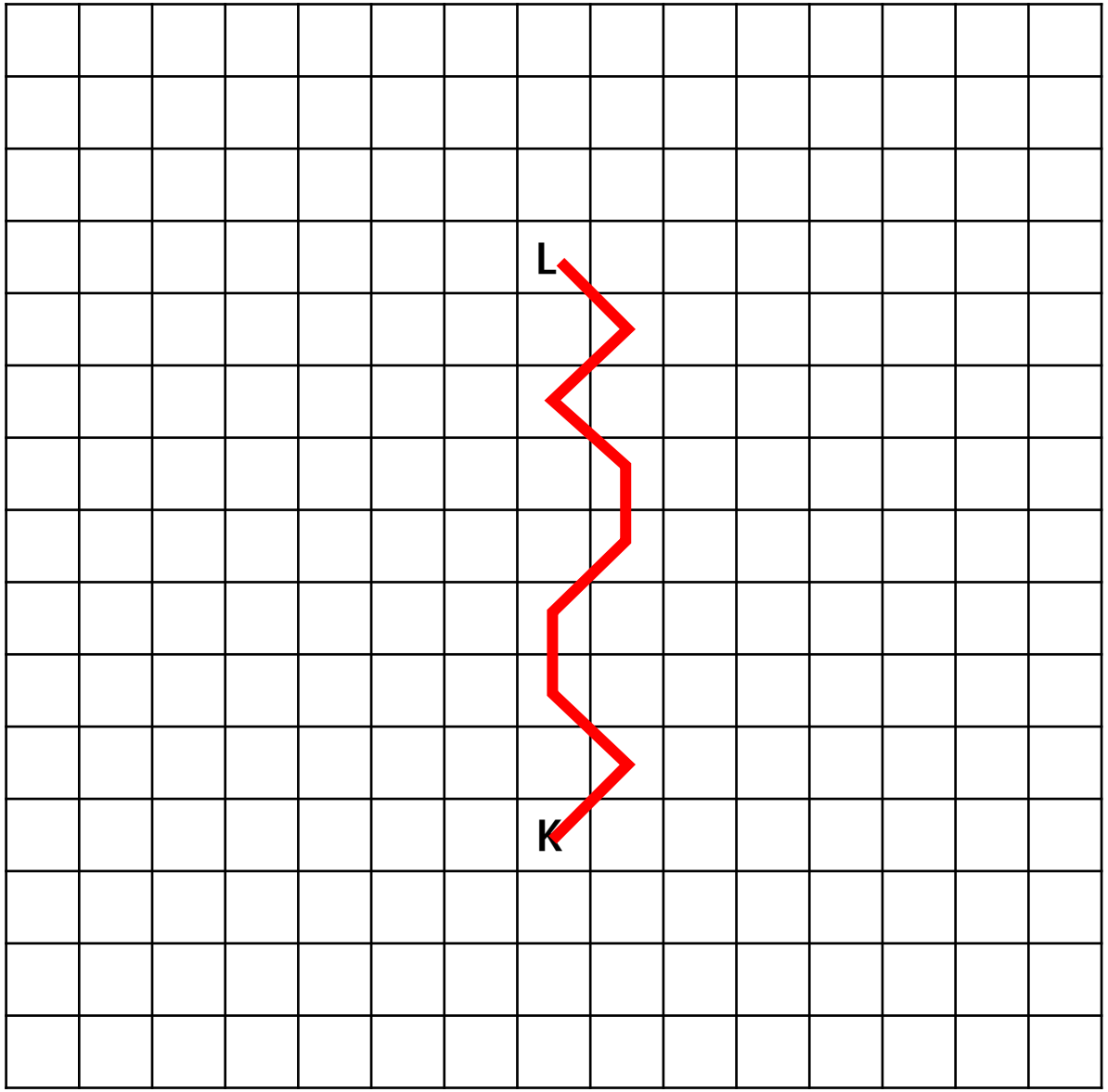
（下→上，在一列）



案例5

从起点K到终点L的路径搜寻结果

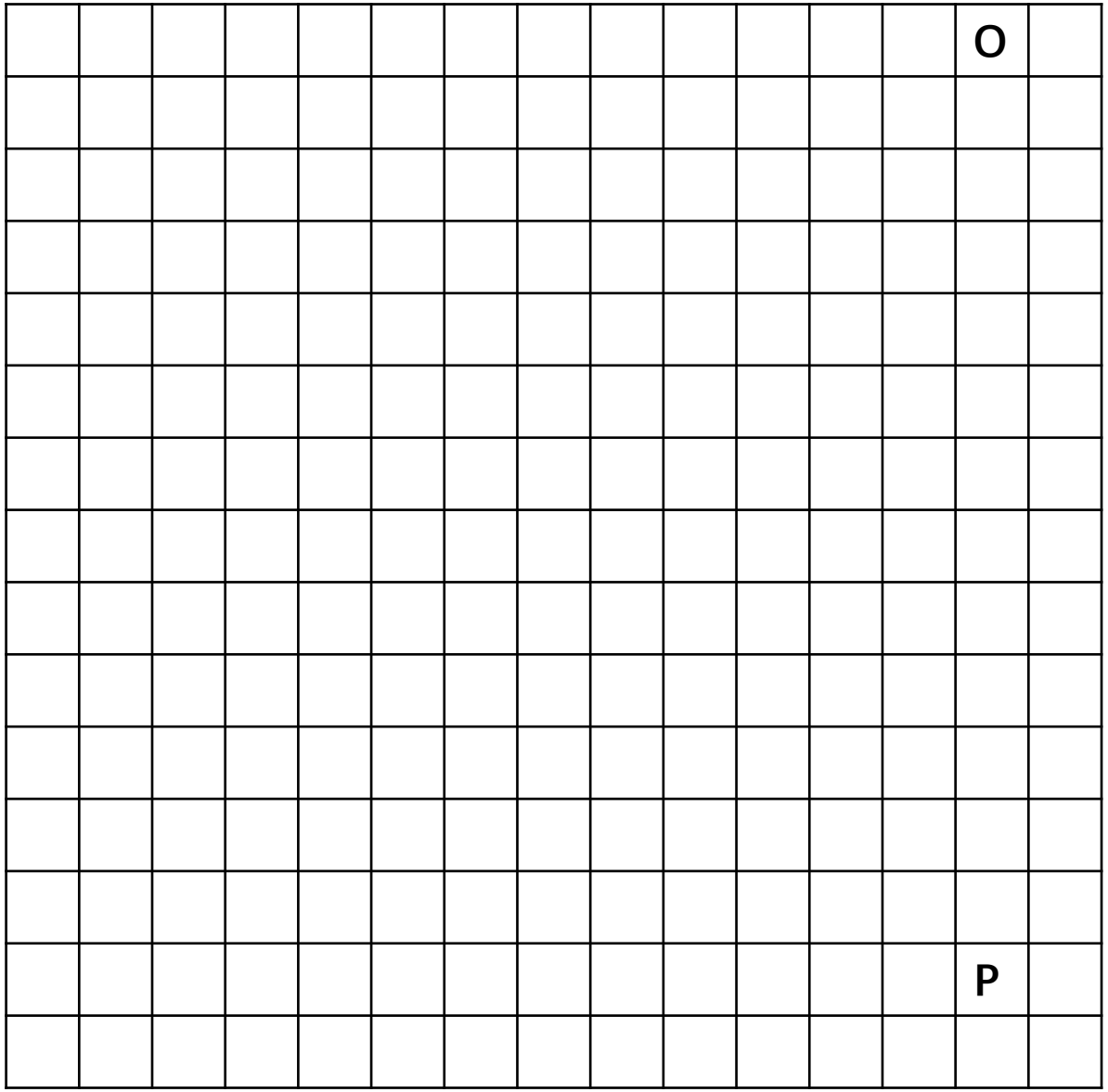
（下→上，在一列）



案例5

从起点K到终点L的路径搜寻结果

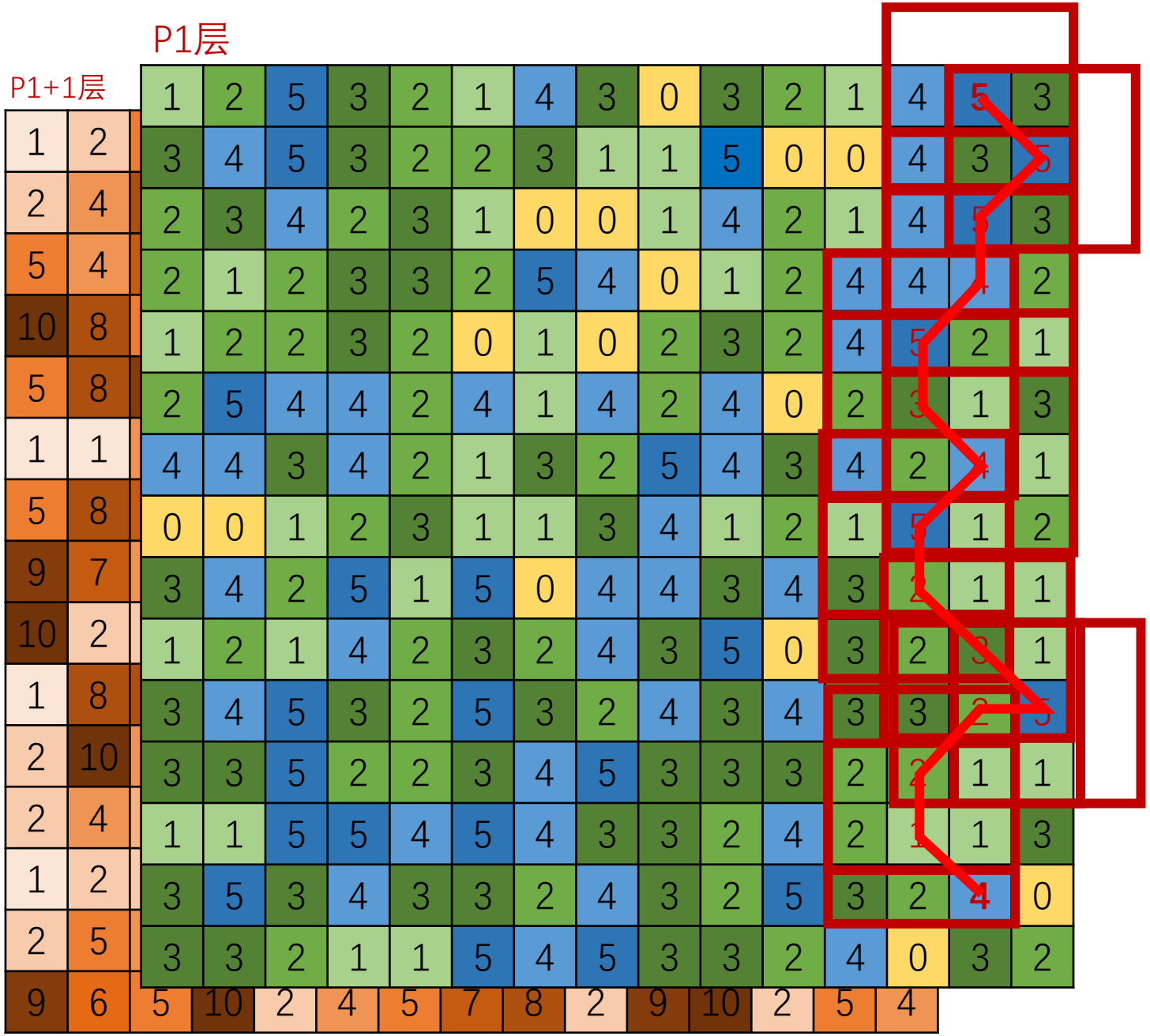
（下→上，在一列）



案例6

从起点O到终点P的路径搜寻

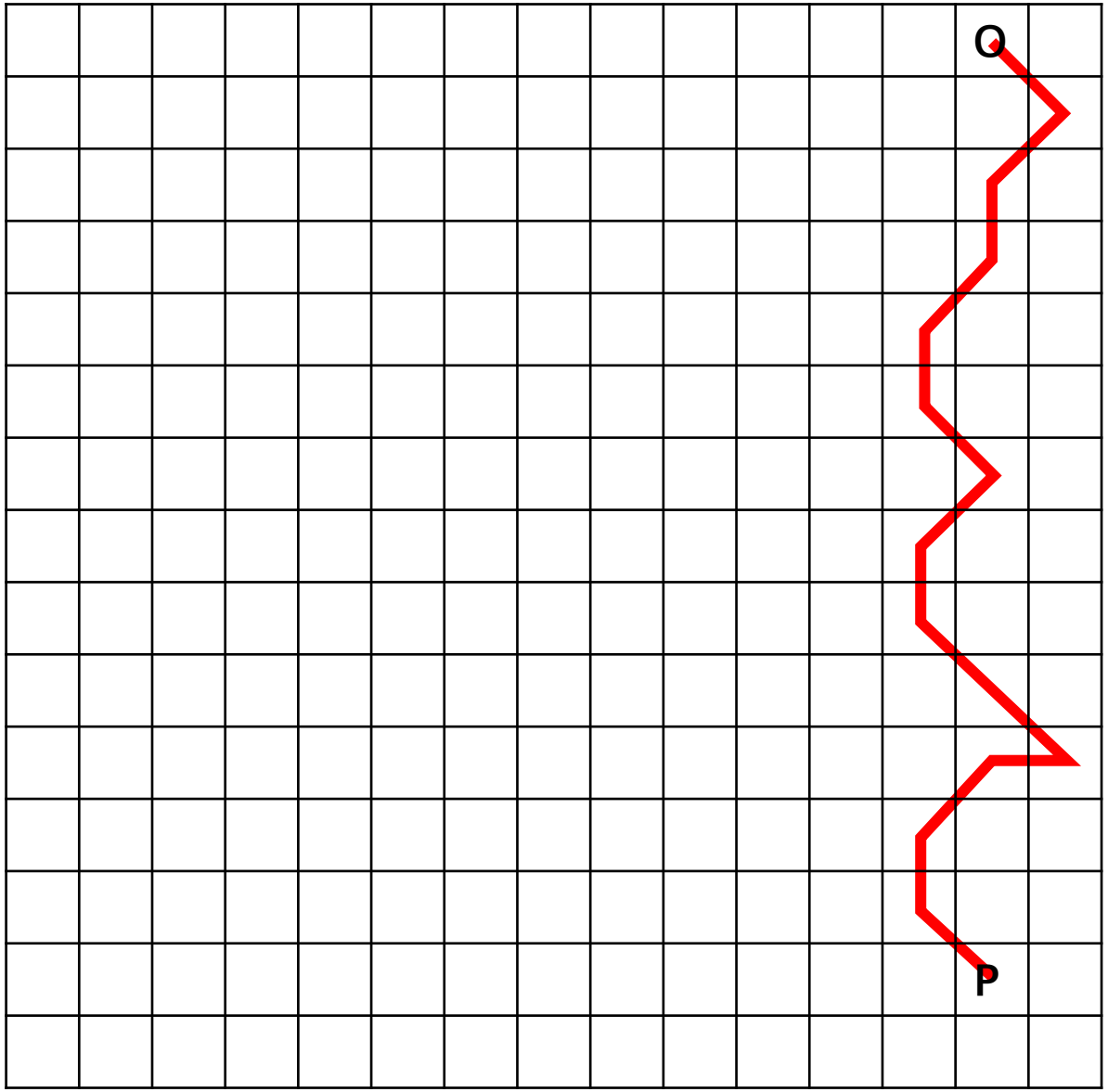
（上→下，长距离）



案例6

从起点O到终点P的路径搜寻结果

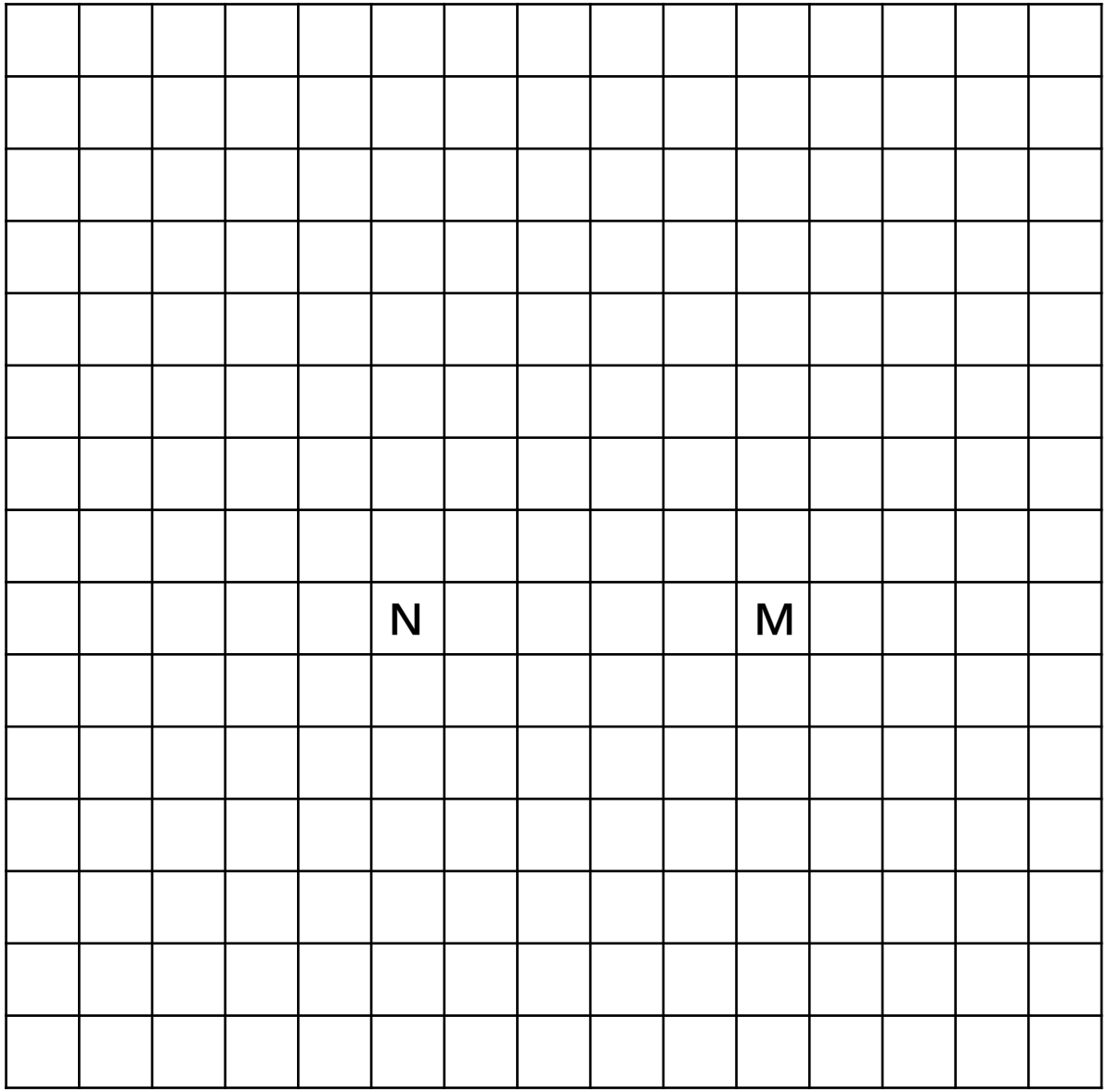
（上→下，长距离）



案例6

从起点O到终点P的路径搜寻结果

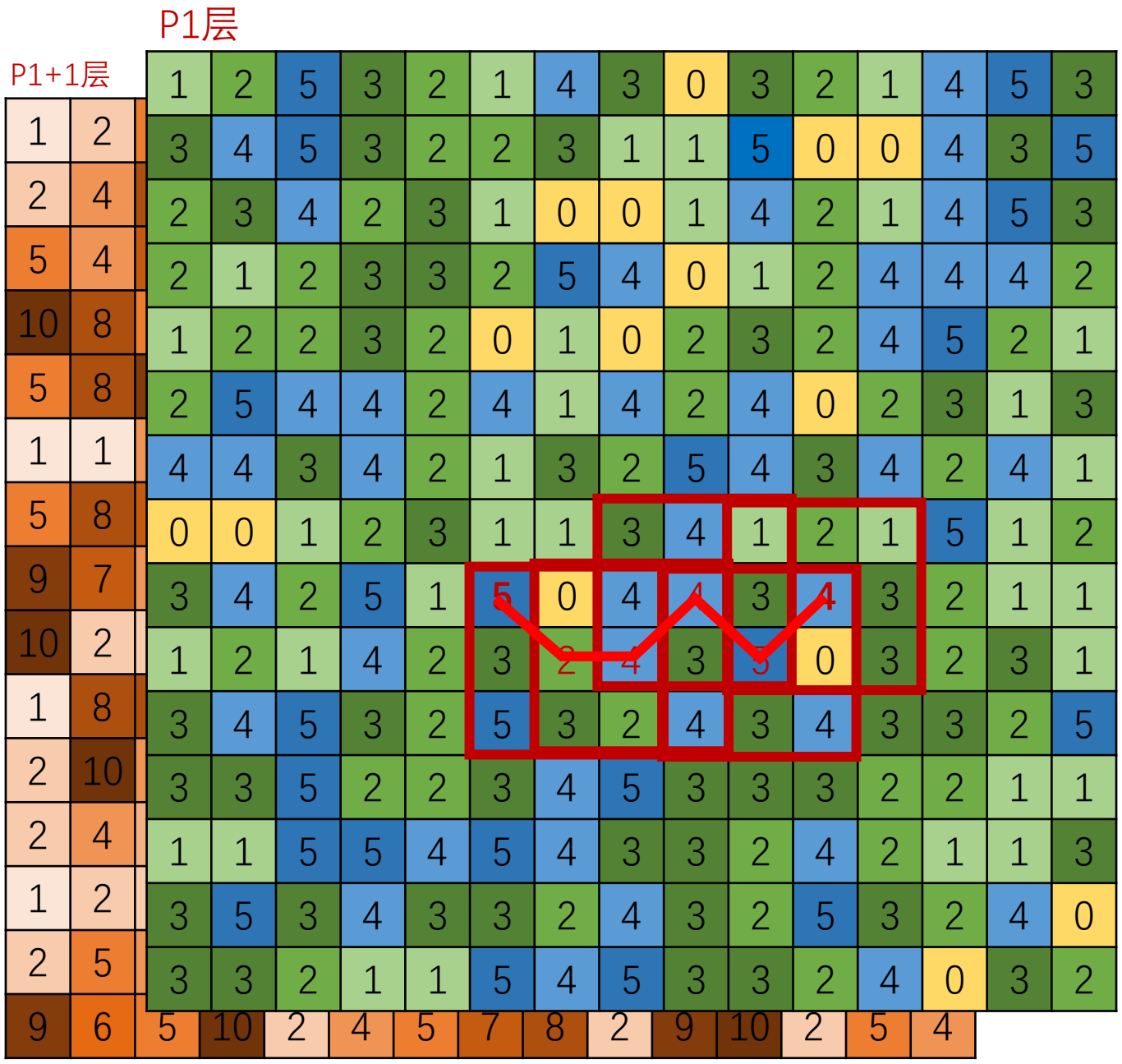
（上→下，长距离）



案例7

从起点M到终点N的路径搜寻

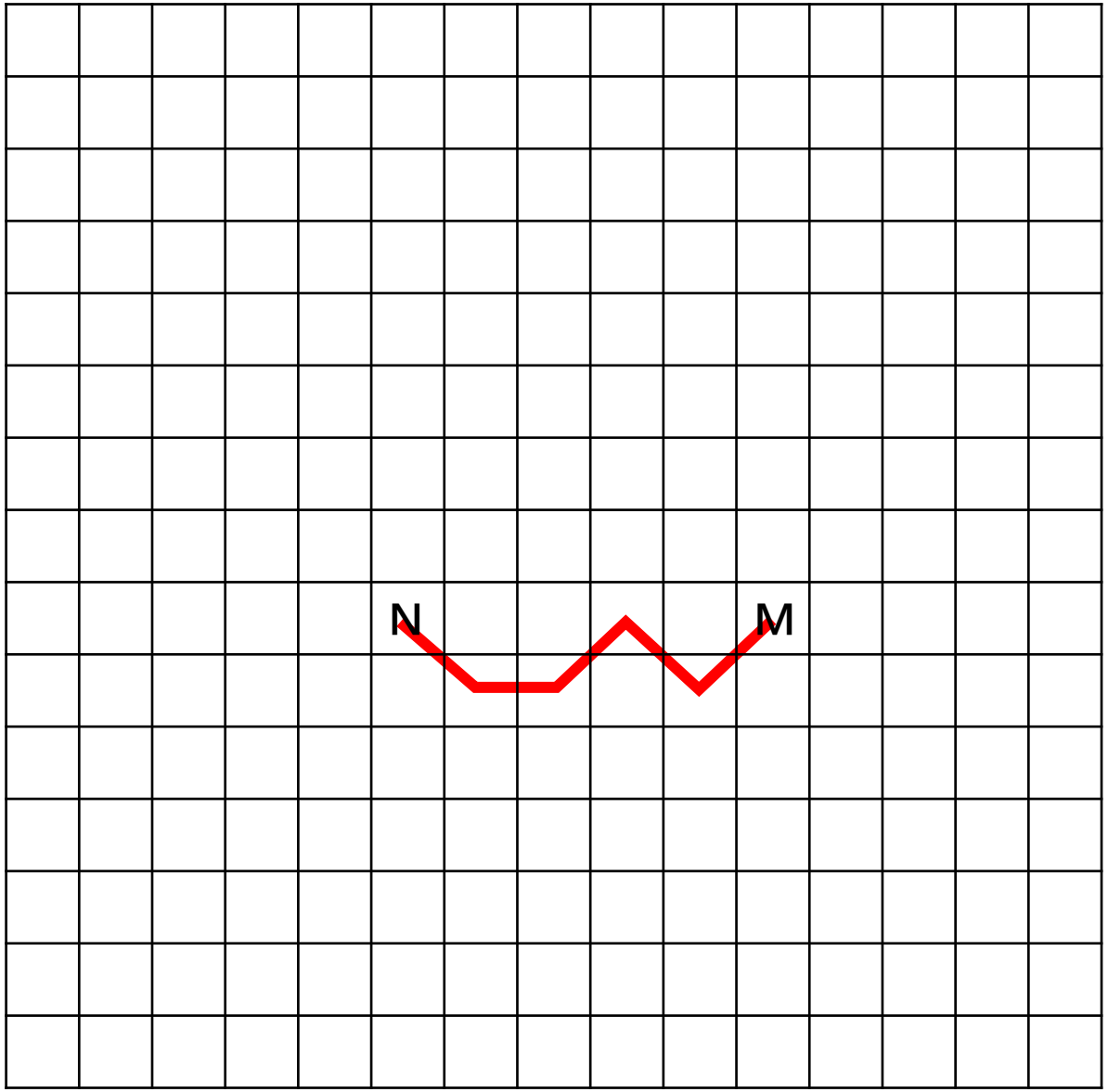
（右→左，近距离）



案例7

从起点M到终点N的路径搜寻结果

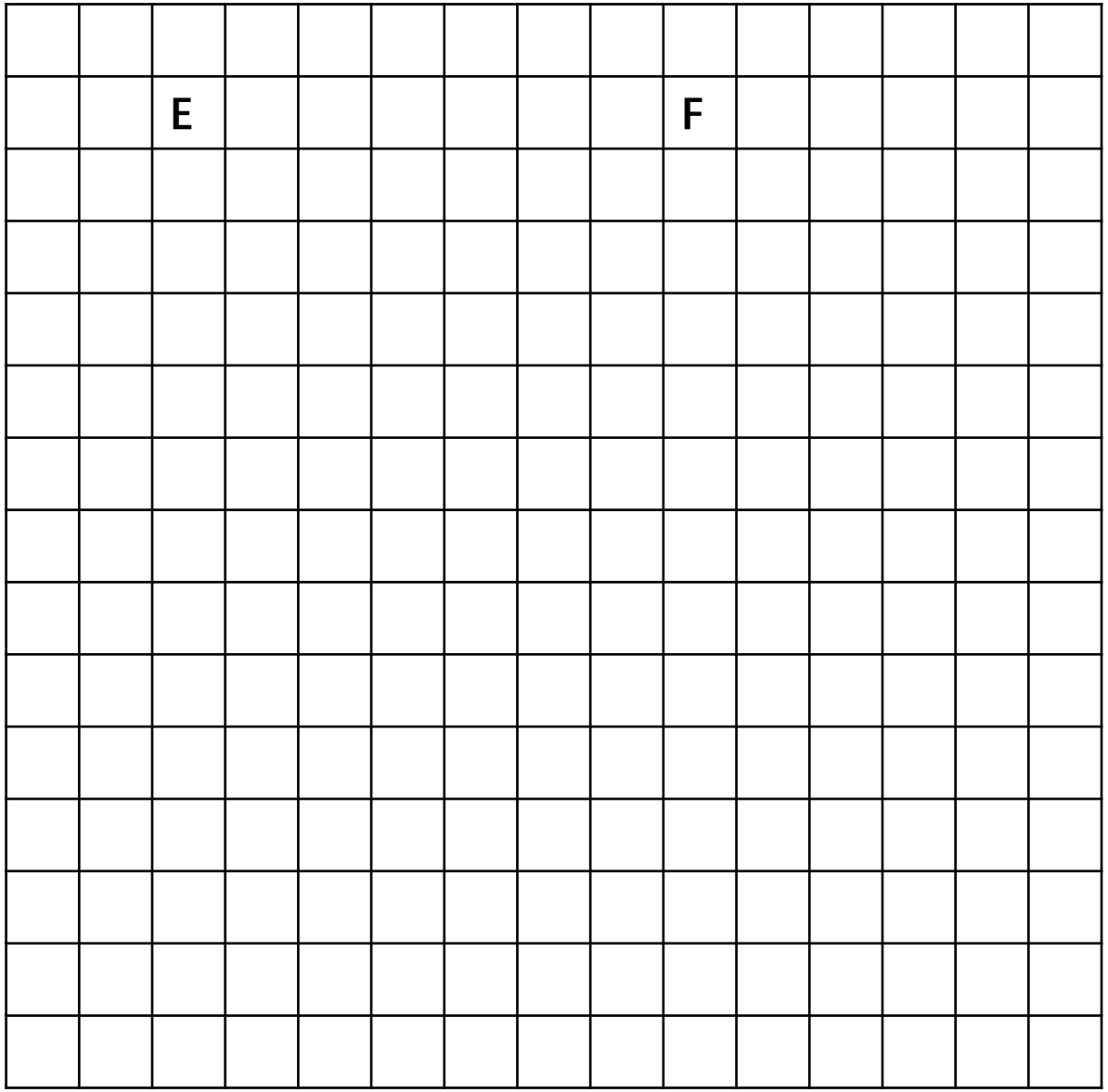
（右→左，近距离）



案例7

从起点M到终点N的路径搜寻结果

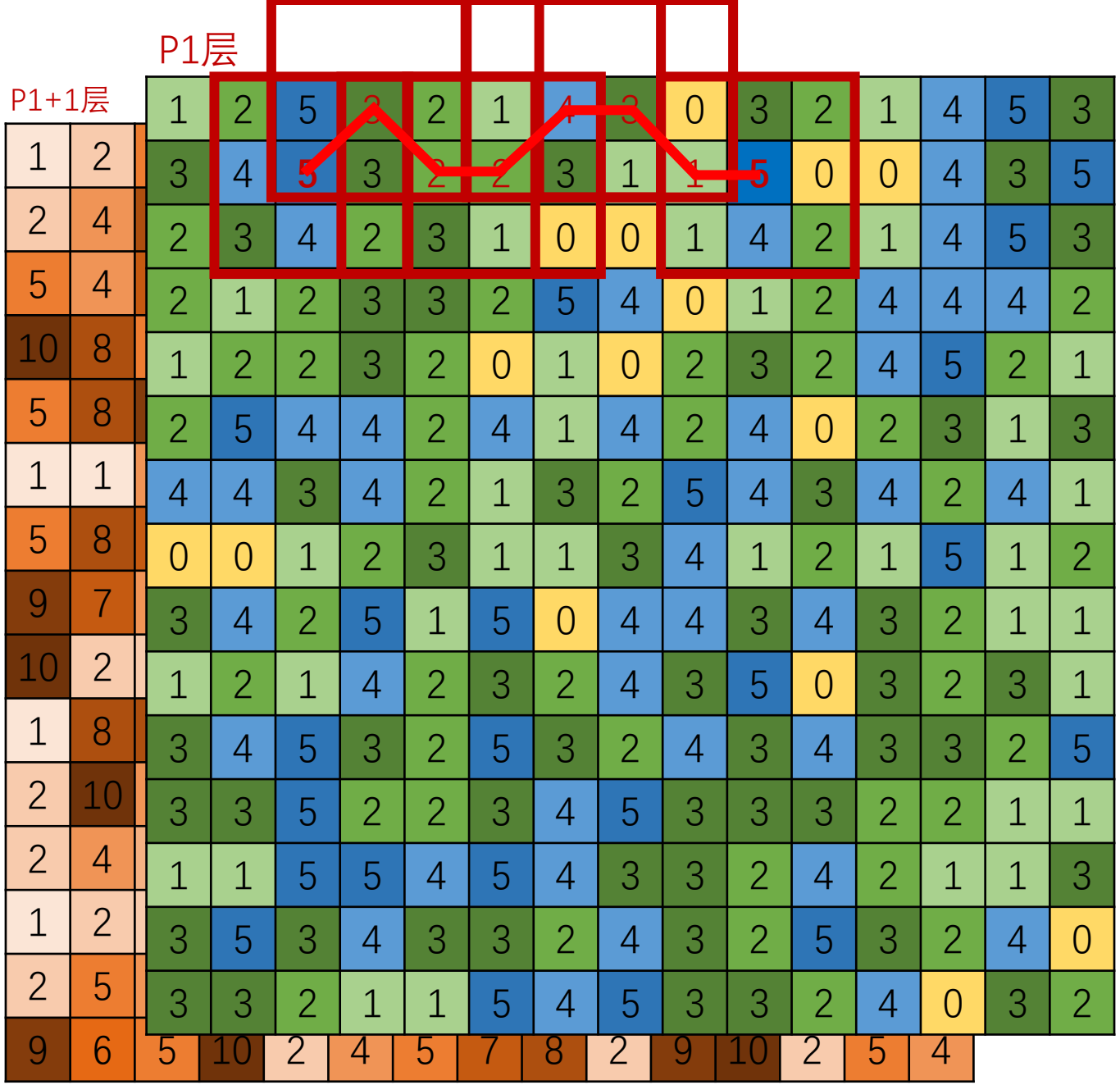
（右→左，近距离）



案例8

从起点E到终点F的路径搜寻

（左→右，在一行）



案例8

从起点E到终点F的路径搜寻结果

（左→右，在一行）

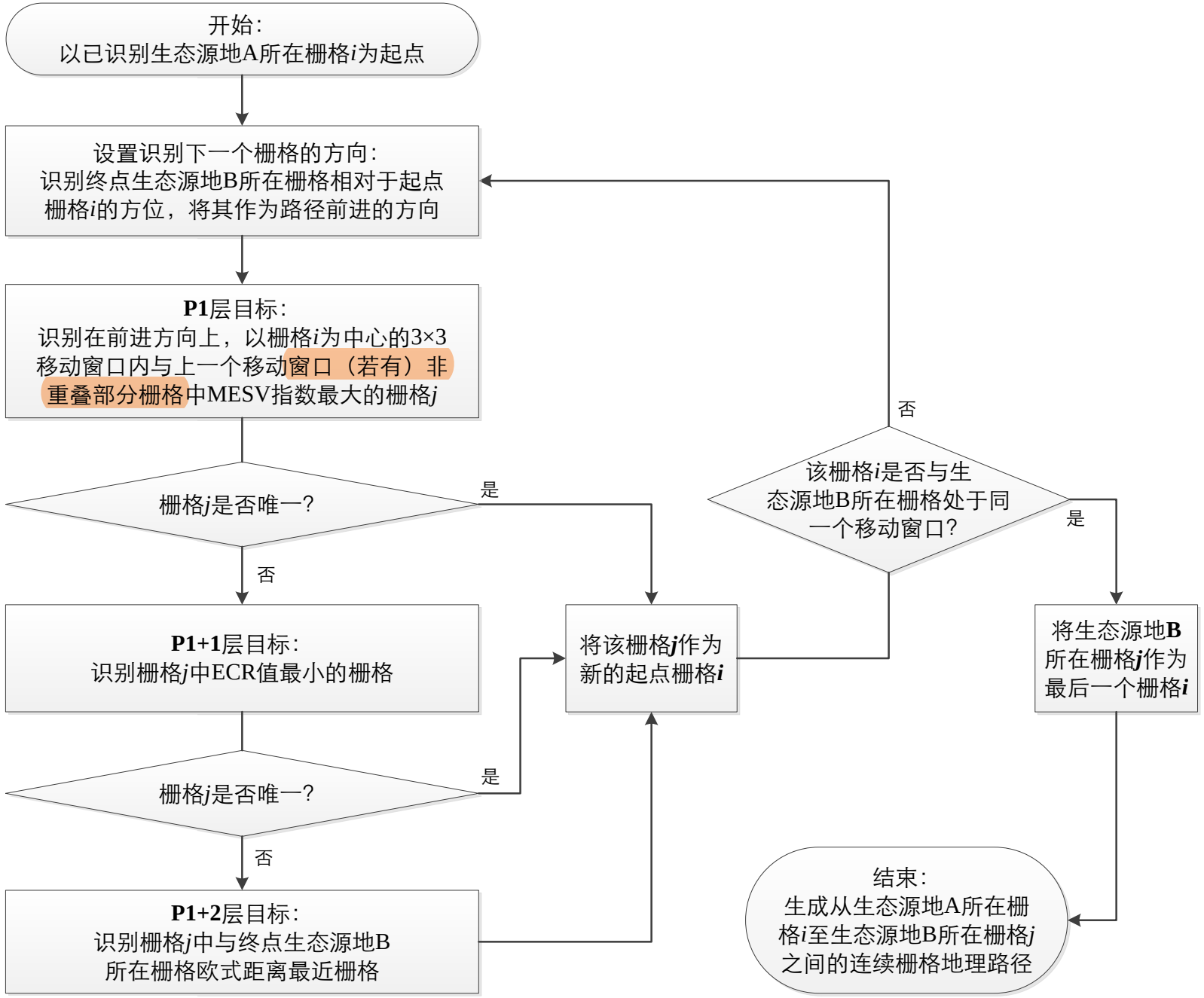
当终点栅格位于起点栅格正右方时，栅格识别前进的方向是包含上中下的右侧栅格；

当移动窗口覆盖超出栅格边界时，仅对于范围内有赋值的栅格执行贪婪选择的规则。

案例8

从起点E到终点F的路径搜寻结果

（左→右，在一行）


开始：
以已识别生态源地A所在栅格i为起点
设置识别下一个栅格的方向：
识别终点生态源地B所在栅格相对于起点
栅格i的方位，将其作为路径前进的方向
P1层目标：
识别在前进方向上，以栅格i为中心的3×3
移动窗口内与上一个移动窗口（若有）非
重叠部分栅格中MESV指数最大的栅格j
栅格j是否唯一？
是
否
P1+1层目标：
识别栅格j中ECR值最小的栅格
栅格j是否唯一？
是
否
P1+2层目标：
识别栅格j中与终点生态源地B
所在栅格欧式距离最近栅格
将该栅格j作为
新的起点栅格i
该栅格i是否与生
态源地B所在栅格处于同
一个移动窗口？
否
是
将生态源地B
所在栅格j作为
最后一个栅格i
结束：
生成从生态源地A所在栅
格i至生态源地B所在栅格j
之间的连续栅格地理路径