

1. (a) 状态: 无人机当前所在网格的坐标, 如 (3, g) [行 1-8, 列 a-h]

初始状态: 起点 S 的坐标 (3, b)

目标状态: 终点 E 的坐标 (8, h)

行动: 向相邻方向 (上, 下, 左, 右) 移动一格, 且目标网格不为障碍 (3, f) (5, c) (6, b)

代价函数: 行动的开销 单个行动: 单次行动代价 (1) 行动序列: 多个行动代价之和

(b) 状态: $(s_1, s_2, s_3), s_i \in \{0, 1\}$, 代表 3 个机关当前所处的状态, 激活为 "1", 未激活为 "0"

初始状态: (0, 1, 0)

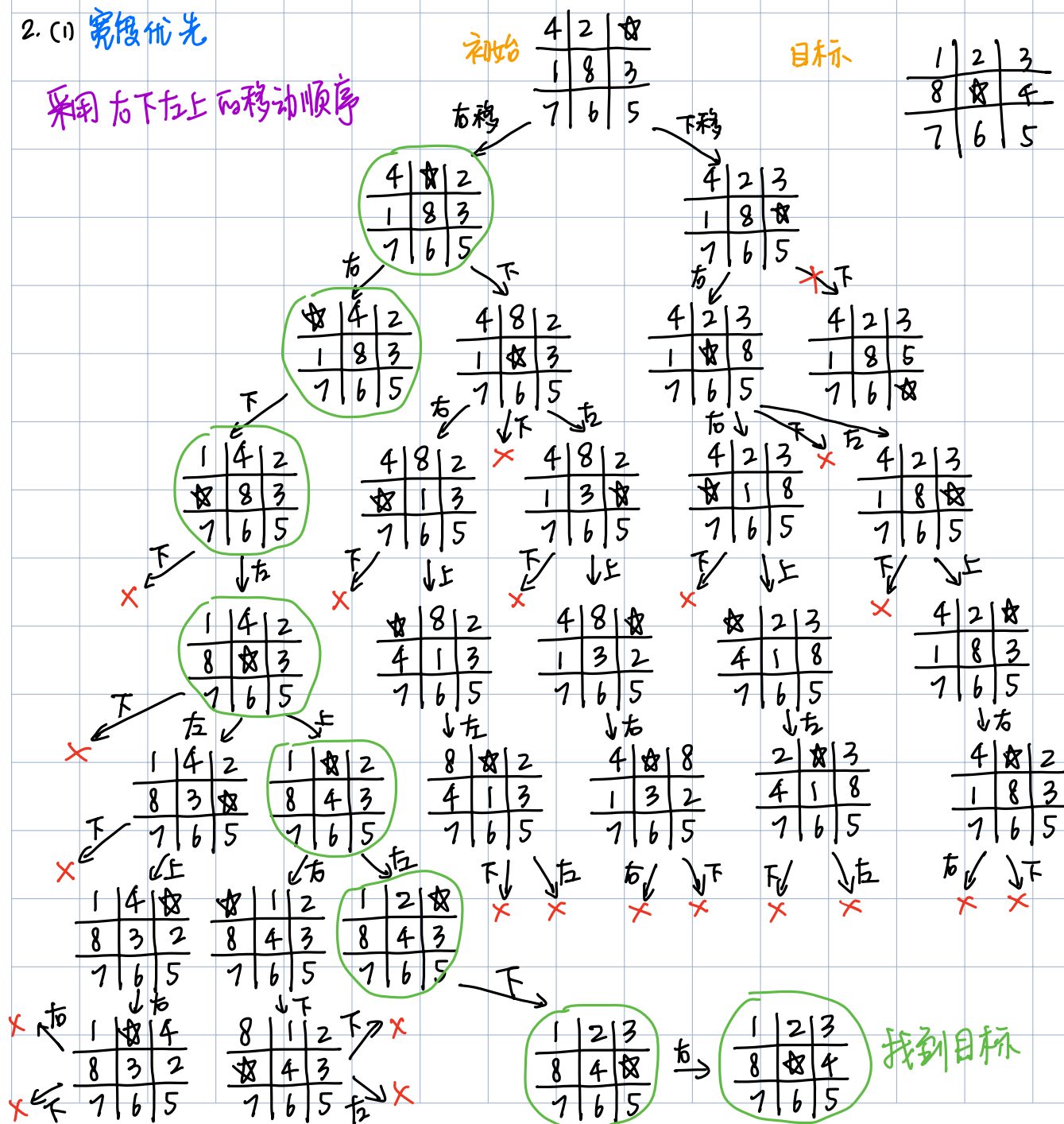
目标状态: (1, 1, 1) 或 (0, 0, 0)

行动: 切换第 i 个机关的状态, $i=1, 2, 3$

代价函数: 行动的开销 单个行动: 单次行动代价 (1) 行动序列: 多个行动代价之和

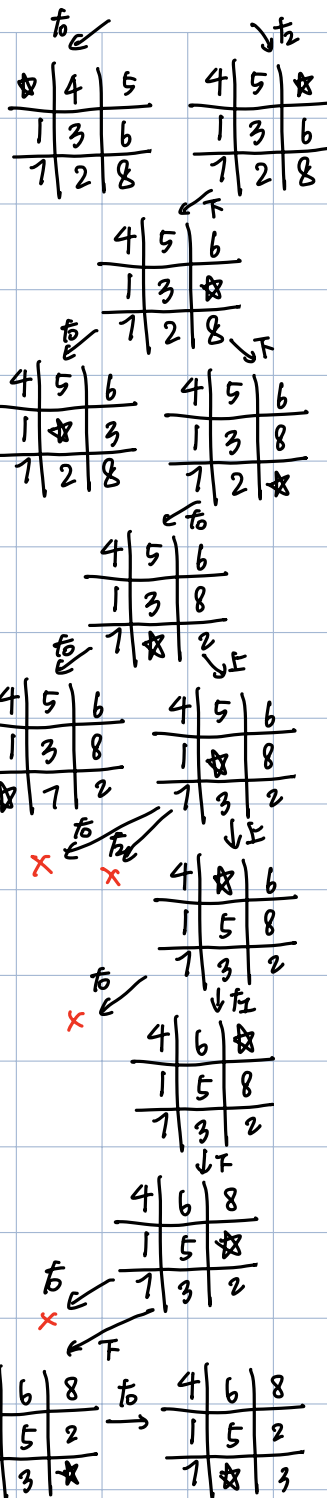
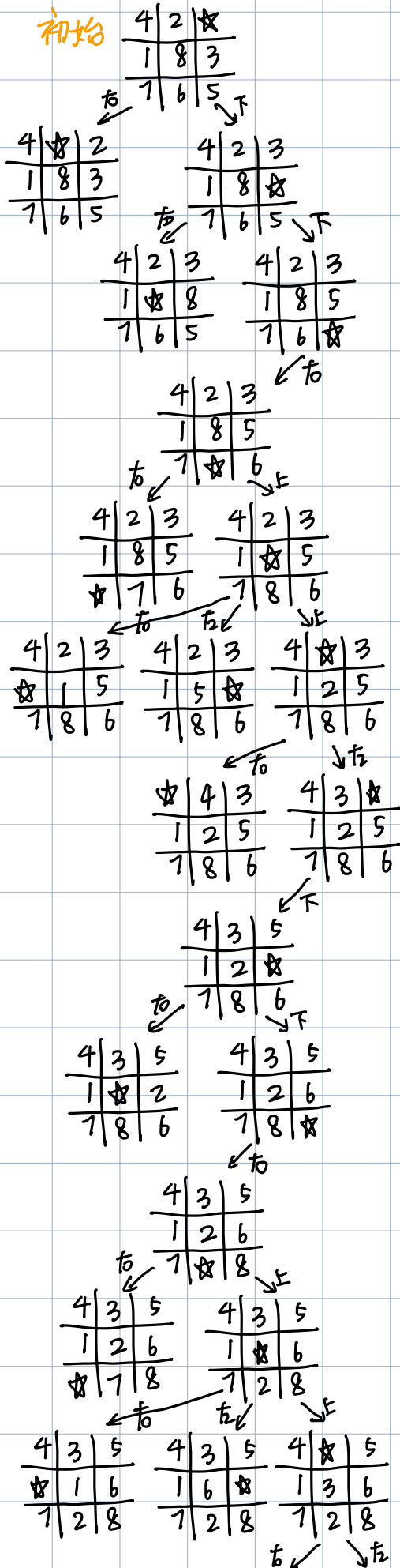
2. (1) 宽度优先

采用右下左上的移动顺序

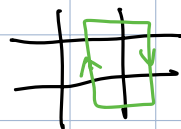


深度搜索：采用右下左上的移动顺序

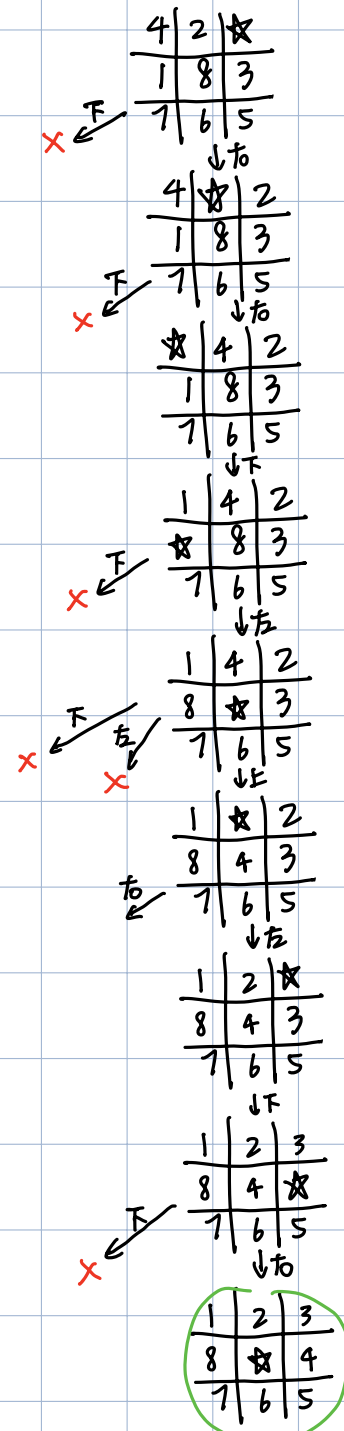
初始



可见“能量核心” $*$ 只会在如下
路径中循环,不会改变1和4的
位置,陷入死循环。



但如果采用下右左上的顺序移动
则很快能找到路径



找到目标

(2) BFS与DFS的优势比较:

- BFS: 总能找到最短路径,
但要遍历每层扩展,占内存,复杂度高
- DFS: 复杂度低且内存占用少,
但可能陷入死循环,也不保证最优解

(3) $g(n)$ = 移动次数

$$h(n) = \text{每个量子比特到目标位置的横向与纵向距离之和 (曼哈顿距离)}$$
