人工智能基础 第 1 次作业

1、对以下问题，定义状态、行动、代价，指明初始状态、目标状态，以及如何产生后继状态。

(a) 小明在思考怎么使用红方马攻击黑方将，请帮小明写出红方马攻击黑方将的路径（即不考虑其他棋子，仅考虑红方马的行动）。

(a) 初始状态

(b) 能够对黑方将军的位置

图 1: 第 1 题 (a)

【解】:

状态：马所处的位置（坐标）。

行动：马按照棋子规则，跳到另一个合法的位置。

代价：每次行动消耗 1 次行棋机会，或者 1 步。

初始状态：马当前所处的位置，即图1(a)所示。

目标状态：能够对黑方将产生将军的位置，如图1(b)蓝色圆点所示。注意另一侧的两个位置被兵阻隔无法形成将军。

产生后继状态：不考虑其它棋子。则马从当前位置开始，按照“日”字形，不被“蹩马腿”规则阻碍，不超越棋盘范围，且目标位置不被其它棋子占据的条件下，能够跳到的下一个位置，就是当前状态的后继状态。

(b) 有三个容器，容量分别为 12 升、8 升、3 升。你可以将任一容量装满水，清空，或是将水移动到其他容器。你需要配出恰好 1 升的水。

【解】:

状态：12L、8L、3L 容器中的水量。可以用一个三元组 (x, y, z) 表示。

行动：进行一次合法的清空、装满或是水的移动操作。

代价：每次行动消耗 1 次操作，或者 1 步。

初始状态：全空。三元组为 $(0, 0, 0)$。

目标状态：某一个容器中含有 1 升水，其它两个容器随意。即 $(1, y, z)$ 或 $(x, 1, z)$ 或 $(x, y, 1)$。

产生后继状态：从当前状态开始，可以清空一个有水的容器，装满一个不满的容器，或是将 a 容器中的水倒到 b 容器（没有量筒，要么 a 容器的水全都倒进 b 停止，要么 b 被倒满停止）。一次操作之后三个容器的水量，就是当前状态的后继状态。

例如，一种可能的解法为：$(0,0,0) \longrightarrow (12,0,0) \longrightarrow (4,8,0) \longrightarrow (1,8,3)$。

2、在 3×3 的方格棋盘上放置分别标有数字 1，2，3，4，5，6，7，8 的八张牌，初始状态为 S_0，目标状态为 S_t。可以使用的算符有：空格左移、空格右移、空格上移、空格下移，对应把位于空格左、右、上、下的牌移入空格。设计一种 A* 算法寻找从初始状态到目标状态的路径（应不同于深度优先、宽度优先和等费用搜索）。定义 g 函数和 h 函数，画出求解该问题的搜索树，指出该问题的搜索结果（路径），搜索树中应说明每一步中 g 函数、h 函数和代价函数的取值。

1	2	3
4	5	6
7	8	

S_0

1	2	3
	4	6
7	5	8

S_f

图 2: 第 2 题

【解】：定义 $g(n)$ 为从初始状态到 n 所经历的转移步数。$g(S_0)=0$。定义 $h(n)$ 为 n 状态下所有数码到其最终位置的曼哈顿距离之和。$h(S_f)=0$。搜索结果路径标为红色。

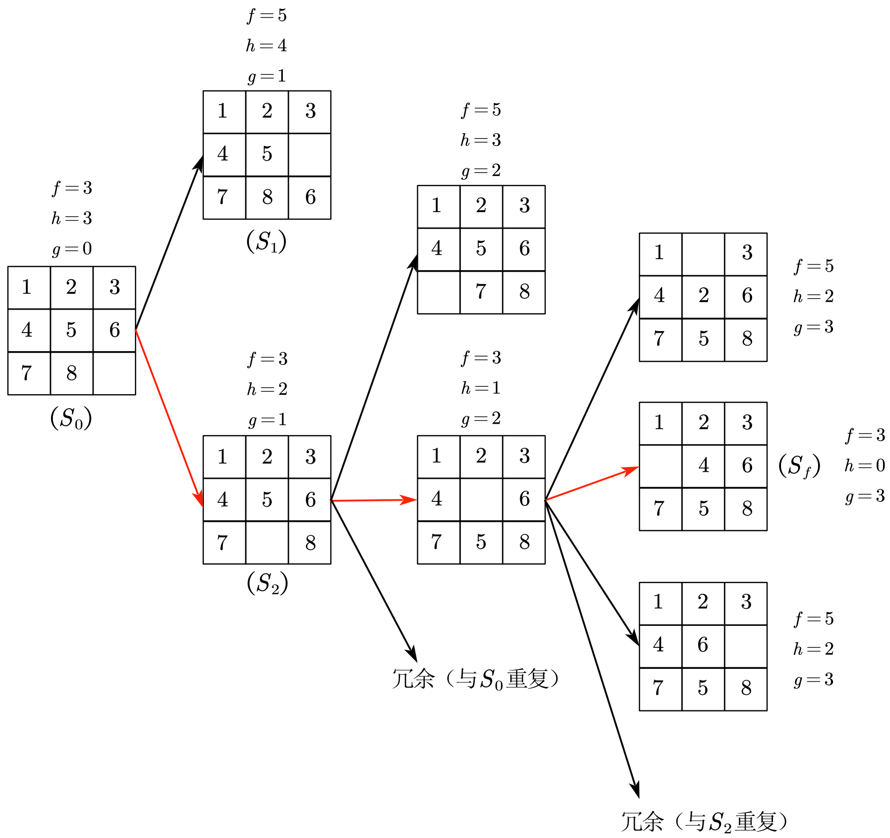

图 3: 第 2 题-搜索树

4、设 $f(n)$ 和 $g(n)$ 为两个函数，请证明或否定以下假设（下述公式中“=”符号与“∈”符号等价，n 为正实数）：

(a) 如果 $f(n) = O(g(n))$，那么 $g(n) = O(f(n))$。

【解】：错误。设 $f(n) = n$，$g(n) = n^2$。存在常数 $c = 1$ 和 $n_0 = 1$，当 $n \geq n_0$ 时，均有 $f(n) \leq cg(n)$，也即 $f(n) = O(g(n))$。但反过来不成立。无论 c 取多大，总能找到 n_0，当 $n \geq n_0$ 时，$g(n) > cf(n)$，因此 $g(n) \neq O(f(n))$。

(b) $f(n) = O\left(f(n)^2\right)$。

【解】：错误。设 $f(n) = \dfrac{1}{n}$。无论 c 如何取，总存在 n_0，使得当 $n \geq n_0$ 时，$f(n) > cf(n)^2$，也即 $f(n) \neq O\left(f(n)^2\right)$。

(c) 如果 $f(n) = O(g(n))$，那么 $\log(f(n)) = O(\log(g(n)))$，其中 $\log(g(n)) \geq 1, f(n) \geq 1$ 对足够大的 n 成立。

【解】：正确。取 n_0 足够大，由 $f(n) = O(g(n))$ 可知必然存在常数 c，满足当 $n \geq n_0$ 时，有 $f(n) \leq cg(n)$。n_0 足够大则 $\log(g(n)) \geq 1, f(n) \geq 1$ 成立，这说明 c 必为正数。

取某正数 k，满足 $k > 1 + \log c$ 且 $k > 1$。当 $n \geq n_0$ 时：

I. 若 $\log c > 0$，则对 $f(n) \leq cg(n)$ 两边取对数，有：（利用 $\log(g(n)) \geq 1$）

$$\begin{aligned}\log(f(n)) \leq \log(cg(n)) &= \log c + \log(g(n)) \\ &\leq \log c \cdot \log(g(n)) + \log(g(n)) \\ &= (1 + \log c) \cdot \log(g(n)) \\ &< k \log(g(n))\end{aligned}$$

II. 若 $\log c \leq 0$，则显然：

$$\log(f(n)) \leq \log(cg(n)) = \log c + \log(g(n)) \leq \log(g(n)) < k \log(g(n))$$

因此，综上所述，$\log(f(n)) = O(\log(g(n)))$。

(d) 如果 $f(n) = O(g(n))$，那么 $g(n) = \Omega(f(n))$。

【解】：正确。由 $f(n) = O(g(n))$，存在 c 和 n_0 使得 $n \geq n_0$ 时都有 $f(n) \leq cg(n)$。一般来说 $c > 0$（在《数据结构》中是这样定义的），因此 $g(n) \geq \dfrac{1}{c} f(n)$。也即 $g(n) = \Omega(f(n))$。

(e) $f(n) + o(f(n)) = \Theta(f(n))$。

【解】：正确。因为：

$$\lim_{n \to +\infty} \frac{f(n) + o(f(n))}{f(n)} = 1$$