

人工智能导论作业 1

自 64 赵文亮 2016011452

2018 年 11 月 13 日

目录

问题 1	2
问题 2	2
问题 3	2
问题 4	3
问题 5	3
问题 6	4
问题 7	5
问题 8	6

问题 1

- a)
- 状态：当前所处景点
 - 初始状态：圆明园
 - 目标状态：十渡
 - 后继函数：从当前景点 X_i 出发，从可以直接到达的下个景点中选择一个 X_j 作为后继状态
 - 代价函数：从圆明园玩到十渡路上花费的时间
- b)
- 状态： $\mathbf{X} = (X_1, X_2, \dots, X_9)$ ，其中 X_i 表示第 i 个小方格上当前数字（用 9 表示小方格为空格）。
 - 初始状态： $\mathbf{X}_{\text{init}} = \text{permutation}(1, 2, \dots, 9)$ ，为 1 ~ 9 的任意一种排列
 - 目标状态： $\mathbf{X}_{\text{dst}} = (1, 2, \dots, 9)$
 - 后继函数：从当前状态分别尝试向上、下、左、右移动一步，所有可能出现的状态均为后继状态
 - 代价函数：从初始状态到目标状态移动总步数
- c)
- 状态： $\mathbf{X} = (X_1, X_2, X_3)$ ，其中 X_i 表示第 i 个容器中水的体积
 - 初始状态： $\mathbf{X} = (0, 0, 0)$
 - 目标状态： $\exists i, X_i = 1 (i = 1, 2, 3)$
 - 后继函数：
 - 装满： $X_1 \leftarrow 12$ 或 $X_2 \leftarrow 8$ 或 $X_3 \leftarrow 3$
 - 清空：任取 $i (i = 1, 2, 3)$ ， $X_i \leftarrow 0$
 - 移动：任取 $i \neq j (i, j \in 1, 2, 3)$ ，如果 $X_i + X_j \leq V_j$ (V_j 表示第 j 个容器的容积)，则 $X_i \leftarrow X_i + X_j$ ， $X_j \leftarrow 0$

问题 2

- a) 在一致代价搜索中，如果任意一个状态到后继状态的代价都为固定值，则退化为宽度优先搜索。
- b) 在最佳优先搜索中，如果从当前状态扩展出的后继状态的启发函数相等，则退化为深度优先搜索。
- c) 在 A^* 算法中有 $f(n) = g(n) + h(n)$ ，如果 $h(n) = 0$ ，则 A^* 退化为一致代价搜索。

问题 3



- a) 正确。深度优先搜索随机扩展任意的未被访问过的后继节点，而使用可采纳启发函数的 A^* 搜索只会扩展评价函数 $f(n)$ 最小的后继节点，所以相对于 A^* 来讲，深度优先搜索会扩展更多无用节点或代价更高的节点，所以该命题正确。
- b) 正确。可采纳的定义为：设节点 n 到目标的真是路径代价为 $h^*(n)$ ，若启发函数 $h(n)$ 满足 $h(n) \leq h^*(n)$ 则称 $h(n)$ 为可采纳的。而 $h(n) = 0$ 显然满足这个条件。
- c) 如果考虑机器人的运动学和动力学，则正确。由于 A^* 算法是基于有限的节点，而机器人动作、状态、感知都是连续的，无法使用 A^* 求解。如果考虑机器人的路径规划（例如走迷宫），可以使用 A^* 搜索规划出大致路线，再进行进一步控制。

- d) 宽度优先搜索算法与代价无关，而只有在深度有限的时候宽度优先搜索才是完备的，所以该命题错误。
- e) 正确。根据可采纳的定义（见 b），曼哈顿距离是假设车辆可以跳过其他车辆的乐观估计，所以可以作为可采纳的启发函数。

问题 4

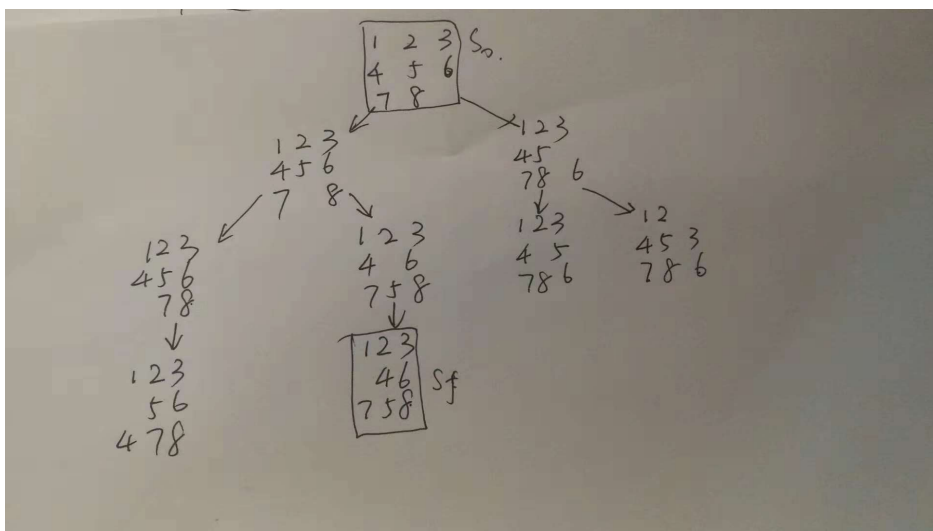
- a) 错误。设 $f(n) = n, g(n) = n^2$ ，则存在 $c_1 = 1, n_0 = 1$ ，当 $n \geq n_0$ 时， $n \leq c_1 n^2$ ，即 $f(n) = \mathcal{O}(g(n))$ 。而反过来若 $g(n) = \mathcal{O}(f(n))$ ，则 $\exists n'_0, c_2$ ，使得 $n^2 \leq c_2 n \forall n \geq n_0$ 成立。解得 $n \leq c_2$ ，矛盾。
- b) 错误。令 $f(n) = 1, g(n) = n$ ，若 $f(n) + g(n) = \Theta(\min(f(n), g(n)))$ ，则 $\exists c_1 > 0, c_2 > 0, n_0, s.t. c_2 g(n) \leq \min(f(n), g(n)) \leq c_1 g(n) \forall n \geq n_0$ 成立，即 $c_2 n \leq 1 \leq c_1 n, \forall n \geq n_0$ ，显然不成立。
- c) 正确。由已知， $\exists c_1, n_0, s.t. f(n) \leq c_1 g(n)$ ； $\exists n_1, f(n) \geq 1, \forall n \geq n_1$ ； $\exists n_2, \log(g(n)) \geq 1, \forall n \geq n_2$ 。令 $n_3 = \max(n_0, n_1, n_2)$ ，则 $\forall n \geq n_3$ ，有 $\log(f(n)) \leq \log(c_1) + \log(g(n)) \leq (\log(c_1) + 1) \log(g(n)) \triangleq c_3 \log(g(n))$ 。即 $\log(f(n)) = \mathcal{O}(\log(g(n)))$
- d) 错误。设 $f(n) = 2n, g(n) = n$ ，则 $f(n) = \mathcal{O}(g(n))$ 。而若 $\exists c_1, n_0, s.t. 2^{f(n)} \leq c_1 2^{g(n)}, \forall n \leq n_0$ 则有 $2^{2n} \leq c_1 2^n$ ，或 $n \leq \log_2 c_1$ ，矛盾。
- e) 错误。令 $f(n) = \frac{1}{n}$ ，若 $\exists c_1, n_0, s.t. f(n) \leq c_1 f^2(n), \forall n \geq n_0$ ，即 $\frac{1}{n} \leq \frac{c_1}{n^2}$ ， $n \leq c_1$ ，矛盾。
- f) 正确。若 $f(n) = \mathcal{O}(g(n))$ ，则 $\exists c_1, n_0, f(n) \leq c_1 g(n), \forall n \geq n_0$ ，即 $g(n) \geq \frac{1}{c_1} f(n) \triangleq c_2 f(n), \forall n \geq n_0$ ，则 $g(n) = \Omega(f(n))$
- g) 不正确。令 $f(n) = 4^n$ ，则 $f(n/2) = 2^n$ 。由 $c_2 2^n \leq 4^n \leq c_1 2^n, \forall n \geq n_0$ ，可得 $2^n \leq c_1$ ，矛盾。
- h) 正确。由于

$$\lim_{n \rightarrow \infty} \frac{o(f(n))}{f(n)} = 0$$

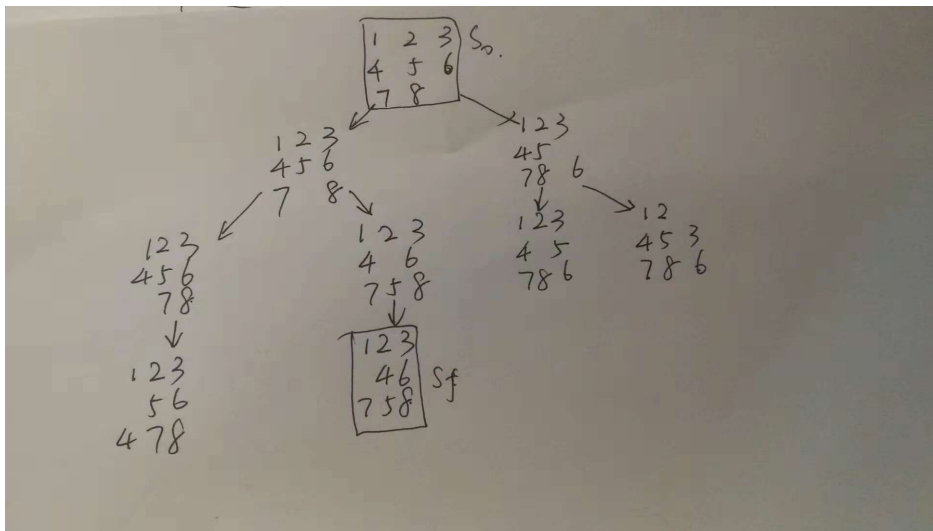
则 $\exists n_0, |o(f(n))| \leq 0.5|f(n)|, \forall n \geq n_0$ ，则 $0.5f(n) \leq f(n) + o(f(n)) \leq 1.5f(n), \forall n \geq n_0$ ，故 $f(n) + o(f(n)) = \Theta(f(n))$

问题 5

- 1) 宽度优先搜索和深度优先搜索的搜索树分别如图 1a 和图 1b 所示。其中由于深度优先搜索的步数过多，将不含目标节点的子树省略。
- 2) 宽度优先搜索的优势：在目标节点深度有限的情况下（例如本例），可以找到最优解。且时间复杂度为 $\mathcal{O}(b^d)$ ，比深度优先的 $\mathcal{O}(b^m)$ 小；深度优先搜索的优势：空间复杂度为 $\mathcal{O}(bm)$ ，比宽度优先搜索的 $\mathcal{O}(b^d)$ 要小很多，节约了大量内存。
- 3) 设 $g(n)$ 为到达节点 n 时走过的步数， $h(n)$ 为所有数码到目标位置的曼哈顿距离之和。搜索过程如图 2 所示



(a) 宽度优先搜索



(b) 深度优先搜索

图 1: 八数码问题宽度/深度优先搜索树

问题 6

$\alpha - \beta$ 剪枝是一种在对抗搜索中很有效的算法。在双人对抗搜索的搜索树中，从某一方来看，己方决策时的层为 MAX 层，对方决策的层为 MIN 层。最基本的极大极小算法求解时扩展了所有需要的节点，极大地浪费了时间和空间。而在 $\alpha - \beta$ 剪枝中，设 α 为极大层节点的下界， β 为极小层节点的下界。这样如果发现某一极小层节点的 β 值小于等于它任意父节点的 α ，则可以停止对该节点的搜索，即为 α 剪枝；如果发现某一极大层节点的 α 值大于等于它任意父节点的 β ，则可以停止对该节点的搜索，即为 β 剪枝。在相同的时间里， $\alpha - \beta$ 剪枝可以预测大约 2 倍的步数。

使用 $\alpha-\beta$ 剪枝与不使用 $\alpha-\beta$ 剪枝的结果分别如图 3 和图 4 所示。不用 $\alpha-\beta$ 剪枝一共扩展了 12 个终叶节点, 而使用 $\alpha-\beta$ 剪枝一共扩展了 7 个终叶节点, 可见 $\alpha-\beta$ 剪枝确实可以提高效率。

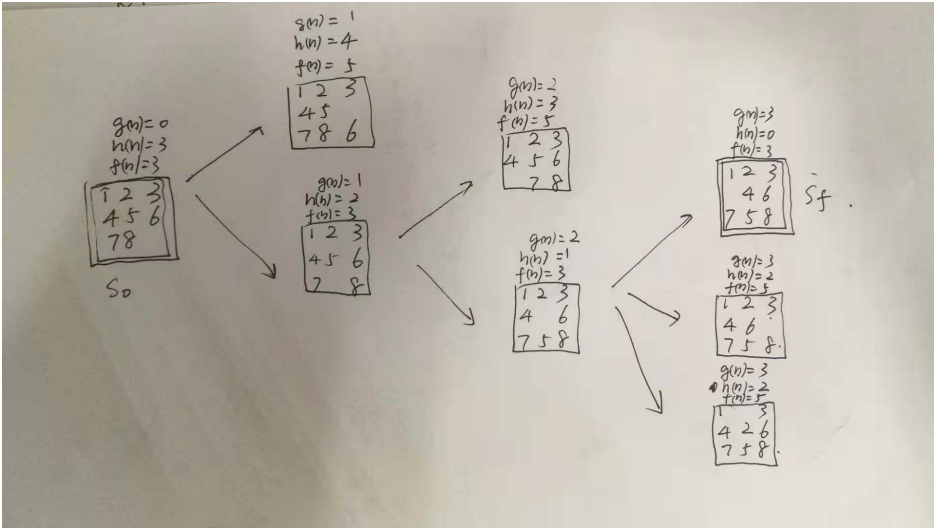


图 2: 八数码问题 A* 算法搜索树

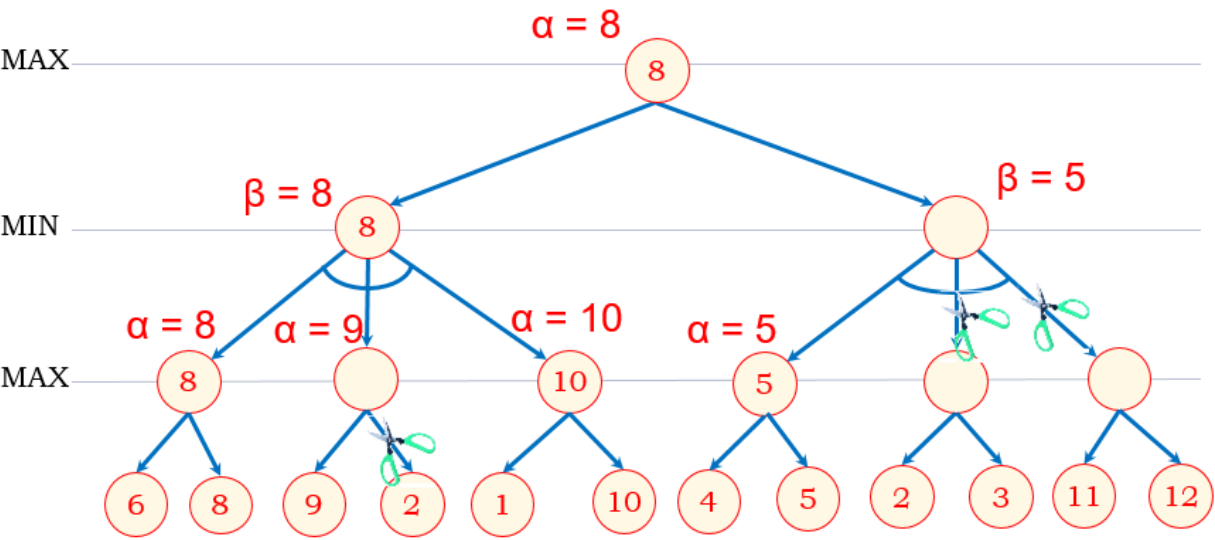
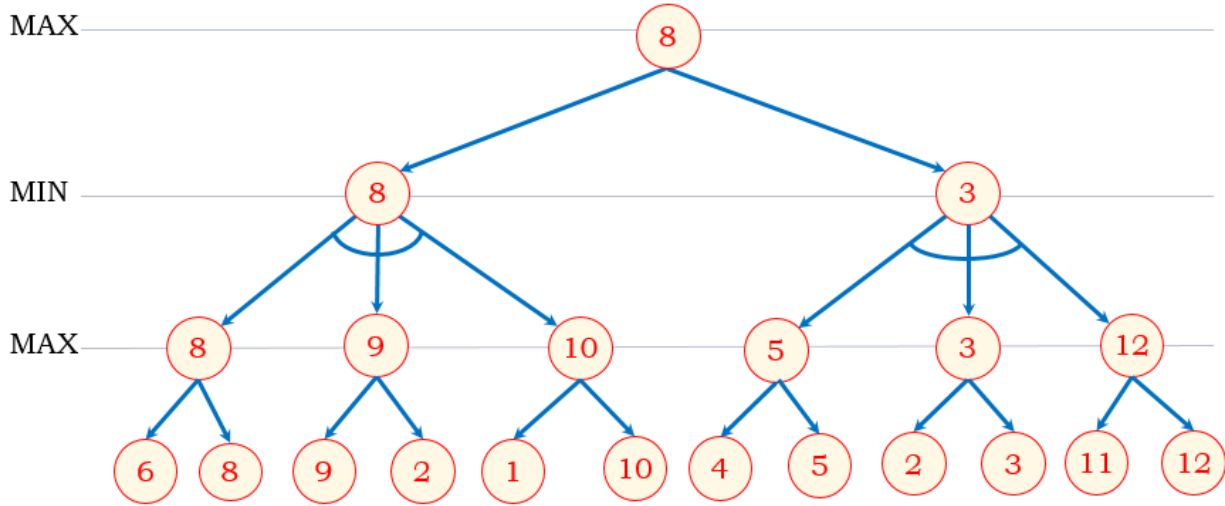


图 3: 使用 $\alpha - \beta$ 剪枝

问题 7

- a) 不考虑一个格子至多一架飞行器的约束下，任意时刻每个飞行器所在的位置有 n^2 种可能性，则状态空间为 $\mathbf{X} = (X_1, X_2, \dots, X_n)$ ，其中 $X_i = (x_i, y_i)$ 为第 i 个飞行器所处的坐标。状态空间大小为 n^{2n} 。
- b) 每次可以对任一飞行器执行至多 5 种操作，其中保持不动的操作认为状态没有变化。则分支因子为 $b = 4n$ 。
- c) 令启发函数 h_i 为第 i 架飞行器到目的地的曼哈顿距离。很显然，当空域中没有其他飞行器时，这个 h_i 即为真实值 h_i^* 。
- d) 选择 $h = \sum_{i=1}^n h_i$ 。这是一个一致的启发函数，下面证明。首先证明 h_i 是一致的。设第 i 个飞行器的终点



图 4: 不使用 $\alpha - \beta$ 剪枝

为 (\hat{x}_i, \hat{y}_i) , 则

$$\begin{aligned}
 h_i(n) &= |x_n - \hat{x}_i| + |y_n - \hat{y}_i| \\
 &= |x_n - x_{n'} + x_{n'} - \hat{x}_i| + |y_n - y_{n'} + y_{n'} - \hat{y}_i| \\
 &\leq |x_n - x_{n'}| + |x_{n'} - \hat{x}_i| + |y_n - y_{n'}| + |y_{n'} - \hat{y}_i| \\
 &= h_i(n') + c(n, n')
 \end{aligned}$$

则 h_i 是一致的。则显然 $h = \sum_{i=1}^n h_i$ 也是一致的, 证毕。同时显然可知, h 可以由原问题的松弛问题得到。另一方面, (2)(3) 选项并不能从原问题的松弛问题中得到, 所以选择第一个启发函数。

问题 8

- 1) 是。从下一问的解答中会给出所有追上的最短路径。
- 2) 分步数讨论, 以棋盘的行列表示坐标, 例如 (i, j) 表示从上到下数第 i 行、从左到右数第 j 列的方格。初始状态下, 兵位于 2 点, 即 $(1, 1)$; 马位于 $(8, 2)$ 。

- 1 步或 2 步: 此时兵最终到达在 3 或 4 点, 而马在竖直方向至多向上前进两个格, 不可能追上;
- 3 步: 此时兵最终到达 5 点, 坐标为 $(3, 2)$, 马有可能追上兵。搜索过程如图 5 所示。每次扩展节点时, 设当前点为 $n(x_n, y_n)$, 终点为 $d(x_d, y_d) = (3, 2)$, 已走步数为 s , 若 $|x_d - x_n| > 2(3 - s)$ 或 $|y_d - y_n| > 2(3 - s)$ 则说明在剩下的几步中根本不可能走到终点, 则不再扩展这种节点。

从图中可以看到, 3 步时存在四条路径, 所以这些就是所求最短路径, 分别为

1. $(8, 2) \rightarrow (6, 1) \rightarrow (5, 3) \rightarrow (3, 2)$
2. $(8, 2) \rightarrow (6, 3) \rightarrow (5, 1) \rightarrow (3, 2)$
3. $(8, 2) \rightarrow (6, 3) \rightarrow (4, 4) \rightarrow (3, 2)$

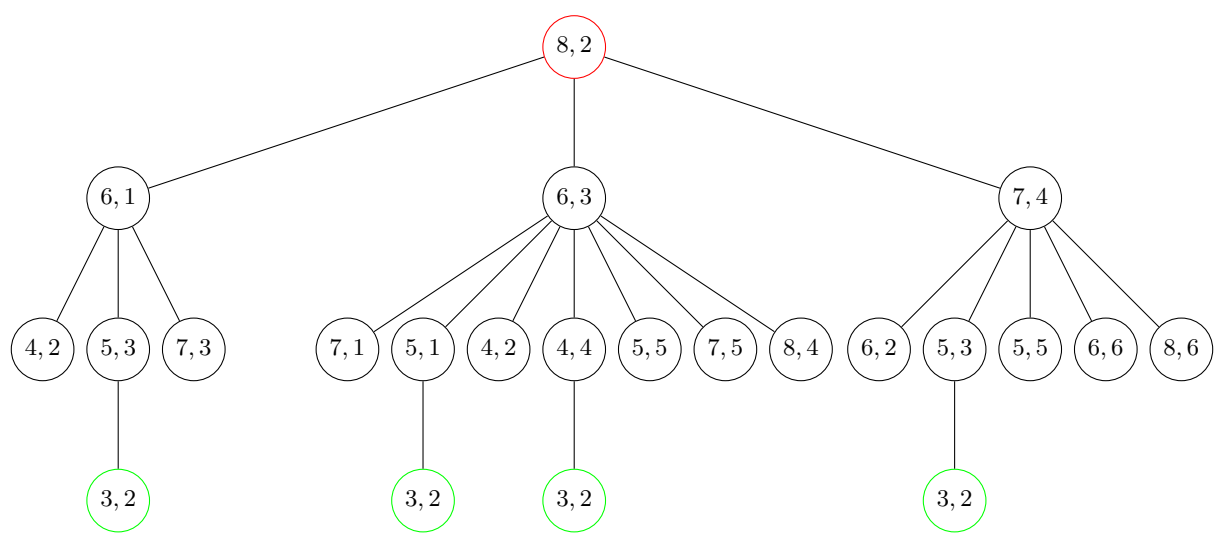


图 5: 马追兵问题搜索过程

4. $(8, 2) \rightarrow (7, 4) \rightarrow (5, 3) \rightarrow (3, 2)$