第一次作业 | 人工智能

- 管仁阳-519021911058

第一题

以下回答采用词汇对应关系：宽度优先搜索（*BFS*），一致代价搜索（*Uni-cost*），深度优先搜索（*DFS*）

从Completeness, Optimality, Time complexity, 和 Space complexity角度分析。
= 表示不优于，≥ 表示在一定条件下不优于，> 表示一般情况下优于，并不表示数值大小。

Complexity: BFS = DFS(graph-version) ≥ Uni-cost > DFS(tree-version)

- BFS是complete：如果解结点的深度是一个有限值，那么BFS一定可以在扩展完比它更浅的节点以后扩展到它。
- DFS(graph-version)也是complete：一定可以搜索完整个图。
- Uni-cost 只有在保证所有的$step\ cost > \epsilon$ 时，才complete；否则会陷入$path\ cost = 0$ 的自环中
- DFS(tree-version)不是complete：很容易陷入环（相对graph而言，在tree中表现为）中。

Optimality：Uni-cost ≥ BFS >DFS

- Uni-cost 在一般情况下（$step\ cost \geq 0$）是optimal的。因为cost非负，一条path上随着新节点的加入path cost不会降低。当扩展到一个节点时，到这个节点的最优路径就已经被找到了。
- BFS 只有当节点的path cost和depth一样时才是optimal的。
- DFS 的graph-version和tree-version都不optimal。

Time complexity：BFS = DFS > Uni-cost

- BFS：$O(b^d)$
- DFS：$O(b^d)$
- Uni-cost：$O(b^{1+\lfloor C^*/\epsilon \rfloor}) >> b^d$。$iff$ 所有$step\ cost$相等时才等于b^d）

Space complexity：DFS = BFS > Uni-cost

- DFS：$O(bm)$
- BFS：$O(b^d)$
- Uni-cost：$O(b^{1+\lfloor C^*/\epsilon \rfloor}) >> b^d$。

第二题

二者的主要区别在于**是否引入与具体问题相关的外部信息。**

对于**无信息搜索**来说，选择下一个扩展的节点只有三种情况：1. 随机选择（如$step\ cost = 1$的BFS）2. 根据图（或树）结构选择（如DFS选择depth最大的结点）3. 根据问题Formulation中的cost来选择（如$step\ cost \neq 0$的BFS和Uni-cost Search）。

对于**有信息搜索**来说，借助与问题本身性质相关、而与Formulation无关的启发函数 $h(n)$ 来选择下一个要expand的结点，如路径规划问题中的path cost可以定义为实际路程，$h(n)$ 可以定义为欧氏距离或曼哈顿距离。

尽管path cost和启发函数都可以用来从最优性的角度对candidate结点排序，但是他们本本质是不同的。具体体现在：1. path cost是问题formulation中定义的、唯一的，而启发函数是根据问题的其他属性定义的、不唯一的 2.path cost是搜索的目的、判别Optimality的依据，而启发函数只是一种近似地衡量与goal的距离的方式 3. path cost本身就可以理解为搜索的目的（搜索就是为了寻找最小的path cost），而启发函数只是加速搜索过程的手段。因此，**有、无信息搜索的分界岭只有是否使用了问题本身的信息**

第三题

- *States*：每一个状态都由障碍物的位置、机器人的位置、机器人的朝向确定。一个网格有五种状态：1. 空 2. 障碍物 3. 机器人一 4. 机器人二 5. 同时有两个机器人；若是机器人，则又有四种朝向。

  State space size：设 $M \times N = n$，当大小为 n 的棋盘上有 k 个障碍物时，共有 $S_k = C_n^k \times [(n-k) \times 4] \times [(n-k-1) \times 4] = 16C_n^k \times A_{n-k}^2 = 16n(n-1)C_{n-2}^k$

  此处应为n-k，不需要-1因为两个机器人可以在一个盘子里。

  则 State space 的大小为

  不需要对k求和，因为k是给定的量

$$\begin{aligned} S = \sum_{k=0}^{n-2} S_k &= \sum_{k=0}^{n-2} 16n(n-1)C_{n-2}^k \\ &= 16n(n-1) \times 2^{n-2} \\ &= n(n-1)2^{n+2} \\ &= 16n(n-1) \times 2^{n-2} \\ &= n(n-1)2^{n+2} \\ &= M \times N(M \times N - 1)2^{M \times N + 2} \end{aligned}$$

- *Actions*：机器人一、机器人二都可以：1. 左转90° 2. 右转90° 3. 直行一个单位。
- *Goal test*：当某一种state中出现 一个格子中有两个机器人 时，说明达到搜索目标。
- Step cost：设直行耗能$E_{forward}$，旋转耗能E_{turn}，则：*1.* 两个机器人都直行一个单位时 $step\ cost = 2E_{forward}$ *2.* 只有一个机器人直行，另一个机器人旋转时 $step\ cost = E_{forward} + E_{turn}$ *3.* 两个机器人都旋转时$step\ cost = 2E_{turn}$（注：此处step cost不一定要取这种定义但一定要是可累加的量，如每一步的 $cost = 1$。我们希望在找到解的同时满足一定的要求，但是这并非题目的要求）*
- 启发函数：$h(n) = g(n) \times E_{forward}$，$g(n)$ 是两个机器人之间的 ***曼哈顿距离**

  可接受性：易证得。棋盘上两个机器人要移动到一起至少需要按照曼哈顿距离对应的路径移动，而在此问题中沿此路径移动的cost至少为$E_{forward}$。因此 $h(n) < f(n)$，$f(n)$是真实最优path cost。 易证得该启发函数也满足**一致性**

第四题

1. BFS就是认为$step\ cost = 1$的UCS，而UCS考虑了更一般（$step\ cost \neq 1$）的情况。当$step\ cost = 1$时，BFS和UCS等价。
2. UCS就是认为$h(n) = 0$的A*，而A* 考虑了更一般（$h(n) \neq 0$）的情况。当$h(n) = 0$时，UCS和A* 等价。
3. 启发函数$h(n)$满足一致性，即$h(n) \leq c(n, a, n') + h(n')$，$n'$为$n$的后继。若启发函数满足一致性，则 A* 算法中：

$$f(n') = g(n') + h(n') = g(n) + c(n, a, n') + h(n') \geq g(n) + h(n) = f(n)$$

4. 假设一条path=$n_1, n_2, \cdots, n_k$，而启发函数满足一致性即：

$$h(n_i) \leq c(n_i, a_i, n_{i+1}) + h(n_{i+1})$$

则

$$h(n_i) - h(n_{i+1}) \le c(n_i, a_i, n_{i+1})$$

因为启发函数满足 $h(n_k) = 0$，两边求和可得

$$h(n_1) = h(n_1) - h(n_k) = h(n_1) = \sum_{i=1}^{k-1} [h(n_i) - h(n_{i+1})] \le \sum_{i=1}^{k-1} c(n_i, a_i, n_{i+1}) = path\ cost$$

即满足可接受性。

第一次作业-反馈

作业完成情况：本课程一共 81 名同学，其中有 77 名同学按时完成了作业。目前仍然有几位同学未提交作业，我已经在 canvas 系统里私发了邮件，希望未提交作业的同学及时完成补交。

1 本次作业可能用到的知识点

本次作业可能会用到以下知识点：

（1）无信息搜索 (uninformed search)：宽度有限算法 (BFS)、一致代价搜索 (UCS)、深度优先算法 (DFS) 的定义、过程、复杂度、性能。

（2）有信息搜索 (informed search)：A^* 搜索的定义、过程。

（3）启发函数的可接受性 (admissible)、一致性 (consistent): 如果启发函数 h 对于每个节点 n 来说都有 $h(n)$ 小于等于从节点 n 到达终点的实际代价 $h^*(n)$，则该启发函数满足可接受性。如果一个启发函数 h 满足每个父节点 n 到达子节点 n' 所花费的代价 $c(n,n',a)$ 加上 $h(n')$ 大于等于 $h(n)$，则称该启发函数 h 满足一致性。

（4）将问题形式化的方法，状态、行动、目标测试、路径耗散的定义。状态空间的大小指的是空间中所有可能的状态数目，例如在 3×3 大小的空间中，某种物体出现在其中，那么对于空间中的每个位置而言有物体出现在该位置和未出现在该位置两种状态，空间的状态一共有 $2^{3\times3}$ 种。

2 第一题

比较宽度优先算法（BFS）一致代价搜索（UCS）、深度优先算法（DFS）的优劣。

解答：可以比较三种算法的完备性、最优性、复杂度等角度比较。BFS 算法时完备的、单位代价的情况下时最优的，但是需要保存所有产生的节点值、空间复杂度高；UCS 算法对于一般性的步骤代价而言是最优的；BFS 与 UCS 算法的时间空间复杂度均为 $O(b^d)$。DFS 算法不是完备的、也不能保证最优性，但是只需要存储一条从根节点到叶节点的路径、以及该路径上每个节点上所有未被扩展的兄弟节点即可，当一个节点被扩展，当它的所有节点都被探索过后该节点就从内存中删除、空间复杂度低，为 $O(bm)$，其中 m 为最大深度。

3 第二题

说明有信息搜索与无信息搜索的区别。

解答：无信息搜索是指我们不知道接下来要搜索的状态哪一个更加接近目标的搜索策略，因此也常被成为盲目搜索。有信息搜索需要访问启发式函数 $h(n)$ 来估算从当前节点到目标的解代价，从而衡量哪一个状态更加接近目标状态，并优先对该状态进行搜索，因此与无信息搜索相比往往能够更加高效得解决问题。

4 第三题

假如我们现在有一个长度为 M 个格子、宽度为 N 个格子的棋盘，盘子中有一些障碍物、形成了一个迷宫。有两个机器人希望根据他们的位置让他们尽快相遇（相遇指的是让两个机器人处在同一个格子里即可，不需要让他们面对面朝向）。机器人在一个行动单位内可以沿着自己朝向的方向移动到一个相邻的格子内、或者旋转自己朝向的方向 $90°$。

（1）请将该问题形式化，我们需要如何简单地定义该问题的状态（state)?

（2）对于该状态而言，你定义的状态其状态空间有多大（size)?

（3）继续形式化该问题，描述该问题的行动、目标测试、路径耗散（简单描述即可）。

（4）如果使用搜索算法解决该问题，请给出一个启发函数 h，该启发函数满足可接受性。

解答：（1、2）答案 a（这是我最开始的想法）：在该问题中，机器人的朝向和位置都在不停改变，我们定义的状态需要包含以上信息。而对于障碍物，一般而言我们认为障碍物不变，可以用一个额外的数组来储存其信息，并且在搜索过程中不断比对机器人的位置和障碍物的信息以剪枝。

机器人位置：一个机器人共有 $M \cdot N$ 种可能，两个机器人共有 $(MN)^2$ 种可能。注意两个机器人可能在同一个格子里，故两个机器人位置的可能并不是 C_{MN}^2。我们在考虑状态空间的时候只需要考虑该状态是否可能存在、是否符合题意，不需要考虑其与目标之间的关系。

机器人朝向：一个机器人共有上下左右 4 种可能，两个机器人一共有 16 种可能。

考虑到共有两个机器人，因此空间大小为 $16 \cdot (MN)^2$。

答案 b：看到大家提交的作业以后，我看到有不少同学提出了这样的想法，我认为也很有道理。在方案 a 中我们认为障碍物是一道"墙"，它会干扰机器人移动的方向，但是不会占据一个格子。事实上，障碍物也有可能是完全占据一个格子的"柱子"，这一点是我出题目的时候没有想到。如果障碍物时占据一个格子的"柱子"，那么障碍物会影响机器人在棋盘上可能布局的位置。

机器人位置：假如棋盘上有 k 个这样的"柱子"障碍物，那么机器人的位置有 $(M \cdot N - k)$ 种可能，注意两个机器人可能在同一个格子里。

这样总的空间大小为 $16 \cdot (MN - k)^2$

答案 c：有同学提出，这两个机器人有可能是相同的，这样交换两个机器人所在的位置（保持两个机器人朝向相同），其实交换前后环境的状态仍然保持一致。事实上这样的问题在现实生活中是存在的，例如 n 皇后问题，我们的 n 枚棋子事实上就是"相同的机器人"，我们关注的是他们在棋盘上的布局，交换其中的两枚棋子仍然是相同的状态。

机器人的位置：假如棋盘上的障碍物是"墙"，那么两个机器人的位置一共有 $(C_{MN}^2 + MN)$ 种可能（两个机器人在不同的格子 + 两个机器人在相同的格子）。

这样总的状态大小为 $16 \cdot (C_{MN}^2 + MN)$。

（3）行动：每个机器人可以采取移动一格或者转变 90°；目标测试：两个机器人是否在同一格；路径耗散：到达当前状态花费的时间

（4）启发函数 h 需要满足可接受性，指每个状态到最终目标花费的实际时间成本需要高于其启发式函数值。因此可以设计启发函数 h 为两个机器人的曼哈顿距离或者欧氏距离，由于机器人一次只能移动到相邻格子、不能斜着移动，因此两个机器人之间的最短距离会大于等于其曼哈顿距离与欧氏距离。除此以外，还需要考虑机器人转向的时间成本，因此实际到达目标花费的时间成本会大于等于启发函数 h。

其他的答案，例如曼哈顿距离处以 2 等，只要能保证实际转移的花费大于启发函数，也是符合要求的。

5 第四题

证明以下结论（仅需要简单说明证明思路即可，不用写太多）：

（1）BFS 搜索算法是 UCS 搜索算法的特殊情况。

（2）UCS 搜索算法是 A^* 算法的特殊情况。

（3）运行 A^* 算法时，若启发式函数 h 满足一致性，那么在启发式搜索算法的搜索树中每条路径的子节点的 f 值大于等于其父节点。

（4）若启发式函数 h 满足一致性，那么它也会满足可接受性。

解答：（1）如果在运行 UCS 算法时取各步骤的花费均一致，那么会使得计算过程中 $g \propto depth(n)$，从而该特殊情况下的 UCS 算法实际上就是 BFS 算法

（2）在运行 A^* 算法时，如果将启发函数 h 取为 0，那么该特殊情况下的 A* 算法实际上就是 UCS 算法。

（3）若启发式函数 h 满足一致性，即搜索树路径中一个父节点点 n 到另一个子节点 n' 满足

$$h(n) <= h(n') + c(n, n', a)$$

其中 $c(n, n', a)$ 指的是从节点 n 扩展到节点 n' 所采取的行动 a 花费的实际代价。那么我们自然有

$$h(n) + g(n) <= h(n') + c(n, n', a) + g(n)$$

其中 $g(n)$ 指的是算法从初始状态扩展到节点 n 所花费的实际代价，那么自然有 $h(n)+g(n) = f(n)$ 与 $g(n) + c(n, n', a) = g(n')$，代入以上不等式即有 $f(n) <= f(n')$。

（4）我们可以选取任意一个节点 G，那么其他节点都会有一条到 G 的最短路径。如果我们选取任意一条最短路径上距离 G 一个步骤的某个节点 n_1，那么我们根据启发式函数的一致性，会有 $h(n_1) <= h(G) + c(n_1, G, a)$。如果我们用 h^* 来表示某点到终点的实际最小代价，那么就会有 $h(n_1) <= h^*(n_1)$，即 n_1 可以满足启发函数的可接受性。之后我们再将 n_1 不断扩展，对于距离 G 最短距离有两个步骤的节点 n_2，我们用类似的方法可以得到 $h(n_2) <= h(n_1) + c(n_2, n_1, a) <= h^*(n_1) + c(n_2, n_1, a) = h^*(n_2)$。

注意：h 是我们估计当前节点到终点的代价，所以一般默认终点的 $h(G) = 0$。有同学提出如果 h 为一个常数的话，比如 h=1000，有可能会满足一致性而不满足可接受性。不过一般我们默认终点的 $h(G) = 0$，所以该命题仍然是成立的。

6 总结

本次作业大家完成得很不错。无论是用的手写还是打印，大家都完成得很认真。

在前两题的概念题中，基本上大家都能说到点子上、而且条件很清晰。部分同学第三题的状态空间大小计算有些错误，建议仔细阅读解答。第四题的证明中，有些同学写得过于简略，例如第三小问第四小问，建议还是要列出数学式子加以说明。除此以外，部分同学可以在 canvas 平台里查看我留下的关于作业的评语。

在问卷调查中，我发现百分之九十以上的同学都能在 2h 内完成这份作业。如果你的完成时间超过 2h，说明你对于基础概念的掌握还不够。如果对于课堂上有些概念没有弄懂，可以参考这份资料notes，里面有非常详细的概念整理。如果还是不懂，建议尽早联系助教和老师，不要堆积到考试前再解决。