1、汉诺塔

（1）可以使用一个列表来表示每个立柱上的盘子的大小和顺序，例如一个有三个盘子的起始状态可以表示为 [ [3,2,1], [], [] ]，其中第一个列表代表最左边的立柱，第二个和第三个分别代表中间和最右边的立柱。盘子的大小用数字表示，数字越大表示盘子越大，同时列表中数字的顺序也表示盘子的摆放顺序，即列表中的第一个元素表示最上面的盘子。

（2）假设有n个盘子 则状态空间大小为3^n

（3）初始状态是最左边的立柱上有 N 个大小不同的盘子，按照从小到大的顺序排列，而右边两个立柱是空的。

（4）①把一个盘子从一个立柱移到另一个立柱。②把一个小盘子放在一个大盘子上

（5）目标状态是所有 N 个盘子都被移动到最右边的立柱上，按照从小到大的顺序排列，而其他两个立柱是空的

2、n-pacman搜索

（1）这个问题的状态空间非常大，因为每个pacman都可以处于任何一个非墙的方块上，而这个迷宫中非墙的方块数可以很大

（2）每个pacman的分叉因子为5

所以分叉因子的最严上限为5^M，其中M是pacman的数量

（3）统一代价搜索算法：在搜索过程中，我们将按照每个节点的代价从小到大进行扩展，直到找到所有pacman都到达同一个位置的解。由于该问题的状态空间非常大，因此在实际中可能需要设置一个节点扩展的上限，以避免搜索时间过长或者内存不足的问题

因为节点拓展上限=C\*B^D其中C是根节点的代价，B是分叉因子，也就是每个节点可以拓展的子节点数，D是限制搜索深度的参数限制搜索深度的参数（D 可以是一个固定值，也可以根据实际情况动态调整。我们可以设置一个较小的初始值，然后在运行搜索算法时根据需要动态调整）

因此，我们可以计算出节点扩展上限为 C \* B^D。例如，如果我们将 D 设置为 10，那么节点扩展上限就是 5^10，约为 9,765,625。这意味着我们最多只能扩展约 9,765,625 个节点，然后停止搜索

3、

（1）扩张节点的顺序：Start→A→C→ B→D→Goal

从 Start 到 Goal 的路径：Start→A→C→ B→D→Goal

（2）扩张节点的顺序：Start→A→B→ C→D→Goal

从 Start 到 Goal 的路径：Start→A→C→ Goal

（3）扩张节点的顺序：Start →A→B→C→D→ Goal

从 Start 到 Goal 的路径：Start → A →C →Goal。

（4）扩张节点的顺序：Start →A→C→ Goal

从 Start 到 Goal 的路径：Start → D →Goal。

（5）扩张节点的顺序：Start→ A→ D→ C→ Goal。

从 Start 到 Goal 的路径：Start→ D→ C→Goal。

4、

（1）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 搜索策略 | A-B-D-G | A-C-D-G | A-B-C-D-F-G |  |  |  |
| 深度优先 | ABDG | ACDG | ABCDG | ACDFG |  |  |
| 广度优先 | ABDG | ACDG | ABCDG | ABDFG | ACDFG | ABCDFG |
| 统一代价 | ABDG | ACDG | ABCDG | ABDFG | ACDFG | ABCDFG |
| A\*搜索采用启发值h1 | ACDG | ABDG | ABCDG | ABDFG | ACDFG | ABCDFG |
| A\*搜索采用启发值h2 | ACDG | ABDG | ACDFG | ABDFG | ABCDG | ABCDFG |

（2）（a）[0, 6.5]，如果我们不知道B到目标状态的最小成本。

[0, 3.5]，如果B的后继节点是E，且h3(B)≤c(B,E)+h3(E)=3+1.5=4.5。

[0, 5]，如果B的后继节点是F，且h3(B)≤c(B,F)+h3(F)=2.5+4.5=7。

[0, 7]，如果B的后继节点是D，且h3(B)≤c(B,D)+h3(D)=1+7=8。

（b）[0, 6.5]，如果我们不知道B到目标状态的最小成本。

[0, 3.5]，如果B的后继节点是E，且h3(B)≤c(B,E)+h3(E)=3+1.5=4.5。

[0, 5]，如果B的后继节点是F，且h3(B)≤c(B,F)+h3(F)=2.5+4.5=7。

[0, 7]，如果B的后继节点是D，且h3(B)≤c(B,D)+h3(D)=1+7=8

（c）ℎ3(C) < ℎ3(A) （节点 C 在节点 A 的下一层，需要比节点 A 的启发值更低）

ℎ3(B) < ℎ3(C) （节点 B 在节点 C 的下一层，需要比节点 C 的启发值更低）

ℎ3(D) < ℎ3(B) （节点 D 在节点 B 的下一层，需要比节点 B 的启发值更低）

ℎ3(B) ∈ [0, ∞]，其中 ℎ3(A) = 10, ℎ3(C) ∈ (1.5, 9)，ℎ3(B) ∈ (0, 1.5)，ℎ3(D) < 0。

ℎ3(B) ∈ [0, ∞]，其中 ℎ3(A) = 10, ℎ3(C) = 9, ℎ3(B) = 1.5，ℎ3(D) < 0。