4.3 习题解析

1. 什么是图搜索过程？其中，重排 OPEN 表意味着什么？重排的原则是什么？

参考答案：

图搜索策略是一种在图中寻找解路径的方法。根据图对应的实际背景可分为或图和与/或图两种。

一般在图中搜索时，是从 OPEN 表中取出第一个节点，重排 OPEN 表意味着有不同的搜索顺序，因此，搜索到一个新节点加入 OPEN 表就需要给出它在 OPEN 表的位置(也就是排序)，其生成的后继节点可以放于以下位置。

(1) OPEN 表的尾部，算法相当于 Breadth-first-Search。

(2) OPEN 表的首部，算法相当于 Depth-first-Search。

(3) 根据启发式函数 f 的估计值确定最佳者，放于 OPEN 表的首部，算法相当于 Best-First-Search。

2. 什么是 A* 算法？它的评价函数如何确定？它与 A 算法有什么区别？

参考答案：

在或图通用搜索算法中，将启发式函数的形式定义为 $f(n)=g(n)+h(n)$，并在第(4)步按启发式函数 f 的值的大小取出一个节点，这样的或图通用搜索算法就称为 A 算法。

A 算法的启发式函数中，$g(n)$表示从 S_0 到 n 点的搜索费用的估计，因为 n 为当前节点，搜索已达到 n 点，所以 $g(n)$可计算出；$h(n)$表示从 n 到 S_g 接近程度的估计，因为尚未找到解路径，所以 $h(n)$仅仅是估计值。

若进一步规定 $h(n)\geqslant 0$，并且定义为

$$f^*(n)=g^*(n)+h^*(n)$$

其中，$f^*(n)$表示 S_0 经点 n 到 S_g 最优路径的搜索费用，也有人将 $f^*(n)$定义为实际最小搜索费用；$g^*(n)$为 S_0 到 n 的实际最小费用；$h^*(n)$为 n 到 S_g 的实际最小费用的估计。当要求 $h(n)\leqslant h^*(n)$时，就称这种 A 算法为 A* 算法。

3. 证明 OPEN 表上具有 $f(n)<f^*(s)$的任何节点 n，最终都将被 A* 算法选择进行扩展。

参考答案：

用反证法，设 $f(n)<f^*(S_0)$且 n 没有被选出来作为扩展节点。

由命题 4.4 可知，A* 算法将找到一条路径 $S_0=n_0,n_1,\cdots,n_k=S_g$，为最优路径，且 $f(n_i)\leqslant f^*(n_i)=f^*(S_0)(i=0,1,\cdots,k)$，但这个不等式中必须等号成立；否则 $f(n)<f(n_i)$。那么 A* 算法应该选择 n，而不应该选择 n_i。但对于 n_k，由于 $f(n_k)=f(S_g)=g(S_g)=g^*(S_g)=f^*(S_0)$，又因为 n_k 是被挑选出来的节点，因此应该有 $f(n_k)\leqslant f(n)$（还是因为 n 没有被选出来作为扩展节点），但这与 $f(n)<f^*(S_0)=f(n_k)$ 矛盾。

4. 见课堂演示与实践指导。

5. 见课堂演示与实践指导。

6. 请给出通用图搜索算法中，OPEN 表和 CLOSED 表所表示的一般的含义。

参考答案：

通用图搜索算法中，OPEN 表和 CLOSED 表实际上是作为表示隐式搜索图的一种数据结构，该隐式搜索图是靠这种数据结构，边搜索、边生成。其中，OPEN 表用于存放搜索过程中已经生成，且已用启发式函数作过估计或评价，但尚未产生它们的后继节点的那些节点，也称为未考查的那些节点；CLOSED 表用于存放已经生成后继节点，且已考查过的节点，也称为关闭节点，即不会再访问的节点，这就避免了回路。

7. 或图和与/或图各对应什么样的实际背景？所对应的最优搜索算法是什么？

参考答案：

或图对应的背景为搜索扩展时，可在若干分支选择其中之一（这就是“或”的意思）；与/或图则是在搜索扩展时，有可能要同时搜索若干分支（这就是“与”的含义），也有可能在若干分支选择其中之一（所以也包含“或”的意思）。

或图搜索所对应的最优搜索算法是 A* 算法，与/或图搜索所对应的最优搜索算法是 AO* 算法。

8. A* 算法有什么样的优良性质？

参考答案：

A* 算法与一般的最佳优先比较，有其特有的性质：如果问题有解，即 $S_0\rightarrow S_g$ 存在一条路径，则 A* 算法一定能找到最优解。这一性质称为**可采纳性**（Admissibility）。

9. A* 算法在什么条件下执行效果最好？为什么？

参考答案：

A* 算法的估计函数若满足单调限制，即如果对所有 n_i 与 n_j，n_j 是 n_i 的后继，则有 $h(n_i)\leqslant c(n_i,n_j)+h(n_j)$，并且 $h(S_g)=0$，即 n_j 到目标的费用估计不会大于 n_i 到目标的费用估计加上 $n_i\sim n_j$ 的费用，执行效果最好。因为 A* 算法所扩充的任一点（即用来求过后继的点）n 必在最优路上，即 A* 算法找到的是最优路径。

10. 与/或图的启发式搜索法 AO* 是通过评价函数 $f(n)=h(n)$ 来引导搜索过程，适用于分解之后得到的子问题不存在相互作用的情况。试证明：若 $S\rightarrow N$ 存在解图，当 $h(n)\leqslant h^*(n)$ 且 $h(n)$ 满足单调性限制条件时，AO* 算法一定能找到最佳解图，即在这种情况下，AO* 具有可采纳性。

参考答案：

与/或图的启发式搜索法 AO* 是通过评价函数 $f(n)=h(n)$ 来引导搜索过程，由 $h(n)$ 满足单调性限制条件可知，$h(S_g)=0$，那么在 AO* 算法搜索到的解图上，最后一层的节点 n_j 上的费用都是边的费用的累加（节点的费用是边的费用加上后继节点的费用），也就是

该节点费用为 $h^*(n_j)$，而费用的估计为 $h(n_j)$，$f(n_j)=h(n_j)\leqslant h^*(n_j)=f^*(n_j)$，因此，这层的节点 n_j 应该被选中。并且，n_j 的上一层节点 n_i 的费用依据满足单调限制条件 $h(n_i)\leqslant c(n_i,n_j)+h(n_j)\leqslant c(n_i,n_j)+h^*(n_j)=h^*(n_i)$，即 $h(n_i)\leqslant h^*(n_i)$，也即 $f(n_i)\leqslant f^*(n_i)$，所以 n_i 应该被选中，倒推上去，AO* 算法一定能找到最佳解图。

11. 请写出在 AO* 图搜索中，可解和不可解节点的递归定义。

参考答案：

可解节点的可递归定义为：终叶节点是可解节点；若 n 为一非终叶节点，且含有“或”后继节点，则只有当后继节点中至少有一个是可解节点时，n 才可解；若 n 为一非终叶节点，且含“与”后继节点，则只有当后继节点全部可解时，n 才可解。

不可解节点的可递归定义为：没有后继节点的非终叶节点为不可解；若 n 为一非终叶节点，且含有“或”后继节点，则仅当全部后继节点为不可解时，n 不可解。若 n 为一非终叶节点，且含有“与”后继节点，则至少有一个后继节点为不可解时，n 为不可解。

4.4 补充习题

1. 请阐述状态空间的一般搜索过程，其中 OPEN 表与 CLOSED 表的作用是什么？

参考答案：

先把问题的初始状态作为当前扩展节点对其进行扩展，生成一组子节点，然后检查问题的目标状态是否出现在这些子节点中。若出现，则搜索成功，找到了问题的解；若没出现，则再按照某种搜索策略从已生成的子节点中选择一个节点作为当前扩展节点。重复上述过程，直到目标状态出现在子节点中或者没有可供操作的节点为止。所谓对一个节点进行“扩展”，是指对该节点用某个可用操作，生成该节点的一组子节点。

OPEN 表用于存放刚生成的节点，对于不同的搜索策略，节点在 OPEN 表中的排序是不同的；CLOSED 表用于存放将要扩展或者已扩展的节点。

2. 设有如图 4.3 所示的与/或搜索树，请用 AO* 算法求出其代价（按边的费用）。

参考答案：

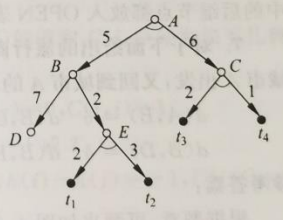

图 4.3 与/或搜索树

$h(A)=h(B)+h(C)=5+6=11$，$h(B)=\min(D,E)=\min(7,2)=2$，$h(C)=2+1=3$，此时 $h(B)$，$h(C)$ 的费用已改变，因此，$h(A)=h(B)+h(C)=2+3=5$。E 不是终端节点，继续扩展 E，得到 $h(E)=2+3=5$，因此，$h(B)=\min(D,E)=\min(7,5)=5$，$h(C)=3$，$h(A)=5+3=8$。这个过程说明：在 AO* 算法求解过程中，如果一个节点没有扩展，那么该节点的评价值按当前评价函数给出其值；如果这个节点扩展之后，这个评价值就失效了，则其评价值要按期后续节点的评价值重新计算。

3. 什么是状态图和与/或图？图搜索与问题求解有什么关系？

参考答案：

状态图是描述寻找目标或路径问题的有向图，即描述一个实体基于事件反应的动态行为，显示了该实体如何根据当前所处的状态在不同的时间做出反应的。与/或图是一种系统地将问题分解为互相独立的小问题，然后分而治之的解决方法。与/或图中有两种代表性的

节点："与节点"和"或节点"，"与节点"指所有的后续节点都有解时它才有解；"或节点"指各个后续节点均完全独立，只要其中有一个有解它就有解。AO* 算法对与/或图表示的问题求解，就是在一个图中寻找一个从初始节点到目标节点的路径问题，图搜索模拟的实际是人脑分析问题、解决问题的过程，它基于领域知识的问题求解过程。

4. 什么是与/或图？什么是可解节点？什么是解图？

参考答案：

与/或图中的弧线表示所连边为"与"关系，不带弧线的边为或关系。这个与/或图中既有与关系又有或关系，因此被称为与/或图。满足下列条件的节点为可解节点：①终止节点是可解节点；②一个与节点可解，当且仅当其子节点全都可解；③一个或节点可解，只要其子节点至少有一个可解。解图实际上是由可解节点形成的一个子图，这个子图的根为初始节点，叶为终止节点，且终止节点一定为可解节点。

5. 什么是评价函数？在评价函数中，$g(x)$和$h(x)$各起什么作用？

参考答案：

评价函数用来估计节点重要性的函数。评价函数 $f(x)$ 被定义为从初始节点 S_0 出发，约束经过节点 x 到达目标节点 S_g 的所有路径中最小路径代价的估计值。它的一般形式为

$$f(x)=g(x)+h(x)$$

其中，$g(x)$为从初始节点 S_0 到节点 x 的实际代价；$h(x)$为从节点 x 到目标节点 S_g 的最优路径的估计代价。

6. 局部择优搜索与全局择优搜索的相同之处与区别各是什么？

参考答案：

局部择优搜索与全局择优搜索的相同之处是都按通用搜索算法的基本步骤进行搜索；其区别是：局部择优搜索扩展节点 N 后仅对 N 的所有子节点按启发函数值大小以升序排序，再进行择优，将选中的放入 OPEN 表中，然后舍弃其他节点；而全局择优搜索扩展节点 N 后要对 N 的所有子节点按启发函数值大小以升序排序，进行择优，将选中的节点和未选中的后继节点都放入 OPEN 表中(不舍弃当前非最优者)。

7. 对于下面给出的旅行商问题，用 d 表示两个城市之间的距离。请用 A* 算法找出从城市 A 出发，又回到城市 A 的最短路径。

$$d(A,B)=6 \quad d(B,C)=6 \quad d(A,E)=7 \quad d(A,C)=1 \quad d(A,D)=5$$
$$d(B,D)=4 \quad d(B,E)=3 \quad d(C,D)=8 \quad d(C,E)=2 \quad d(D,E)=5$$

参考答案：

根据题意，可画出如图 4.4 所示的路线图。

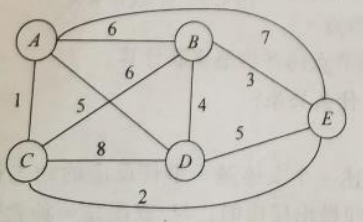

图 4.4 路线图

类似这种类型的问题，由于涉及城市距离的耗费随着深度的增加而增加，因此一般采用搜索树广度优先搜索求解。为此，首先将旅行交通图转换为搜索树，转换方法为：从初始节点 A 开始，把与它直接相邻的节点作为它的后继节点，对其他节点也做同样的扩展，但若一个节点已经作为某个节点的前驱节点，则它就不能再作为该节点的后继节点(避免回路)。另外，

(9) B 向右跳过了两个 W，此时，$h(i)-h(j)=2, C(i,j)=2$。

(10) B 向左跳过了一个 W(可能同时包含一个 B)，此时，$h(i)-h(j)=-1, C(i,j)=1$ 或 2。

(11) B 向左跳过了两个 W，此时，$h(i)-h(j)=-2, C(i,j)=2$。

综上所述，无论是哪一种情况，具有 $h(i)-h(j)\leqslant C(i,j)$，且容易验证 h(目标状态)$=0$，所以该 h 函数是单调的。由于 h 满足单调条件，因此也一定有 $h(n)\leqslant h^*(n)$，即满足 A* 算法的条件。

9. 设数字重写问题的变换规则如下。

6→3,3　6→4,2　4→2,2　4→3,1　3→2,1　2→1,1

问如何用这些规则把 6 变换成一个由若干数字 1 组成的数字序列串？试用 AO* 算法求解，并给出搜索图。设 k-连接符的代价是 k 个单位，$h(x)$启发式函数值规定为 $h(1)=0, h(n)=n(n\neq 1)$。

参考答案：

此题没有完全按照 AO* 算法生成“与/或树”的形式求解，而是一个“与/或图”，不允许生成图中已经存在的节点，图 4.6 是每个循环结束时的搜索树图。

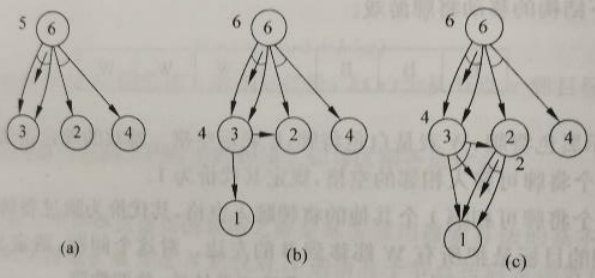

图 4.6　不允许重复生成节点的搜索树图

上面这种做法比较简单，也可以采用生成“与/或树”的形式，也是教材上 AO* 算法生成的“与/或树”，其步骤如图 4.7 所示的搜索树图。

10. 用恰当的表达方法描述 Hanoi 塔问题。在 A 柱上串有若干金碟，小金碟在大金碟上面，现在要求将 A 柱上的金碟全部移到 B 柱上，移动时遵循以下规则。

(1) 一次只能移动一个金碟。

(2) 不能将大金碟放在小金碟之上。

(3) 可以利用 C 柱作为中转地。

参考答案：

(1) 题目描述可转换为如下问题(N 阶汉诺塔问题)

有编号为 A、B、C 的 3 个柱子和标识为 1、2、…、N 依次表示从小到大的 N 个有中心孔的金碟；柱子可以从中心孔穿过，初始状态下 N 个金碟按 1、2、…、N 顺序堆放在 A 号柱子上。目标状态为 N 个金碟以同样次序顺序堆放在 B 号柱子上，金碟的搬移需遵守的规则是：每次只能搬一个金碟，且较大金碟不能压放在较小金碟之上，可以借助于 C 柱作为中转地。

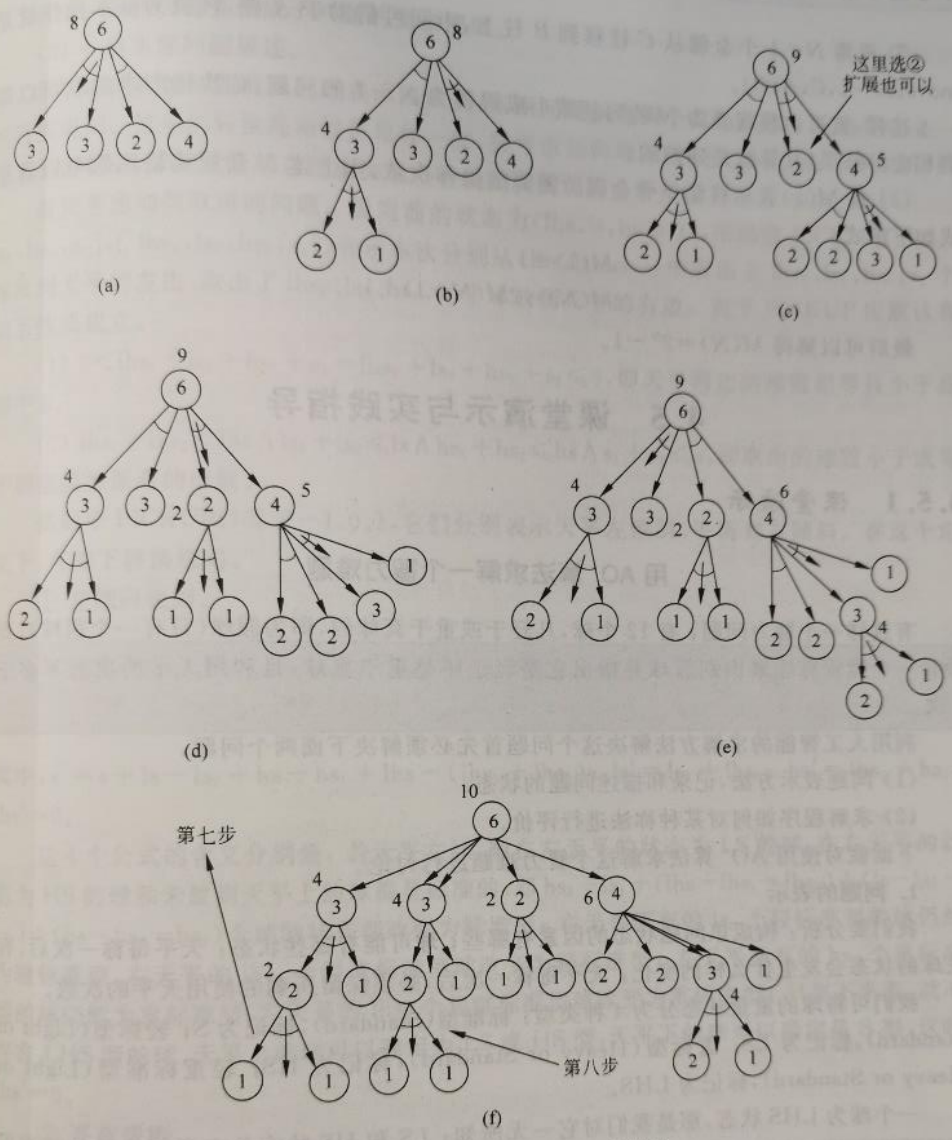

图 4.7　允许重复生成节点的搜索树图

(2) 假设基本操作为 move(x,A,C,B)，表示将 x 个金碟从 A 移到 B 上，中间可借助于 C。当 $N=1$ 时，则无须借助中间的 C 柱，就可以直接实现将 1 个金碟从 A 移到 B 上，这也是问题的最简操作，可表示为 move-one(1,A,B)；当 $N>1$ 时，需要用中间的 C 柱做辅助。其操作又可分解为以下三步。

① 将上部的 $N-1$ 个金碟从 A 柱移到 C 柱上，中间可借助于 B 柱，完成这个任务的基本操作就是 move($N-1$,A,B,C)。

② 将 1 个金碟直接从 A 移到 B 上，转换为基本操作就是 move-one(1,A,B)。

图 4.4 中节点除了初始节点 A 之外，其他的节点都有可能在搜索树中多次出现，但它们都是图中的同一个节点。这就是本书对应的教材上提到的，搜索树图可以以树的形式表示，也可以以图的形式表示，如果用树表示，则没有回路，但可能存在重复节点。

用 A* 算法求解，设评价函数 $f(n)=d(n)+h(n)$，其中，$d(n)$ 为状态的深度；$h(n)$ 为城市间的距离。其搜索树如图 4.5 所示。

其中，以 $f(B)=d(B)+h(B)=9$ 为例，$d(n)=2$（深度为 2），$h(n)=1+6$。

用 A* 算法求解，还可以定义 $h_1=n*k$，其中，n 为还未走过的城市数；k 为还未走过的城市间距离的最小值。$h_2=\sum_{1}^{n}k_i$，其中，n 为还未走过的城市数；k_i 为还未走过的城市间距离中 n 个最小的距离。显然，这两个 h 函数均满足 A* 算法的条件。

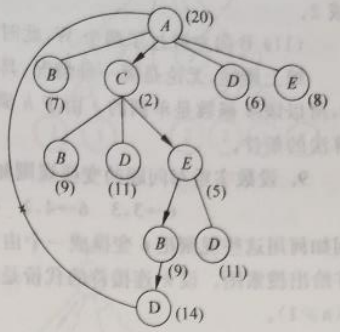

图 4.5 搜索树

8. 设有如下结构的移动将牌游戏。

B	B	B		W	W	W

其中，B 表示黑色将牌，W 表是白色将牌，E 表示空格。游戏的规定走法如下。

(1) 任意一个将牌可移入相邻的空格，规定其代价为 1。

(2) 任何一个将牌可相隔 1 个其他的将牌跳入空格，其代价为跳过将牌的数目加 1。

游戏要达到的目标是把所有 W 都移到 B 的左边。对这个问题，请定义一个启发函数 $h(n)$，并判别这个启发函数表示的策略是否满足可采纳性、单调性等。

参考答案：

可定义 h 为：h=B 右边的 W 的数目。

设节点 j 是节点 i 的子节点，则根据走法不同，$h(i)-h(j)$ 的值和 $C(i,j)$ 分为如下几种情况。

(1) B 或 W 走到了相邻的一个空格位置，此时，$h(i)-h(j)=0$，$C(i,j)=1$。

(2) W 跳过了 1 或 2 个 W，此时，$h(i)-h(j)=0$，$C(i,j)=1$ 或 2。

(3) W 向右跳过了一个 B（可能同时包含一个 W），此时，$h(i)-h(j)=-1$，$C(i,j)=1$ 或 2。

(4) W 向右跳过了两个 B，此时，$h(i)-h(j)=-2$，$C(i,j)=2$。

(5) W 向左跳过了一个 B（可能同时包含一个 W），此时，$h(i)-h(j)=1$，$C(i,j)=1$ 或 2。

(6) W 向左跳过了两个 B，此时，$h(i)-h(j)=2$，$C(i,j)=2$。

(7) B 跳过了 1 或 2 个 B，此时，$h(i)-h(j)=0$，$C(i,j)=1$ 或 2。

(8) B 向右跳过了一个 W（可能同时包含一个 B），此时，$h(i)-h(j)=1$，$C(i,j)=1$ 或 2。

③ 再将 $N-1$ 个金碟从 C 柱移到 B 柱上，中间可借助于 A 柱，转换为基本操作就是 move($N-1$,C,A,B)。

这样，就可以将规模为 N 的问题减小成规模为 $N-1$ 的问题，依次递归求解就可以得到相应的结果，其算法是递归的。

(3) 设 $M(x)$ 表示移动 x 个金碟所需要的操作次数，则上述 N 阶汉诺塔问题可以表示成如下形式。

$$M(1)=1$$
$$M(N)=2M(N-1)+1$$

最后可以解得 $M(N)=2^N-1$。

4.5 课堂演示与实践指导

4.5.1 课堂演示

用 AO* 算法求解一个智力难题

有这样一个智力问题：有 12 个球，凡轻于或重于真球的，即为假球(只有一个假球)，要设计一个搜索算法来识别假球并指出它是轻于还是重于真球，且利用天平的次数不多于 3 次。

利用人工智能的求解方法解决这个问题首先必须解决下面两个问题。

(1) 问题表示方法，记录和描述问题的状态。

(2) 求解程序如何对某种称法进行评价。

下面就对使用 AO* 算法求解这个智力难题进行讨论。

1. 问题的表示

我们要分析：构成该问题状态的因素有哪些；球可能有哪些状态；天平每称一次后，有关球的状态会发生什么样的变化；天平每称一次后，必须保留所剩的使用天平的次数。

我们可将球的重量状态分为 4 种类型：标准型(Standard)，标记为 S；轻标型(Light or Standard)，标记为 LS；重标型(Heavy or Standard)，标记为 HS；轻重标准型(Light or Heavy or Standard)，标记为 LHS。

一个球为 LHS 状态，那是我们对它一无所知；LS 和 HS 状态是有可能为轻的或有可能为重的，当然也可能是标准的；S 状态是已知为标准的。

综上所述，问题的状态空间可表示成一个五元组，即(lhs,ls,hs,s,t)。其中，前 4 个元素表示当前这 4 种类型球的个数，t 表示所剩称球的次数。在这样的状态空间表示下，有

初始状态：(12,0,0,0,3)

目标状态：Sg_1：(0,1,0,11,0)和 Sg_2：(0,0,1,11,0)

Sg_1、Sg_2 分别表示最后找到一个轻的或找到一个重的球，其余 11 个为标准球。

2. 利用 AO* 算法求解

利用 AO* 算法求解问题需要找出如下要素。

(1) 初始问题的描述。