**3.6.**

**a.** 可以使用图论的方法来形式化这个问题。将地图的每个地区看作图的一个节点，相邻的地区之间用边连接。然后，我们可以使用回溯法或深度优先搜索来遍历每个节点，并在遍历的过程中为每个节点着色。在着色的过程中，需要检查相邻节点的颜色，确保相邻地区的颜色不相同，如果颜色冲突，则回溯到上一个节点，尝试其他颜色，直到找到合适的着色方案为止。最终得到的着色方案满足题目要求。

**b.** 可以使用算法来形式化这个问题，设立一个状态空间，每个状态包含猴子的位置、箱子的位置和香蕉的位置。然后我们可以使用搜索算法(如广度优先搜索)来在状态空间中寻找猴子吃香蕉的路径。每一步的移动规则是：猴子可以在地上移动，也可以爬上箱子。箱子可以在地上移动，也可以叠放在另一个箱子上。猴子吃到香蕉的目标状态是香蕉和猴子在同一位置且没有箱子。通过搜索算法，可以找到猴子吃香蕉的最短路径或可行路径。

**c.** 对于这个问题，我们可以使用二分法或其他搜索算法来找出导致程序输出“不合法的输入记录”的具体记录。首先，将文件中的记录分成两半，分别对两半进行处理。如果第一半中出现了“不合法的输入记录”，则继续在第一半中继续二分搜索，否则，在第二半中进行搜索。以此类推，直到找到导致输出的具体记录为止。

**d.** 可以使用搜索算法来形式化这个问题。将每个状态定义为三个水壶中的水量组合。然后，我们可以使用广度优先搜索或深度优先搜索等算法来在状态空间中搜索，直到找到一个状态，其中任意一个水壶的水量为1加仑，即可达到量出恰好1加仑水的目标。在搜索的过程中，需要对每个状态执行相应的操作，即装满水、倒空水或从一个壶向另一个壶倒水。最终得到的操作序列即为量出1加仑水的方法。

**3.9.**

**a.**

**状态：** 两岸传教士和野人的数量（传教士在两岸的数量都不少于野人数量）以及船的位置（在船的哪一侧）

**初始状态：** 三个传教士和三个野人和船都在河的一侧

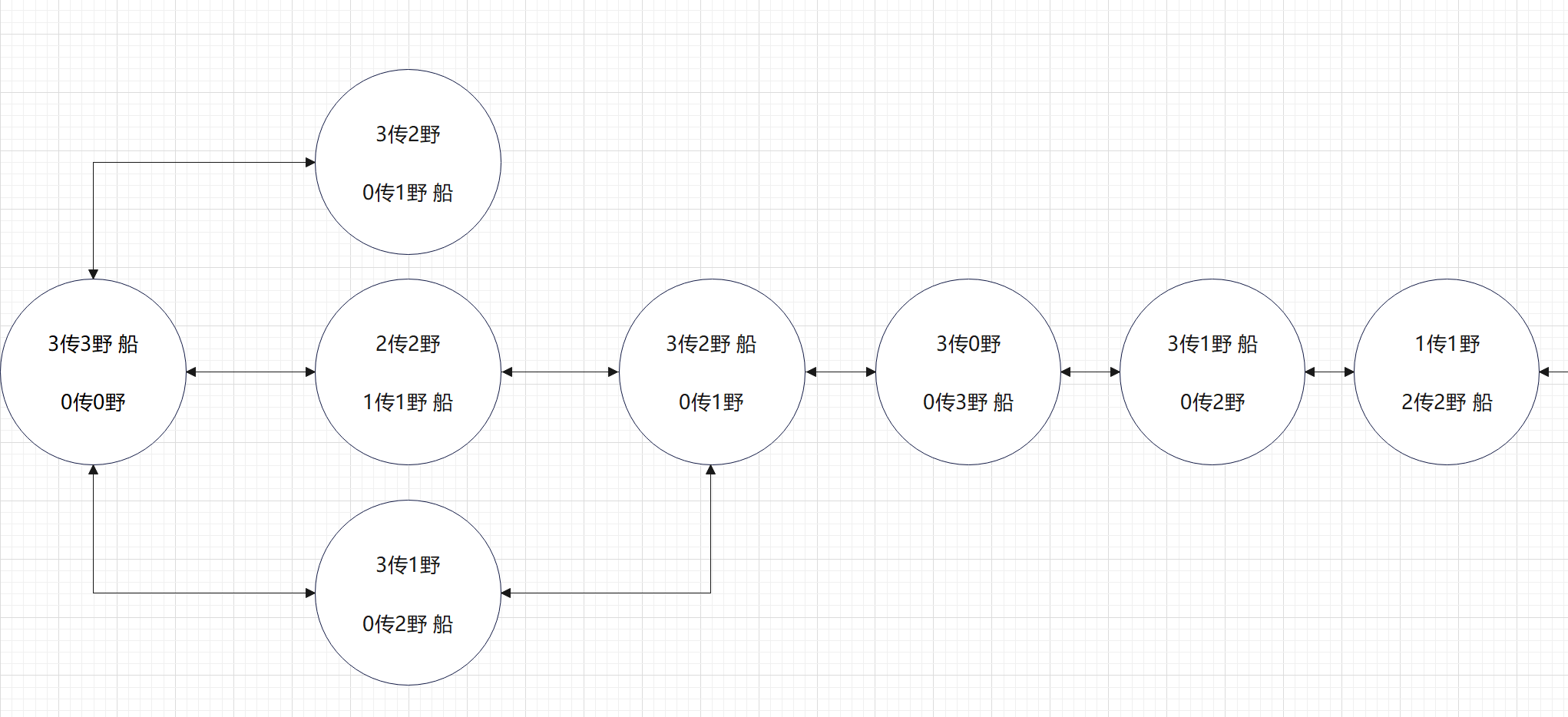
**行动：**  利用船将传教士河野人在两岸间运输，保证在船上，两岸的的传教士都不少于野人的数量

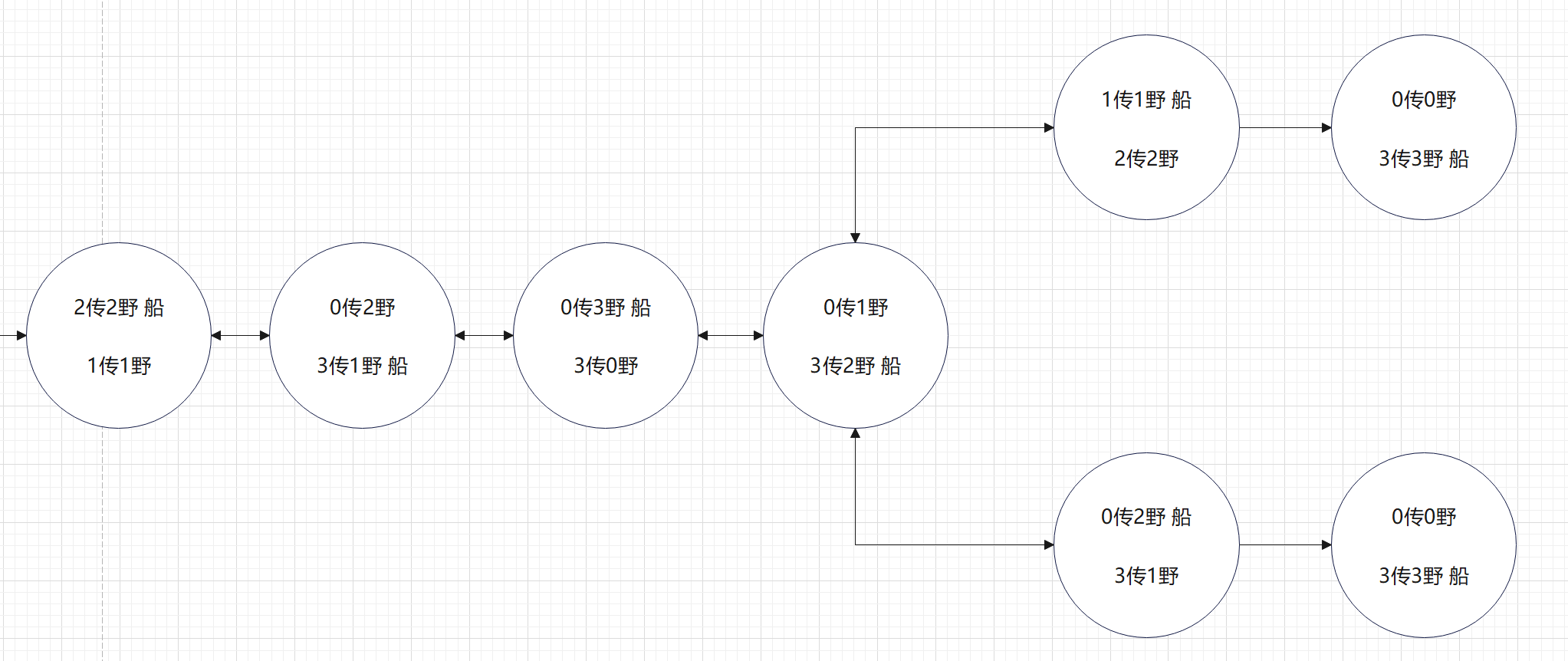
**转移模型：** 船运输后的两岸的传教士和野人的数量和船的位置作为当前状态

**目标测试：** 传教士和野人是否都到达岸的另一侧

**路径耗散：** 船每摆渡一次耗散为1

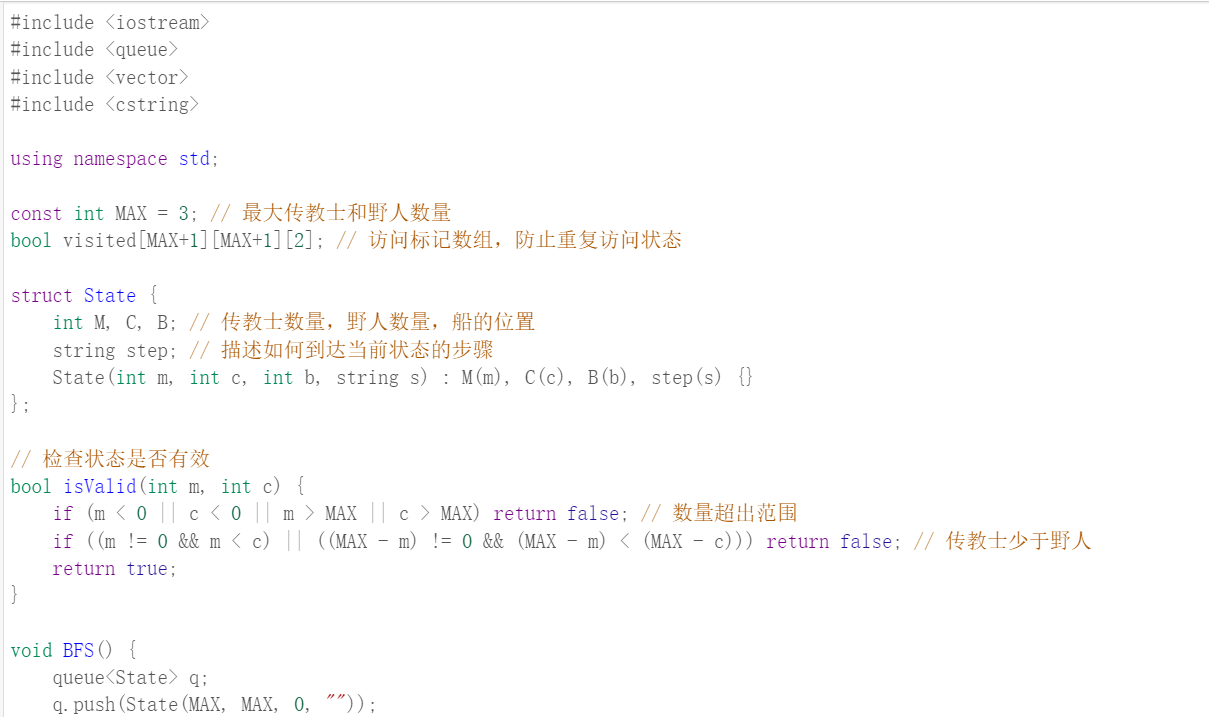
**状态空间图：**



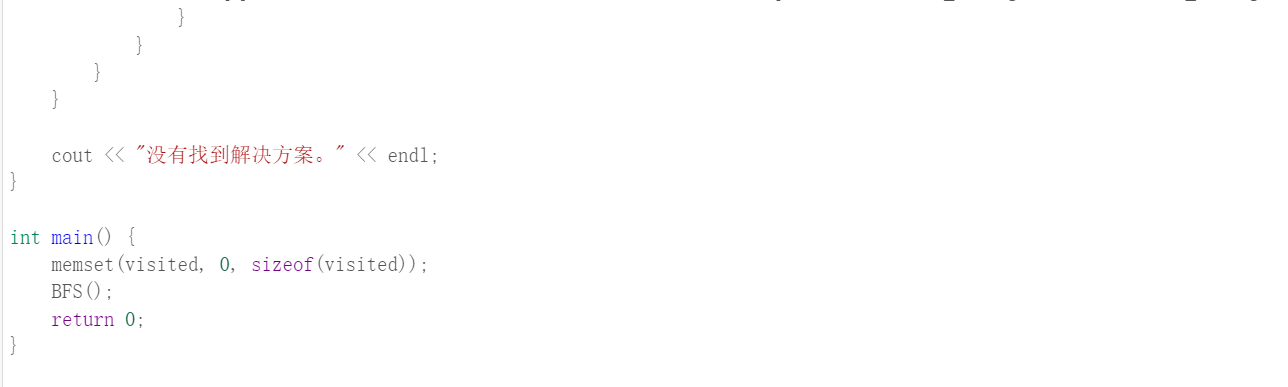


**b.**

因为要求求出最优解，所以我采用宽度优先搜索，代码如下图：







对于这个问题是要检查重复状态的，因为相邻两个合法状态可以无止境的来回转换，所以为了规避这种情况，需要检查重复状态

**c.**

①状态转化的条件比较特殊

②在不排除非法情况时状态很多

③需要一个合适的搜索算法

**3.21**

**a.** 错误。宽度优先搜索是一种一致代价搜索，但不一定是一致代价搜索的特殊情况。一致代价搜索是指扩展代价最低的节点，而宽度优先搜索是按照扩展的节点深度进行的，可能存在代价较高的情况。

**b.** 错误。深度优先搜索不一定是最佳优先搜索的特殊情况，最佳优先搜索是基于启发式函数的，会选择估计代价最低的节点进行扩展，而深度优先搜索是不考虑代价的，只是按深度扩展。

**c.** 正确。一致代价搜索是A搜索的一种特殊情况。A搜索是一种启发式搜索算法，结合了代价和启发式函数的估计，选择代价+启发式函数估计最低的节点进行扩展。一致代价搜索即为A\*搜索中启发式函数等于零的情况，也就是只考虑代价的情况。

**3.25**

当w取1和0时是最优的；

w=0：一致代价搜索

w=1：A\*搜索

w=2：贪婪最佳优先搜索