**传教士与野人问题**

1. 问题描述：三个传教士和三个野人需要渡过一条河流，只有一艘小船每次最多容纳两个人，在任何地方传教士的数量都必须大于野人的数量，找到一种方法使得所有人都能够安全渡河。

状态表示：我们用一个5元组表示时时刻刻的状态(m\_left,c\_left,pos,m\_right,c\_right),其中m\_left表示左岸传教士的数量，c\_left表示左岸野人数量，pos表示船的位置（1代表左岸，0代表右岸），m\_right代表右岸传教士的数量，c\_right代表右岸野人数量。

初始状态：（3,3,1,0,0）,目标状态:（0,0,0,3,3）;

每次可执行操作：

(1) （m\_left-1,c\_left,1-pos,m\_right+1,c\_right）；

(2) （m\_left-2,c\_left,1-pos,m\_right+2,c\_right）；

(3) （m\_left-1,c\_left-1,1-pos,m\_right+1,c\_right+1）；

(4) （m\_left,c\_left-1,1-pos,m\_right,c\_right+1）；

(5) （m\_left,c\_left-2,1-pos,m\_right,c\_right+2）；

(6) （m\_left+1,c\_left,1-pos,m\_right-1,c\_right）；

(7) （m\_left+2,c\_left,1-pos,m\_right-2,c\_right）；

(8) （m\_left+1,c\_left+1,1-pos,m\_right-1,c\_right-1）；

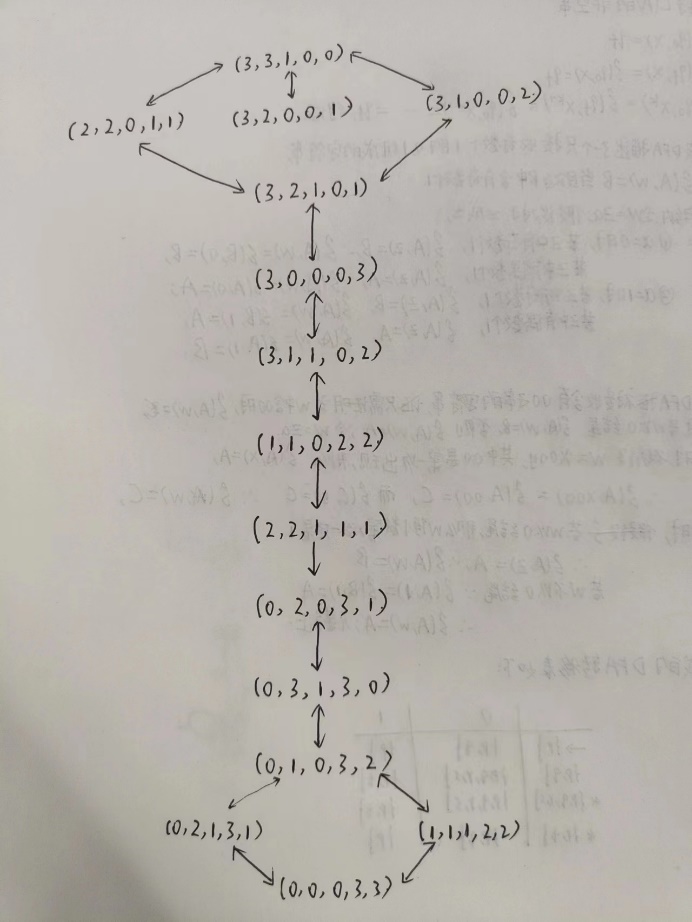
(9) （m\_left,c\_left+1,1-pos,m\_right,c\_right-1）；

(10) （m\_left,c\_left+2,1-pos,m\_right,c\_right-2）；

同时要注意每次状态转移都要满足：

1. m\_left>=c\_left
2. m\_right>=c\_right

分析之后我们得出该问题的完全状态空间转换为：



(b)对于这个问题我们可以采用深度优先搜索、广度优先搜索、A\*搜索等搜索算法，注意检查重复状态以及不合法状态。

(c)虽然状态空间比较简单，但是检测重复状态是一个困难，另外，在当前状态选取下一个合法状态要求不重不漏的枚举所有合法状态也存在困难，当一个状态无合法状态时我们需要回溯求解，这些都使得认为求解变的复杂。

本题中我采用了BFS搜索的搜索策略：

搜索序列：

(3,3,1,0,0)->(2,2,0,1,1)->(3,2,0,0,1)->(3,1,0,0,2)->(3,2,1,0,1)->(3,0,0,0,3)->(3,1,1,0,2)->(1,1,0,2,2)->(2,2,1,1,1)->(0,2,0,3,1)->(0,3,1,3,0)->(0,1,0,3,2)->(0,2,1,3,1)->(1,1,1,2,2)->(0,0,0,3,3)

解序列：

(3,3,1,0,0)->(2,2,0,1,1)->(3,2,1,0,1)->(3,0,0,0,3)->(3,1,1,0,2)->(1,1,0,2,2)->(2,2,1,1,1)->(0,2,0,3,1)->(0,3,1,3,0)->(0,1,0,3,2)->(0,2,1,3,1)->(0,0,0,3,3)