Exercise 5.2

a. • **初始状态：**两个任意的 8-puzzle 状态

• **后继函数：**在未解决的 8-puzzle 中随机选择一个进行空格的移动，包括 Left、Right、Up 或 Down 动作，但如果空格位于边缘或角落，则不是所有动作可用。

• **目标测试：**两个 8-puzzle 的滑块是否都到达目标位置

• **动作代价：**每次移动滑块代价都为 1。

b. 单个 8-puzzle 的可达状态数为 $\frac{9!}{2}$=181,440

双个 8-puzzle 的可达状态数为 $(\frac{9!}{2})^2$=181440^2 = 32,921,779,200

c. 可以使用**期望极小化极大算法**

d. 这取决于初始状态的情况：

如果初始时两个 8-puzzle 都只差一步解决，那么优先抛硬币的玩家获胜。

如果初始时只有一个 8-puzzle 差一步解决，那么获胜者将是抛硬币最先抛中该 puzzle 的玩家。

如果初始时两个 8-puzzle 都比较复杂（需要大于一步解决）时，假设某一时刻玩家 1 抛中了一个只剩两步解决的 puzzle，如果他选择靠近目标一步，那么对手获胜的概率将是 1/2+1/4+1/16……=2/3，自己获胜概率为 1/3，所以最佳行动应该为远离目标一步，按这样考虑，表现完美的玩家之间不可能分出胜负。

Exercise 5.7

证明：考虑一个 MAX 节点，其有 n 个子 MIN 节点，这 n 个 MIN 节点的子节点均为叶节点。

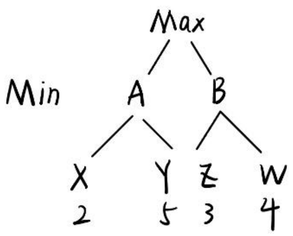
设第 i 个 MIN 节点的叶节点中最低效用为 C_i*，而第 i 个 MIN 节点选择了一个次低效用 C_i，

则有 $C_i >= C_i^*$。

设 MAX 节点基于最优 MIN 节点最优策略的效用值为 $u_{opt}=max(C_1^*,......,C_n^*)$，基于最优 MIN 节点最优策略的效用值为 $u_{sub}=max(C_1,......,C_n)$。

由于 $C_i >= C_i^*$，对 i=1,......,n 均成立，所以必定有 $u_{sub} >= u_{opt}$ 。**得证。**

构造:



如图，规定 MAX 和 MIN 需要同时做出行动，在双方都行动之前不知道对方会如何行动。

由于 MIN 层采取次优策略，A 节点会选择 X=5，B 节点会选择 Z=4。

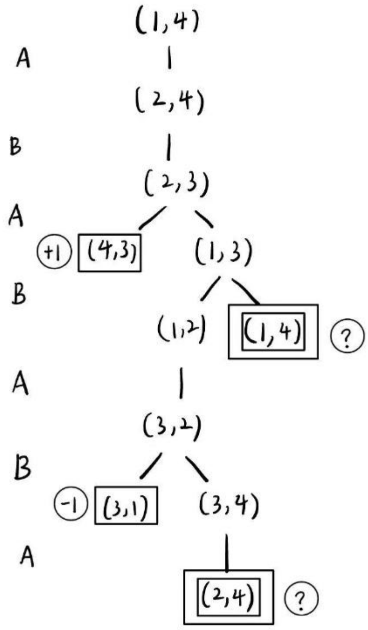
若 MAX 采取最优策略, MAX 节点认为 MIN 采取最优策略, 即 A 选择 X=2, B 选择 Z=3，那么 MAX 会选择 B。然而实际上 MAX 的选择得到的效用为 4。

若 MAX 采取次优策略，在同上猜测的情况下，MAX 会选择 A。实际上 MAX 的选择得到的效用为 5。

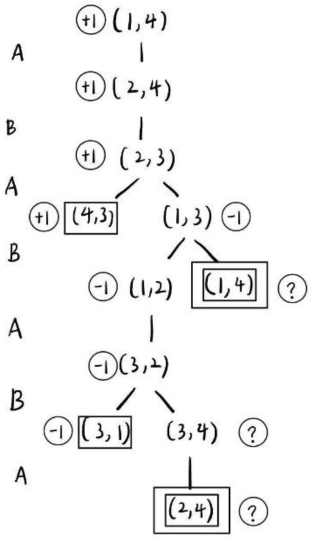
可见 MIN 采取次优策略时，MAX 采取次优策略的收益可能高于采取最优策略的收益。

Exercise 5.8

a.



b. 如果一个节点的后继值均为“？”，那么它的值也是“？”；如果不全为“？”，根据当前的行动者选择获胜值，即 A 选择+1，B 选择 -1。



c．标准的极小极大算法是深度优先的，在该问题中会陷入无限循环。

修复方法：记录所有走过的状态，每次生成一个后继状态时，与记录的状态一一比较，若重复，则标记为“？”并停止生成该重复节点的后继。对“？”值的处理同 b.。d

d. 证明（归纳法）：

n=3 时，显然先行动的 A 会输掉。

n=4 时，按照上面画出的博弈树，A 有必胜策略。

假设 n<=2k-1 时所有 n 为偶数的游戏，A 会获胜；所有 n 为奇数的游戏，B 会获胜。

n=2k 时，考虑到 A、B 双方的第一步都只能互相靠近，第一次行动后必有 A 在方块 2，B 在方块 n-1。考虑到子游戏[2, 2k-1]中，A 是必胜的，即 A 会先于 B 到达 2 之前到达 n-1，因此也会先于 B 到达 n，那么 A 不会选择回到 1，否则 B 到 1 的距离会小于 A 到 n 的距离而导致 B 必胜。如果 B 选择回到 n，同理 A 也是必胜的。

n=2k+1 时，同上分析，但在子游戏 [2, 2k]中 B 是必胜的，所以这种情况下 B 必胜。

综上，由数学归纳法，命题得证。

Exercise 5.13

1.双人非零和博弈、独立效用函数的算法调整：

设两个玩家的效用函数分别为 U1(s) 和 U2(s)

(1)极小化极大算法的调整

节点数据：每个节点需要同时记录两个玩家的效用值 (U1,U2)

节点行为：MAX 节点（玩家 1）：选择最大化 U1(s)的行动

MIN 节点（玩家 2）：选择最大化 U2(s)的行动

策略调整：两玩家不再是单纯的对抗关系，可能还有合作关系，需考虑对方的效用函数对自身决策的影响

(2) α-β 剪枝算法的调整

剪枝条件：　**Max 节点**(玩家 1)：仅当某个子节点的 U1 值高于当前已知的最高 U1 值时，保留该分支；否则剪枝。

Min 节点（玩家 2）：仅当某个子节点的 U2 值高于当前已知的最高 U2 值时，保留该分支；否则剪枝。

2. 无约束终端效用时的剪枝可能性：

如果两个玩家的终端效用函数无任何约束（即 U1 和 U2 完全独立），传统的 α-β 剪枝可能无法生效，因为无法通过单一效用值的上下界直接推断其他玩家的选择。

3. 近似零和博弈约束下的的剪枝可能性

改变约束条件：$U1 \in [-k-U2, k-U2]$

剪枝规则调整：**Max 节点**（玩家 1）：更新 **α**=*当前已知的* ***U1 下限***，通过 **U1≥α** 剪枝。

Min 节点（玩家 2）：更新 **β**=*当前已知的* ***U1 上限***，通过 **U1≤β** 剪枝。