HW4

5.9

5.9 本题以井字棋（圈与十字游戏）为例练习博弈中的基本概念。定义 X_n 为恰好有 n 个 X 而没有 O 的行、列或者对角线的数目。同样 O_n 为正好有 n 个 O 的行、列或者对角线的数目。效用函数给 $X_3 = 1$ 的棋局 $+1$ ，给 $O_3 = 1$ 的棋局 -1 。所有其他终止状态效用值为 0 。对于非终止状态，使用线性的评估函数定义为 $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$ 。

a. 估算可能的井字棋局数。

b. 考虑对称性，给出从空棋盘开始的深度为2的完整博弈树(即，在棋盘上一个 X 一个 O 的棋局)。

c. 标出深度为2的棋局的评估函数值。

d. 使用极小极大算法标出深度为1和0的棋局的倒推值,并根据这些值选出最佳的起始行棋。

e. 假设结点按对 $\alpha - \beta$ 剪枝的最优顺序生成，圈出使用 $\alpha - \beta$ 剪枝将被剪掉的深度为2的结点。

a

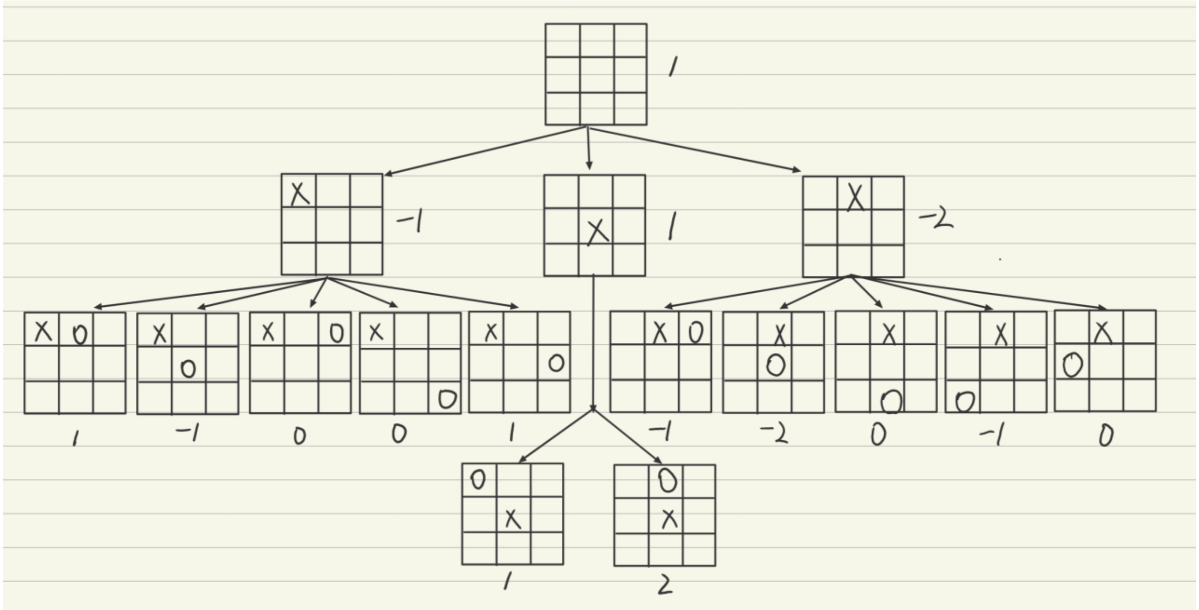
最少是先手三轮获胜，为9!/4!局

最多是9个格子全部下满，为9！局

所以在9!/4!到9！局之间

b、c、d

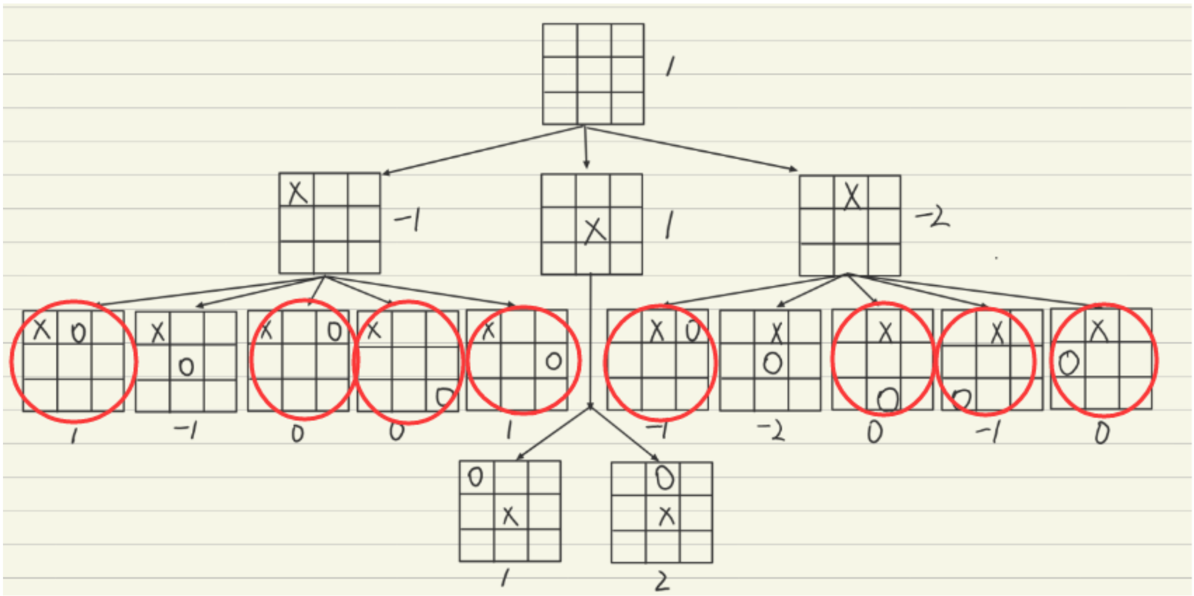
如图



最佳起始行棋是在棋盘中间

e

如图



5.8

5.8 考虑**图5.17**中描述的两人游戏。

a. 根据如下约定画出完整博弈树:

- 每个状态用 (s_A, s_B) 表示，其中 s_A 和 s_B 表示棋子的位置。
- 每个终止状态用方框画出，用圆圈写出它的博弈值。
- 把循环状态（在到根结点的路径上已经出现过的状态）画上双层方框。由于不清楚他们的值，在圆圈里标记一个“? ”。

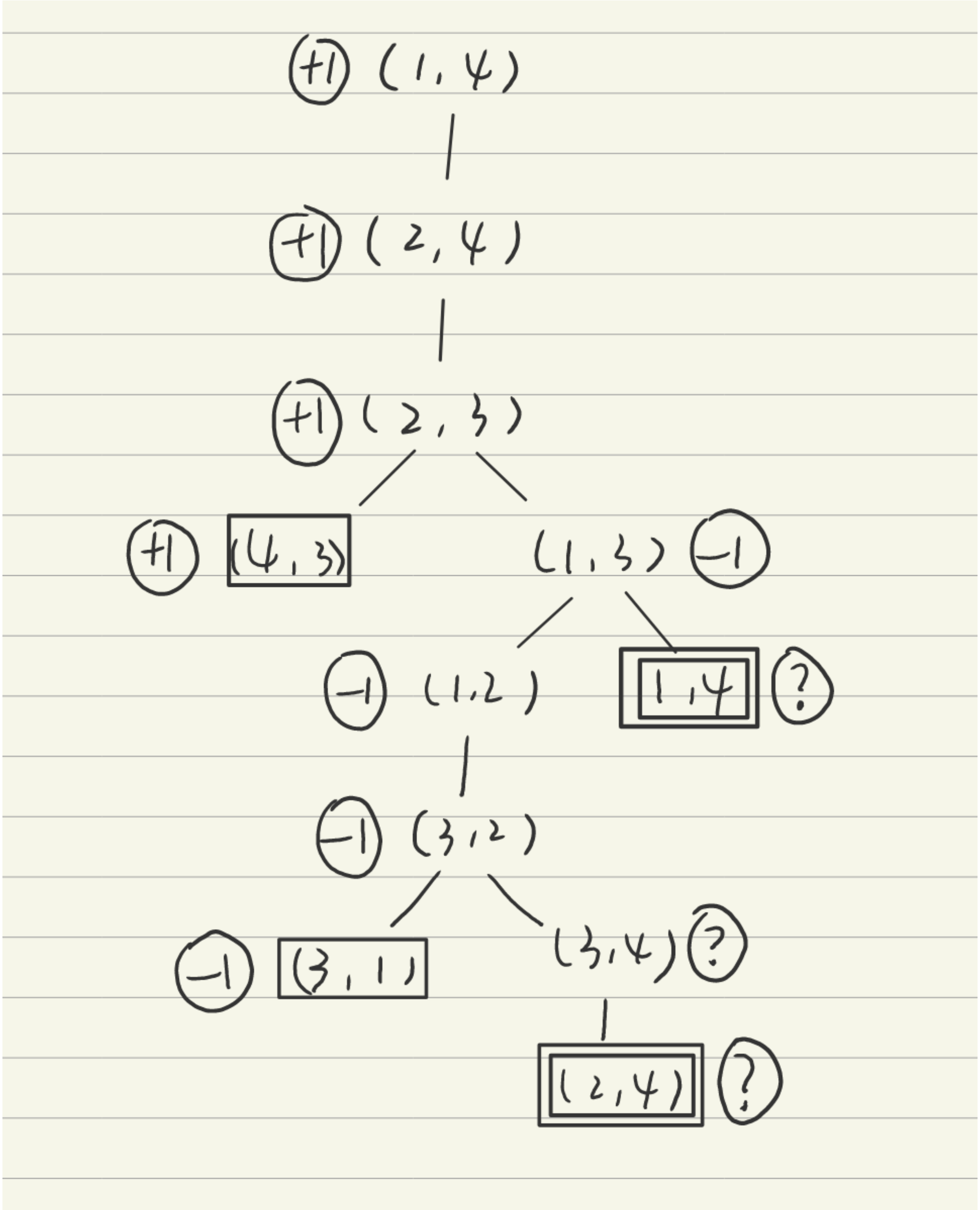
b. 给出每个结点倒推的极小极大值（也标记在圆圈里）。解释怎样处理“?”值和为什么这么处理。

c. 解释标准的极小极大算法为什么在这棵博弈树中会失败，简要说明你将如何修正它，在（**b**）的图上画出你的答案。你修正后的算法对于所有包含循环的游戏都能给出最优决策吗?

d. 这个4-方格游戏可以推广到 n 个方格，其中 $n > 2$ 。证明如果 n 是偶数 A 一定能赢，而 n 是奇数则 A 一定会输。

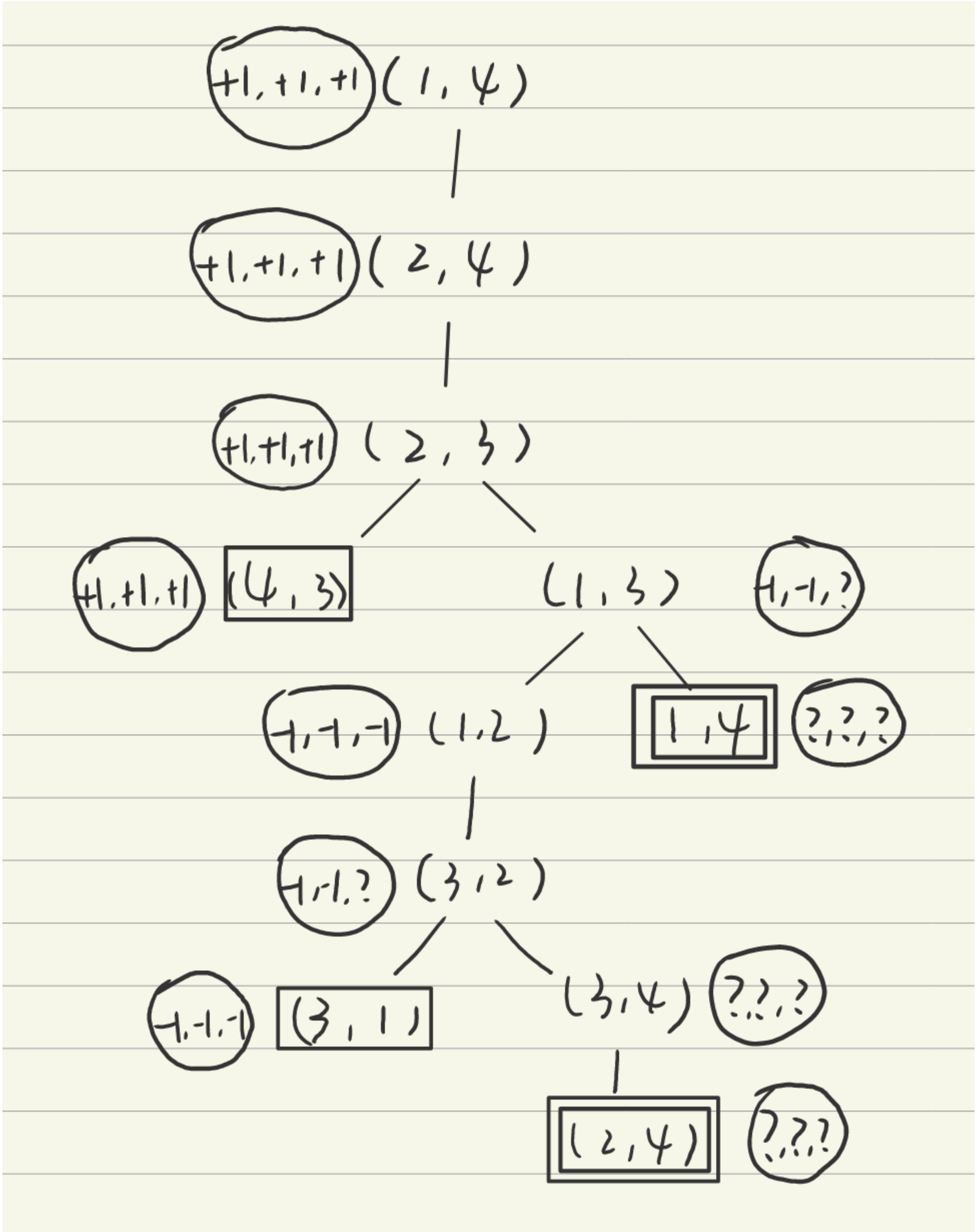
Figure 5.17 The starting position of a simple game. Player A moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game to A is $+1$; if player B reaches space 1 first, then the value of the game to A is -1.

a



b

在圆圈中加上极小极大值（在博弈值后）：

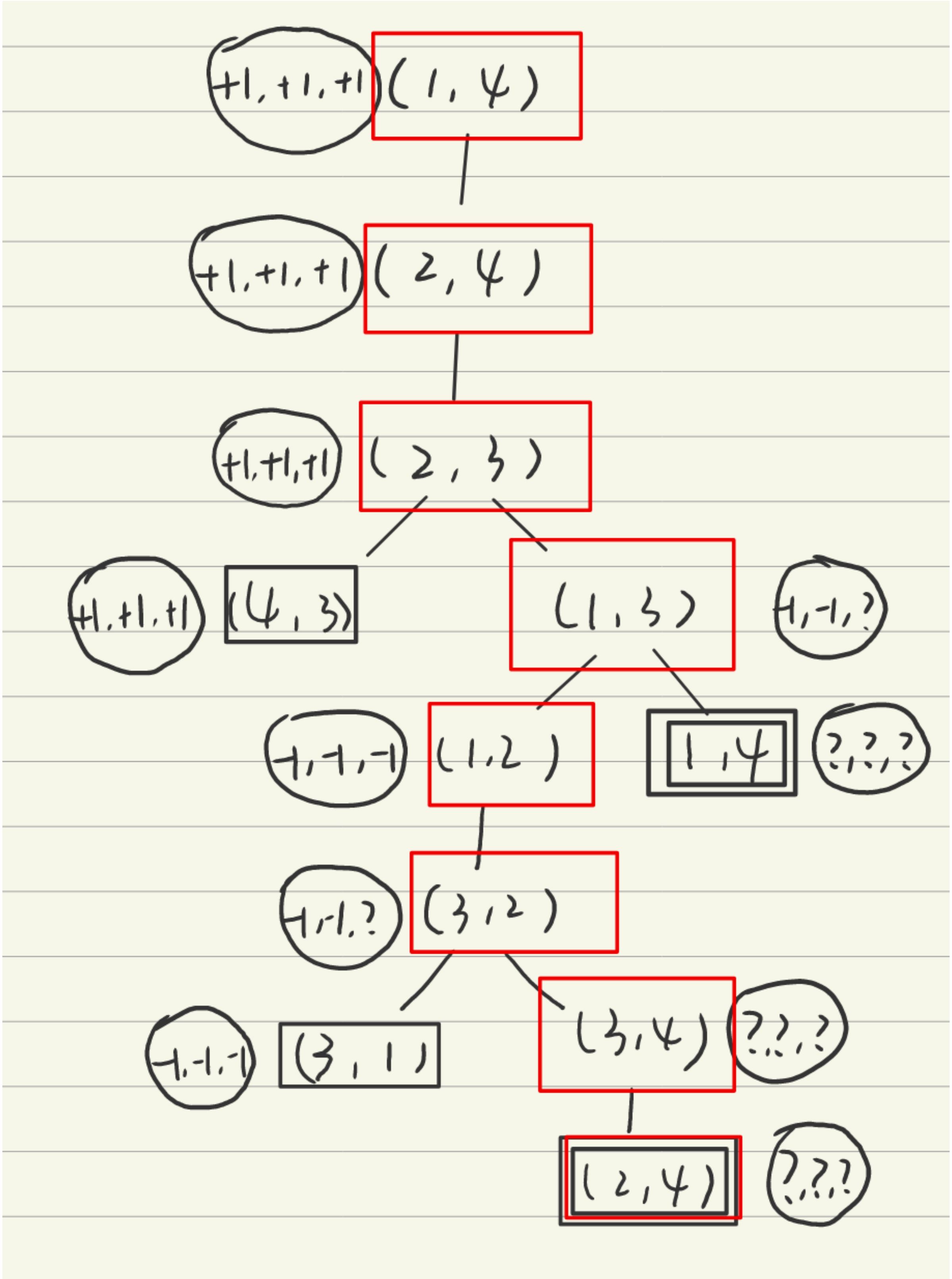


对？的处理方法是视为-1到+1之间的值，因为A如果能够选择则会选择+1，除非后继都是？

c

标准的极小极大算法是深度优先的，可能会陷入循环，应该通过检测重复状态进行修正，遇到重复状态直接返回？

修正后答案如图所示：



修正后的算法对于所有包含循环的游戏不一定能给出最优决策

因为并不能区分不同的？值，当获胜的情况不唯一时就有可能丢失其他获胜状态

d

首先考虑n=3时，A输，n=4时，A赢

因此对于大于4的n，如果A在n-2能赢，则开始时A和B各自向前行棋一步，此时只要A不向原位置移动，则两者就处于n-2的状态，则A在n时能赢；同理，A在n-2时输，则n时也输

综上：A在n为奇数时输，在n为偶数时赢

5.13

5.13 请给出 $\alpha-\beta$ 剪枝正确性的形式化证明。要做到这一点需考虑**图5.18**。问题为是否要剪掉结点 n_j ,它是一个**MAX**结点，是 n_1 的一个后代。基本的思路是当且仅当 n_1 的极小极大值可以被证明独立于 n_j 的值时，会发生剪枝。

a. n_1 的值是所有后代结点的最小值: $n_1 = min(n_2, n_{21}, \ldots, n_2b_2)$ 请为 n_2 找到类似的表达式，以得到用 n_j 表示的 n_1 的表达式。

b. 深度为 i 的结点 n_i 的极小极大值已知， l_i 是在结点 n_i 左侧结点的极小值（或者极大值)。同样， r_i 是在 n_i 右侧的未探索过的结点的极小值(或者极大值)。用 l_i 和 r_i 的值重写 n_1 的表达式。

c. 现在重新形式化表达式，来说明为了向 n_1 施加影响， n_j 不能超出由 l_i 值得到的某特定界限。

d. 假设 n_j 是**MIN**结点的情况，请重复上面的过程。

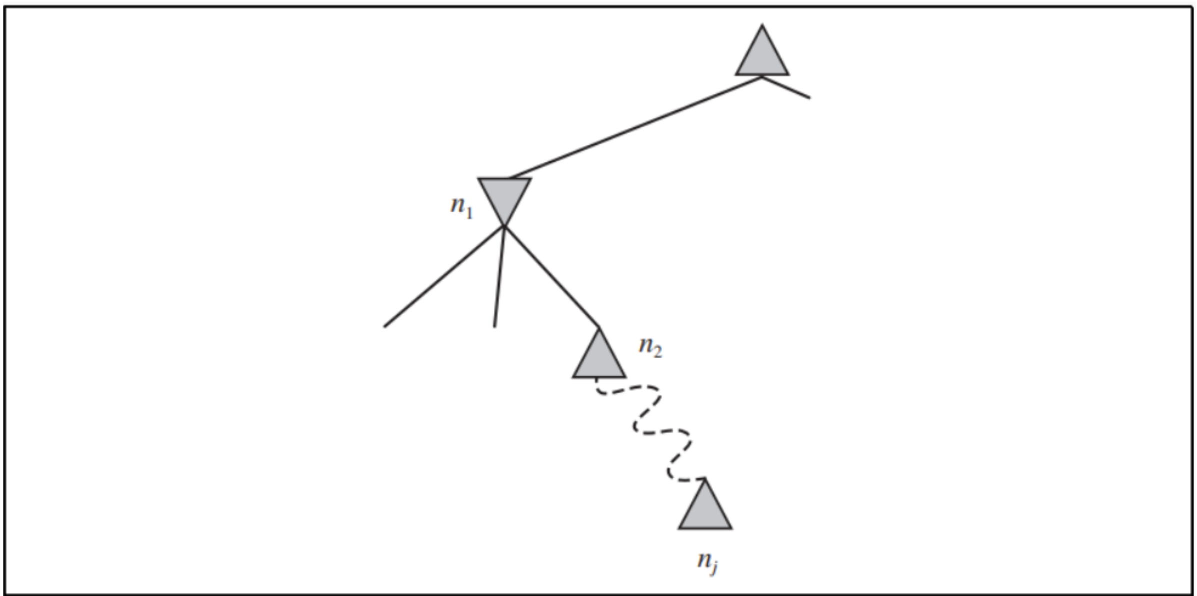


Figure 5.18 Situation when considering whether to prune node n_j.

a

$$n_2 = \max(n_3, n_{31}, \cdots, n_{3b_3})$$

依次往下类推，得到

$$n_1 = \min(\max(\cdots\max(n_j, n_{j1}, \cdots, n_{jb_j})), n_{21}, \cdots, n_{2b_2})$$

b

$$\begin{aligned} n_1 &= \min(l_2,\ n_2,\ r_2) \\ &= \min(l_2,\ \max(l_3,\ n_3,\ r_3),\ r_2) \\ &\cdots \\ &= \min(l_2,\ \max(l_3,\ \max(\cdots\max(l_j,\ n_j,\ r_j),\ \cdots),\ r_3),\ r_2) \end{aligned}$$

c

$n_i = \min(l_{i+1}, \max(l_{i+2}, \max(\cdots \max(l_j, n_j, r_j), \cdots), r_{i+2}), r_{i+1})$

为了向n_1施加影响，n_j，也就是上式中的max结点，需要不超过l_j，从n_1开始，所以n_j应该小于l_2、$l_4 \ldots l_j$，即不超过$min(l_2, l_4, \ldots, l_j)$

d

交换上述的min和max即可，也就是可以得到

$n_i = \max(l_{i+1}, \min(l_{i+2}, \min(\cdots \min(l_j, n_j, r_j), \cdots), r_{i+2}), r_{i+1})$

从而同c得到：

n_j不小于$max(l_3, l_5, \ldots, l_j)$