人工智能基础第四次作业

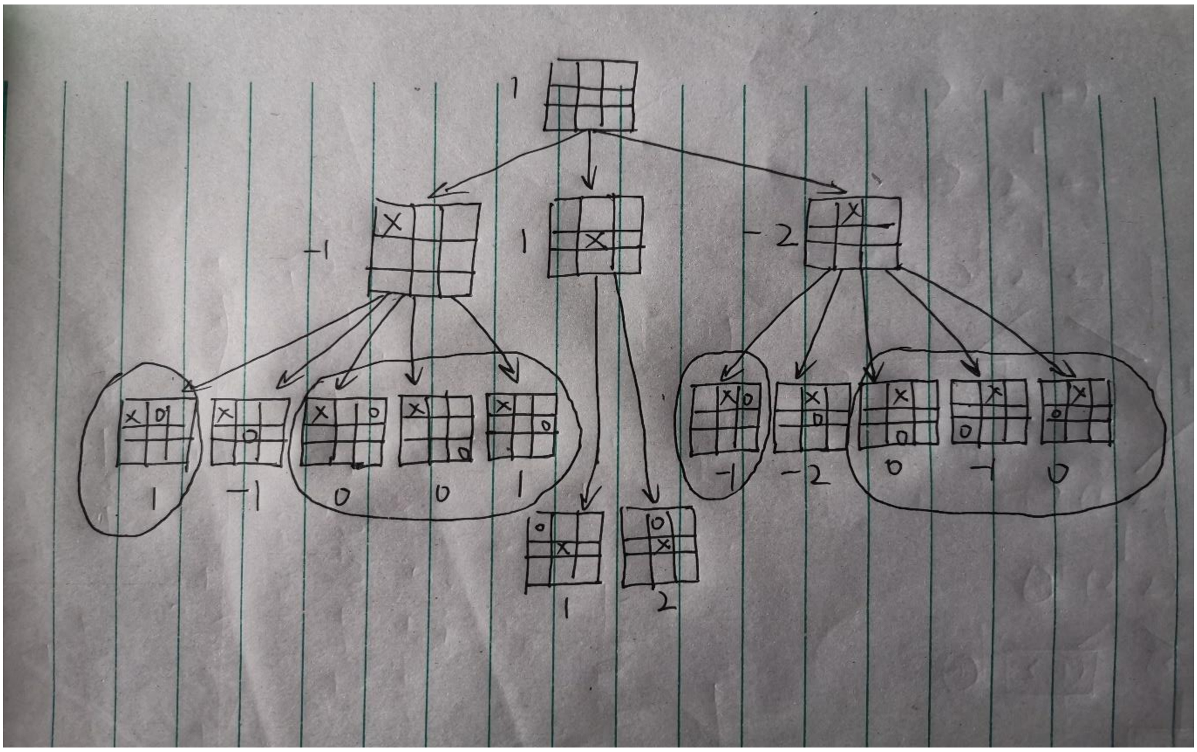
PB17151767 焦培淇

5.9

a: 对整个棋局进行放缩，假设一方在胜利后继续行棋，那么此时会得到整个可能棋局的一个上界，因此可能的棋局情况如下所示：

$$2[\sum_{k=1}^{4}(C_9^k C_{9-k}^k + C_9^k C_{9-k}^{k-1}) + C_9^5$$

b: 博弈树如下：



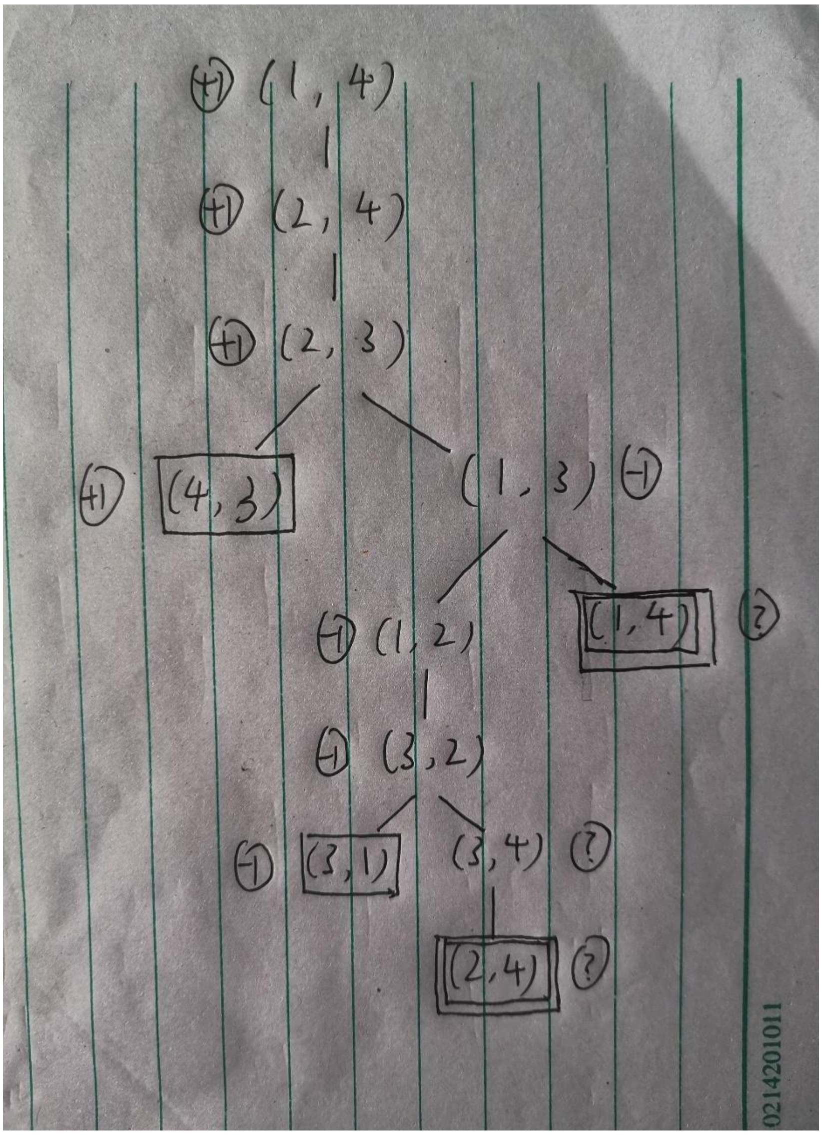
c: 效用函数的值见上图所示。

d: 回传值如上所示，最佳的起始步为选择中间步行棋。

e: 在每次都能选择最优的分支遍历的情况下，图中被圈出的分支是会被剪掉的。

5.8

a:　结果如下图所示：



b:　对于每个选手我们可以这样设定：当他可以选择一种胜利走法时，不会去选择进入一个不确定的循环状态；也就是说：min(-1,?)=-1，max(1,?)=1；当所有的后继均为?时，返回的博弈值为?。

c:　对于传统的 minmax 算法，由于其采用深度优先的遍历策略，所

以在遇到重复节点是，会陷入无限循环状态，因此算法无法结束，从而无法处理这种情况。改进策略对于每次遍历到的一个状态，将它和当前遍历栈里面的状态进行比较，如果发现重复的状态，便返回？作为其博弈值。而对于？的处理按照 b 中的方法进行。

然而这种算法虽然使用于本例，但是对于有些存在不同获胜策略的游戏，每种获胜方式得到的效应值不同，此时算法即无法比较？值和不同的获胜值，并在其中做出选择。

d: 当 n 为偶数时，只需要证明 A 有必胜策略即可：在双方共走了 n-2 步后，到达（n/2, n/2 + 1）的状态，此时 A 跳一步到达（n/2 + 2, n/2 + 1）的状态，若 B 向左走到达（n/2 + 2, n/2）, A 此后每次向右移动一步即可赢得胜利，若 B 选择“阻拦”跳过 A 到达（n/2 +2 , n/2 + 3）的位置，A 只需跳过 B 即可。若此后 B 不“阻拦”，则 A 每次向右移动一步获得胜利，若 B 持续“阻拦”，最终将到达（n-1, n）的状态，此时搜索树下只有 A 赢的分支。综上，A 一定赢。

当 n 为奇数时，双方经过 n-2 步后到达（(n+1）/2, (n+1)/2 + 1）的位置，但此时 B 行动，B 只需采取当 n 为偶数时 A 采取的策略即可获胜，原理相同，唯一不同的是当 A 持续“阻拦“B 时，B 最后可以直接跳过 A 到达（2, 1）获得游戏的胜利。

5.13

a: n2=max(n3, n31, n32, ⋯, n3b3)

对于一个第 i 层的节点，若其为 min 层,则

ni = min (n(i+1), n(i+1)1, n(i+1)2, …,n(i+1)b(i+1)),

如果其为 max 层, 则 ni = max (n(i+1), n(i+1)1, n(i+1)2, …,n(i+1)b(i+1)),

如此以来，逐层展开即可得到含有 nj 的 n1 的表达式。

b: n1 = min (l2, n2, r2) = min (l2, max (l3, n3 ,r3) ,r3) = …

依次递归的展开下去到 nj 即可。

c: 变换上述表达式可见：nj 只需要超过 min(l2,l4,l6……lj)即可，此时按照上述式子产生的 n1 值即和 nj 无关，此时即可成功剪枝。

d:当节点为 min 时，同样参考上面的推导式，可见此时的上界为 max(l3,l5,l7……lk)