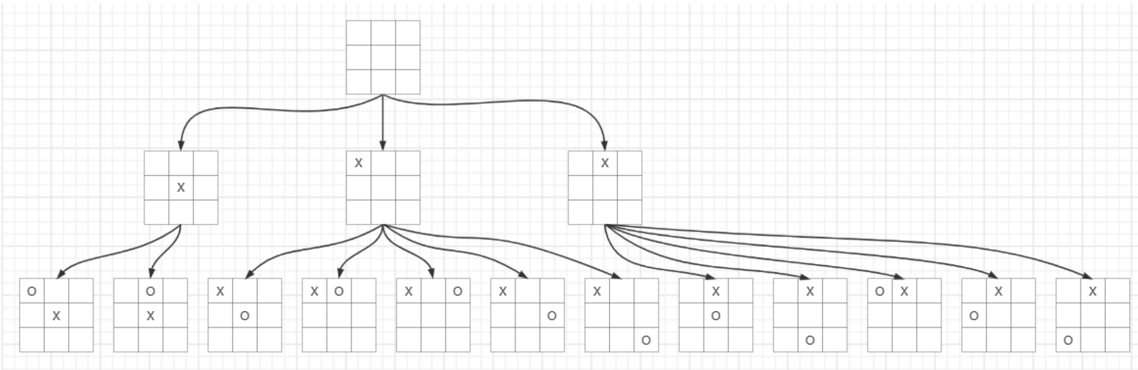
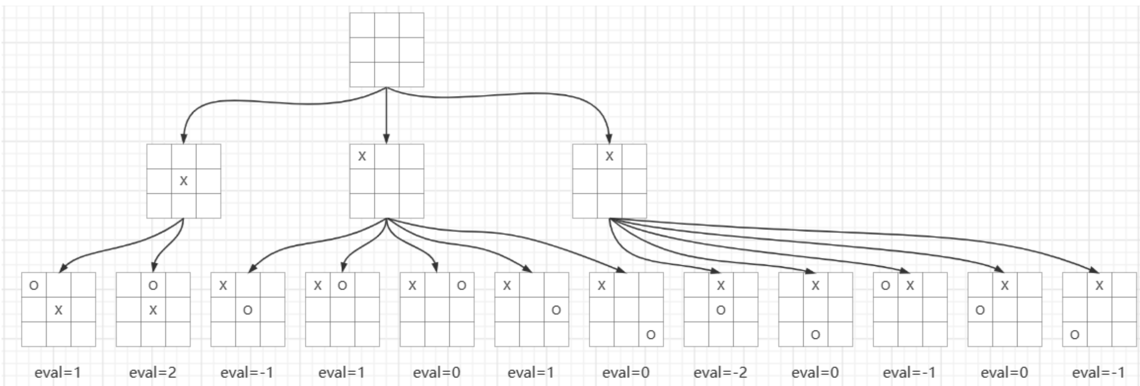
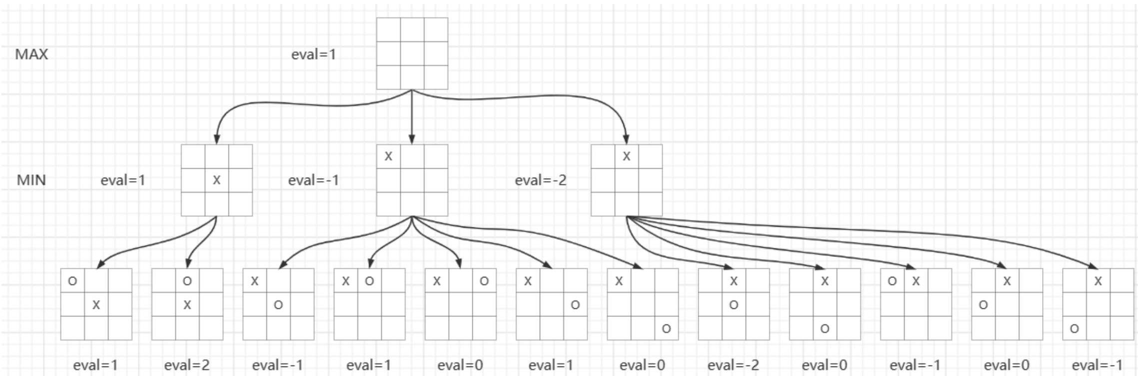
HW4

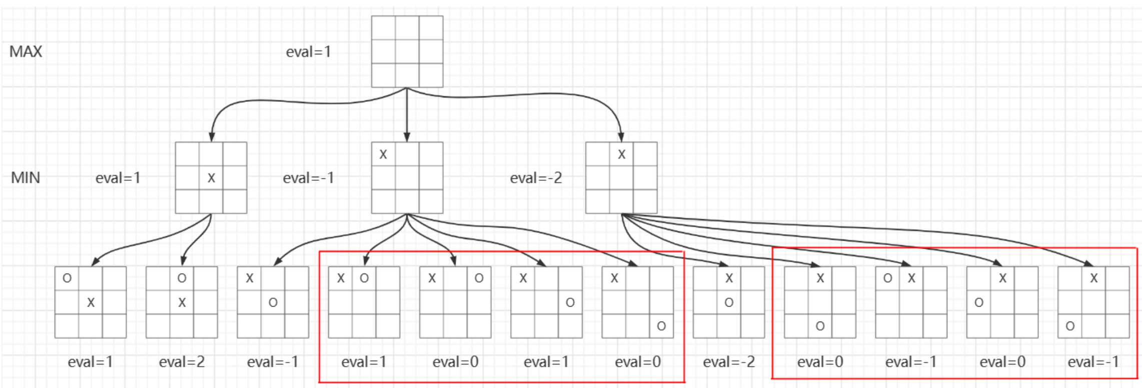
PB20111689 蓝俊玮

5.9

1. 最多可能有 $9!$ 个井字棋局数。这是两位玩家将井字棋所有位置都填满的情况。一般情况下的井字棋局数可能比较早结束，因此一般情况下不会有这么多。
2. 
3. 

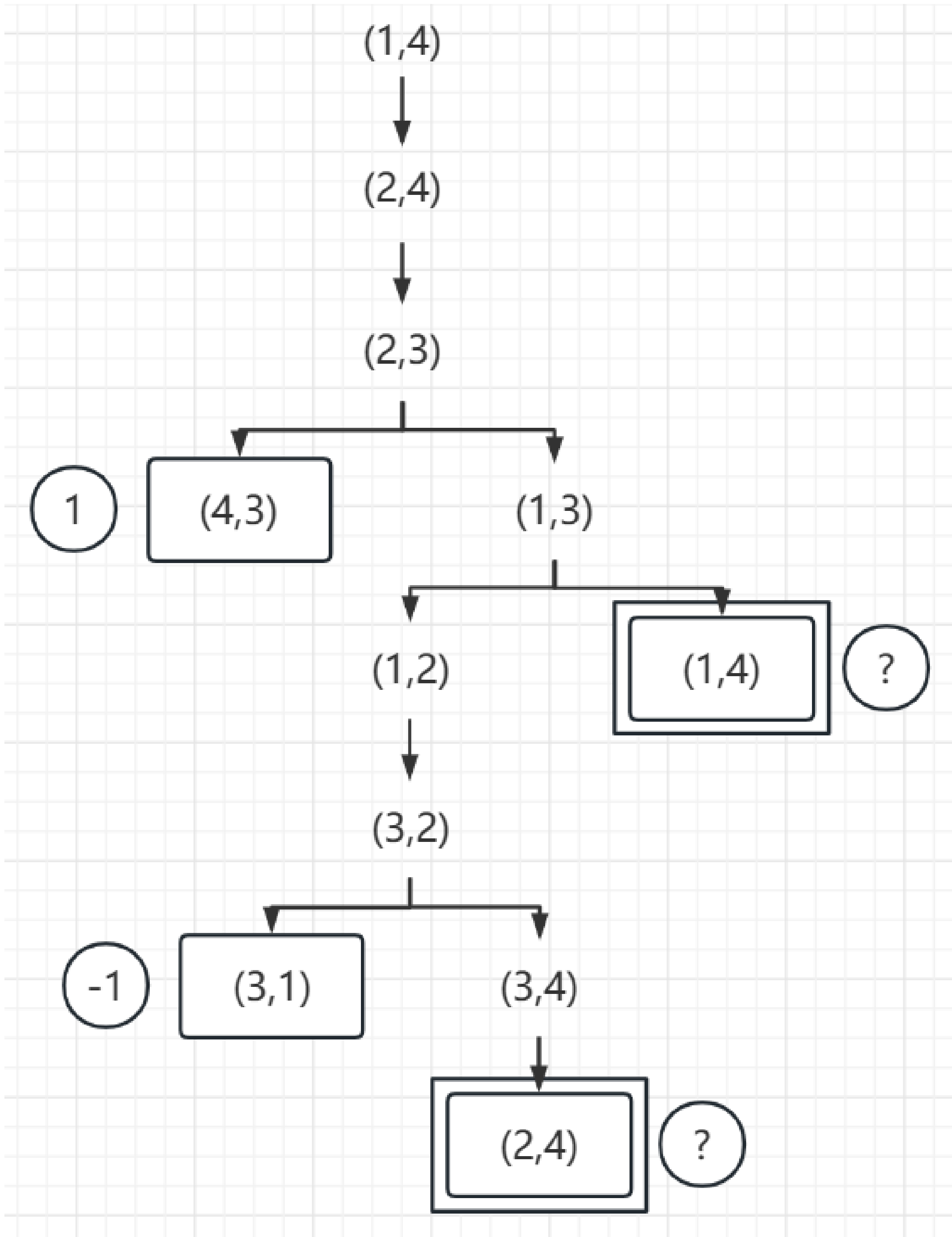
4. 

   根据这些值，我们会选择深度为 2 的第一个棋局。即 X 第一步放到正中间，而 O 放到角落。
5. 假设节点按对 $\alpha-\beta$ 剪枝的最优顺序生成，那么首先走最左边的树，可以得到如下结果：



5.8

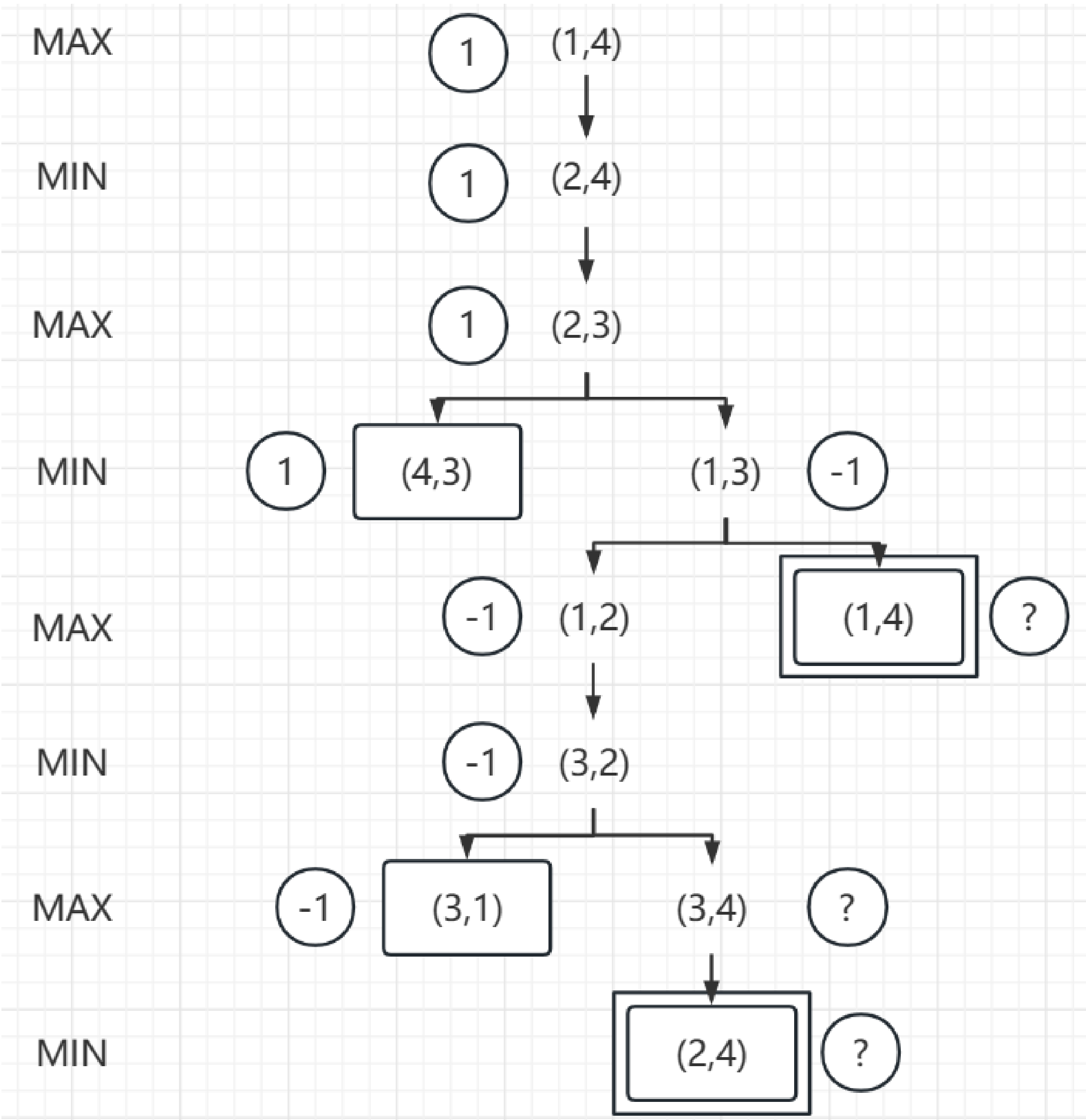
1.



2. 因为 $?$ 的取值都是正负 1，因此我们使用 $\min()$ 和 $\max()$ 函数可以推出这样的结论：

$$\min(?,1) = ?,\quad \max(?,1) = 1$$
$$\min(?,-1) = -1,\quad \max(?,-1) = ?$$
$$\min(?,?) = ?,\quad \max(?,?) = ?$$

所以使用 minimax 之后的结果如下：



3. 如果使用标准的 minimax 算法的话，则会因为出现重复状态而导致无限循环。解决方案：记录已经访问过的状态，每当新动作产生一个状态时先判断其是否访问过，如果访问过，就按 (b) 中处理方案，给重复状态记录上值？，并按照？的 $\min()$ 和 $\max()$ 函数规则更新计算 minimax 算法。修改后的图就如上。
4. 当 $n=3$ 的时候，我们知道 $(1,3)\to(2,3)\to(2,1)$，所以 A 一定会输。而当 $n=4$ 的时候，从上述例子我们可以看出 A 一定会赢。那么当 $n>4$ 的时候，采用归纳法。假设 $n=2k$ 的时候，A 一定会赢成立，而当 $n=2k+1$ 的时候，A 一定会输成立。则当 $n=2(k+1)$ 的时候，由于开始时 A 和 B 都只能相向的走一步，因此这时 $n=2(k+1)-2=2k$，由假设可知，所以 A 一定会赢。而当 $n=2(k+1)+1$ 的时候，由于 A 和 B 都只能相向的走一步，因此这时 $n=2(k+1)+1-2=2k+1$，由假设可知，所以 A 一定会输。

5.13

1. 对于 n_2，我们可以给出类似的表示：

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

类似地，将这些表示带入，可以得到用 n_j 表示 n_1 的结果如下：

$$n_1=\min(\max(\min(\ldots(\min(n_j,n_{j1},\ldots,n_{jb_j}),n_{(j-1)1},\ldots,n_{(j-1)b_{(j-1)}})),n_{31},\ldots,n_{3b_3}),n_{21},\ldots,n_{2b_2})$$

2. 用 l_i 和 r_i 的值重写 n_1 的表达式：

$$n_1=\min(l_2,\max(l_3,n_3,r_3),r_2)$$

则用 n_j 表示 n_1 的结果如下：

$$n_1 = \min(l_2, \max(l_3, \min(\ldots(\ldots, \max(l_{j-1}, \min(l_j, n_j, r_j), r_{j-1}), \ldots)\ldots), r_3), r_2)$$

3. 若 $n_j > l_j$，那么 $\min(l_j, n_j, r_j)$ 就与 n_j 无关。

   若 $n_j > l_{j-2}$，那么 $\min(l_{j-2}, \max(l_{j-1}, \min(l_j, n_j, r_j), r_{j-2}), r_{j-2})$ 最终也与 n_j 无关，因此可以推出，若 $n_j > \min(l_2, l_4, \ldots, l_j)$ 的话，则 n_1 就与 n_j 无关。

   同理，若 $n_j < l_{j-1}$ 的话，则 $\max(l_{j-1}, \min(l_j, n_j, r_j), r_{j-2})$ 也与 n_j 无关。

   所以综上所述，需要满足 $\max(l_1, l_3, \ldots, l_{j-1}) < n_j < \min(l_2, l_4, \ldots, l_j)$ 才能影响到 n_1 不被剪枝。

4. 替换一下得：

   若 $n_j < l_j$，那么 $\max(l_j, n_j, r_j)$ 就与 n_j 无关。

   若 $n_j < l_{j-2}$，那么 $\max(l_{j-2}, \min(l_{j-1}, \max(l_j, n_j, r_j), r_{j-2}), r_{j-2})$ 最终也与 n_j 无关，因此可以推出，若 $n_j < \max(l_2, l_4, \ldots, l_j)$ 的话，则 n_1 就与 n_j 无关。

   同理，若 $n_j > l_{j-1}$ 的话，则 $\min(l_{j-1}, \max(l_j, n_j, r_j), r_{j-2})$ 也与 n_j 无关。

   所以综上所述，可以得到最终结果如下：

$$\max(l_2, l_4, \ldots, l_j) < n_j < \min(l_1, l_3, \ldots, l_{j-1})$$