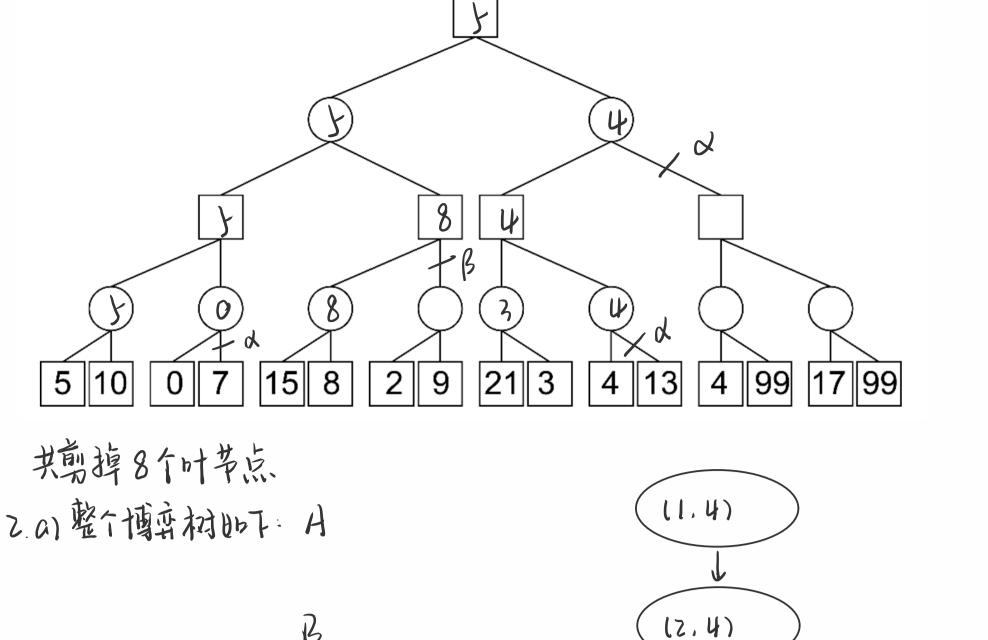
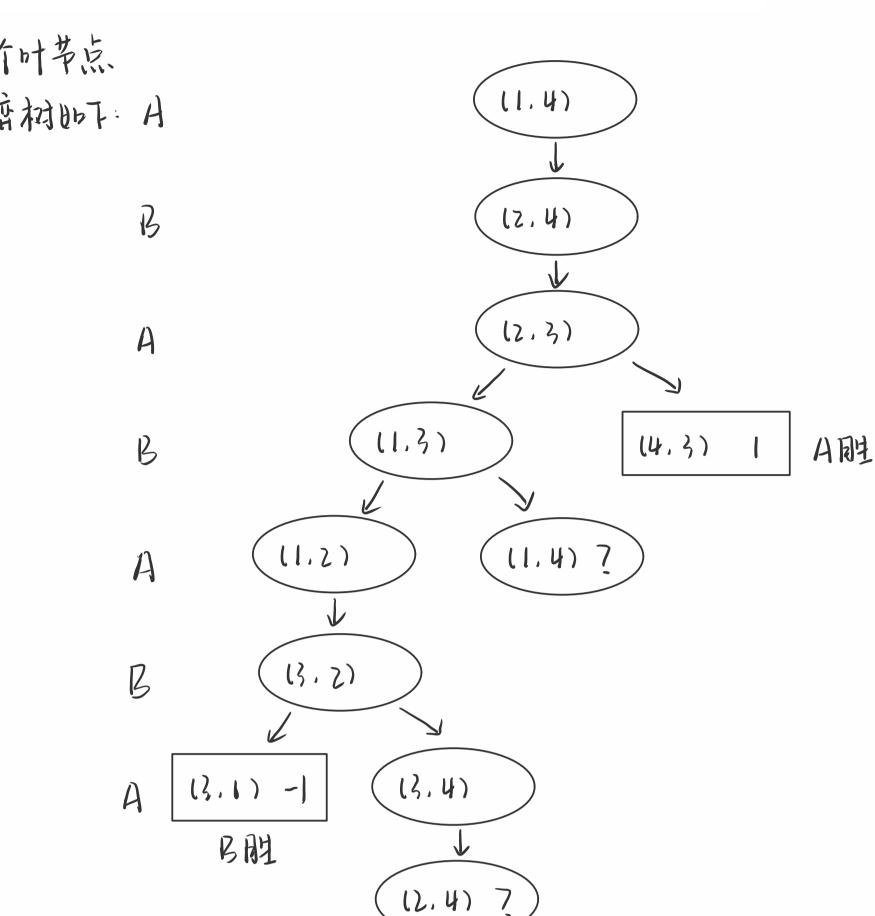
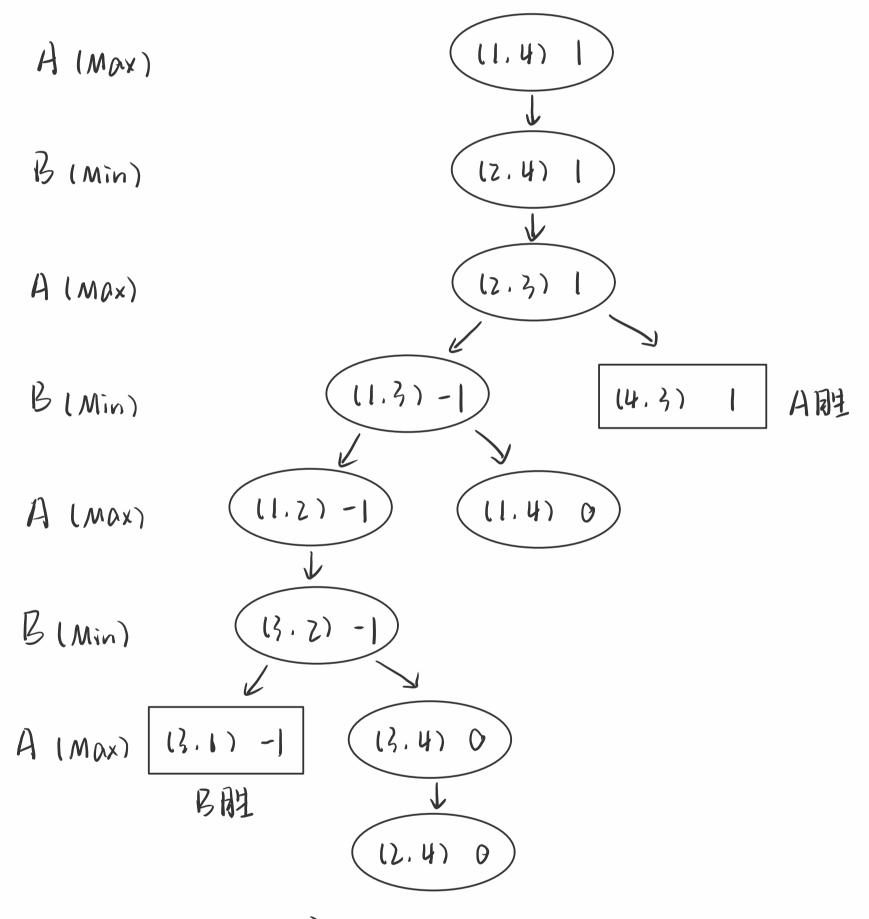
1.结果如下:





的给出每个节点倒推的极大极小值。结果如下:



对于循环节点,设置重复状态检测,若已经访问过该状态,则进行剪枝,避免再次拓展相同状态,并考虑将该节点的价值设置为0,意味着A或片都无法获胜。

原因:可以防止搜索过程中陷入死循环,若不进行重复状态检测,搜索构中会反复访问相同的状态,可能导致搜索算法无限循环下去,无法得出有效的结果;可以提高搜索效率,节省时间和计算资源;可以避免存储已经访问过的状态信息,从而节省内存消耗;并且将价值设置为 0,在例准取极大极小的过程中一般会被会弃掉,可以认为对最终结果的影响较小。

C) 因为博弈树中存在循环,标准的极小极大算法设有处理循环节点的机制,所以可能导致程序陷入死循环。 修正在搜索过程中维护一个已活的过状态的集合。当遇到一个新状态明、首先检查是否已访问过该状态,若 已访问过,则不再就展该节点,并将该节点的价值设置为 0,从便不后绕倒,推取极大极小值 不能保证给出最优决策。因为存在循环的情况下,博弈树的搜索空间可能非常庞大,即使使用重复状态检 测、仍然可能无法完全穷尽搜索空间,错过最优解;并且在处理循环的,通常需要启发式的方法来估计节 点的价值,但受限于问题的复杂性和特殊性,启发式方法不能保证准确估计节点的价值,从向导致无法找到最优解。

d) 使用归纳法证明如下:

n=1或2时,无弦进行游戏,元需考虑;

n-3时, A首先移动到2, B侧过A移动到1, 显然B获胜; n=4时, 上述已证明A获胜;

假设加水时,A获胜;老底加水时隔沉;

A首先移动到 2, 只移动到 2k+1, 考虑 2, …, 2k+1 定 水子位置, 则可从移游戏状态变为 ni- 2k 的情况;根据归纳设设 ni- 2k 的 A 获胜, 故当 A 移动到 2k+1 位置的, R处然不在 2位置, 否则 ni- 2k 时 B 先子 A 获胜, 与假设矛盾, 所以 A 可以先于 B 到达 2k+2位置, 即 ni- 2k+2 时, A 获胜。

n为奇数时,证明过程与n为偶数同俚。

故A当n为偶数时获胜, 奇数时失败, 证毕