知识点总结（搜索）

- 搜索算法框架：树搜索
  - 集合$\mathscr{T}$用于保存搜索树中可用于下一步探索的所有候选结点
- 图搜索–不允许环路的存在
  - 记录已访问节点：维护一个已访问节点集合
- 贪婪最佳优先搜索
- A算法
  - 评价函数：f(n)=g(n)+h(n)
    - g(n)表示从起始结点到结点n的开销代价值
    - h(n)表示从结点n到目标结点路径中所估算的最小开销代价值
    - f(n)可视为经过结点n、具有最小开销代价值的路径。
- 对抗搜索
  - 最小最大搜索
  - Alpha-Beta剪枝搜索
  - 蒙特卡洛树搜索
- 蒙特卡洛树搜索
  - 单一状态蒙特卡洛规划： 多臂赌博机
  - 多臂赌博机问题是一种序列决策问题，这种问题需要在利用(exploitation)和探索(exploration)之间保持平衡。
  - 两种策略学习机制：
    - 搜索树策略: 从已有的搜索树中选择或创建一个叶子结点(即选择和拓展)。搜索树策略需要在利用和探索之间保持平衡。
    - 模拟策略：从非叶子结点出发模拟游戏，得到游戏仿真结果。
  - 上限置信区间 (Upper Confidence Bound, UCB)
    - 在UCB方法中，使$X_{i,T_i(t-1)}$来记录第i个赌博机在过去t-1时刻内的平均奖赏，则在第t时刻，选择使如下具有最佳上限置区间的赌博机：
    - 其中$c_{t,T_i(t-1)}$取值定义如下：$c_{t,s}=\sqrt{((2lnn)/(T_i(t-1)))}$
    - 也就是说，在第⊠时刻，UCB算法一般会选择具有如下最大值的第⊠个赌博机：
      - $UCB=X_j{}^-+\sqrt{((2lnn)/n_j)}$ 或者$UCB=X_j{}^-+C\times\sqrt{((2lnn)/n_j)}$
  - 蒙特卡洛树搜索基于采样来得到结果、而非穷尽式枚举 (虽然在枚举过程中也可剪掉若干不影响结果的分支)。
  - UCB1算法中的置信上界R并不是对单次奖励的上界，而是对期望奖励的置信上界。它是一个统计估计，不是对单次结果的限制
  - 在多臂赌博机问题中,过度利用可能导致算法对部分臂膀额奖励期望估计不准确。

- 过度利用（Over-exploitation）指的是算法过早地集中选择当前看似最优的臂膀，而忽略了对其他臂膀的探索。
- 如果某个臂膀很少被选择，样本数量n_i很小，小样本导致估计值方差大，不够可靠
- 算法进入扩展步骤时,当前节点的所有子节点必然都未被扩展。（×）
  - ✔️ 节点未完全扩展（还有动作没对应子节点）
  - ❌ 不是所有子节点都未被扩展（可能已有部分子节点）
  - MCTS的扩展是增量式的，每次只扩展一个子节点，而不是等到所有子节点都不存在时才开始扩展。这种设计更加高效和实用。
-