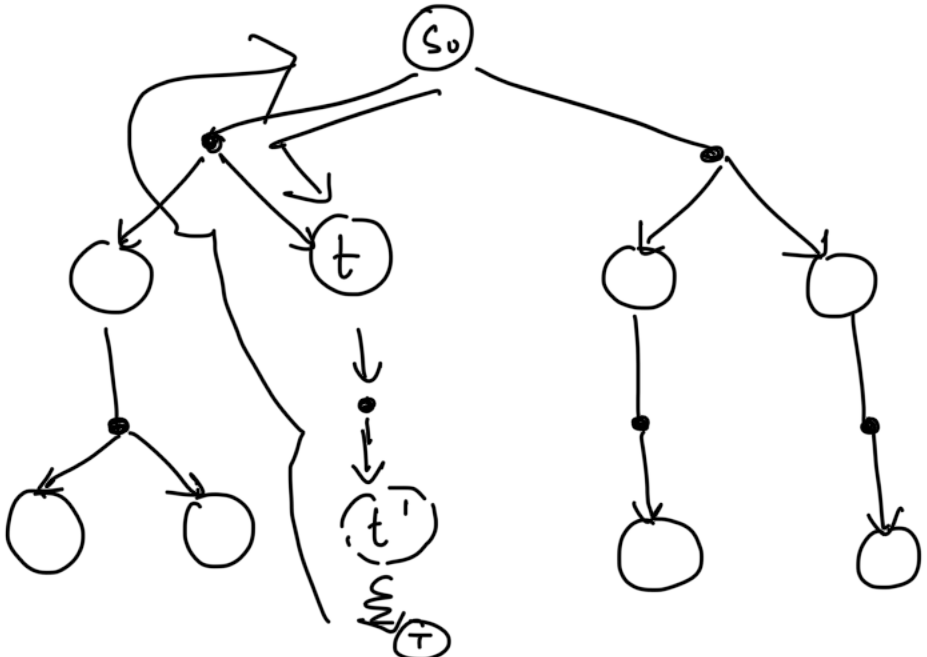
Week-12 RL

L1 Monte Carlo Tree Search

- Concept of roll outs

$s_t \quad \pi(s_t) \quad \sim \pi(s_t, \cdot)$

- Online planning → planning is done immediately before executing an ac^n. Once an ac^n / seq. of ac^{ns} is executed, we start planning from the new state. So, planning & execn are interleaved.
  1. for each state s, the set of ac^{ns} $A(s)$ are partially evaluated.
  2. $Q(s, a)$ is calculated by avg. the expected reward of traj. ours.
  3. ac^n chosen is $\text{argmax}_a Q(s, a)$.
- consider MDP as a tree —



1. selec^n – start at the root node & successively select a child until you hit a node that is n't fully expanded.
2. expansion – expand the children of the selected node by choosing an ac^n with selec^n policy & creating new nodes using the ac^n outcomes.
3. Simula^n – choose one of the new node & perform a random simula^n of MDP to the terminating state with rollout policy.
4. backup – given reward r at the terminating state, backup & save the reward to calculate the value $Q(s, a)$ at each state on the path.

- MCTS Search algo.

```
func^n (MCTS)(s_0)
    while budget do
        selected node ← select(s_0)
        child ← expand(selected node)
        G ← simulate(child)
        Backup(child, G)
    return argmax_a(Q(s_0, a))
```

- properties of MCTS –

1. aheuristic – no domain specific knowledge.
2. asym. – tree expansion in terms of finding the next nodes is asym.
3. anytime – it backups the outcome of each simulation immediately which ensures all-values are up-to-date with every iteration of algo. So, we can return back to any stage at anytime.

- UCT (Upper confidence for trees)

UCT = UCB1 + MCTS

$$\rightarrow \text{argmax}_{a \in A(s)} \left[Q(s,a) + 2c \sqrt{\frac{2 \ln N(s)}{N(s,a)}} \right]$$

$N(s) \rightarrow$ no. of tonies a state node was visited.

$N(s,a) \rightarrow$ no. of tonies 'a' has been selected from this node.

$c > 0 \rightarrow$ exploration constant

- applicⁿ —

1. AlphaGo 0
2. Chess, Shogi
3. Real time strategy games
4. planning & scheduling