Exercise 6
Gregory Attra
03.10.2022
CS 7180 - Prof. Amato
Code Setup:

1. Unzipped the 'ex6.zip' file
2. 'cd' into the 'code' directory
3. run 'source ./init' to setup the pyenv

Questions:

1. (a) It would perform slightly better as, after the first episode, the final n-steps would be updated. But it would still not perform as well as Dyna-Q as we would still not make as many updates. In Dyna-Q, we could sample $n+1$ states and have better results, especially if we used prioritized sampling. We would also make updates to states not necessarily in our trajectory, unlike with n-step SARSA.

   (b) i. Advantage: We have a higher likelihood of making meaningful updates; in other words we have a higher likelihood of sampling a trajectory where $Q(s,a)$ was updated from learning or from a prior planning update

   ii. Disadvantage: Computationally more expensive: $M \times N$ samples each step.

2. The benefit of Dyna-Q+ is that it incentivizes exploring, but this can also mean taking suboptimal actions. In this case, Dyna-Q+ learned a more optimal policy initially, but overtime began to explore less-visited states, which brought down the average reward.

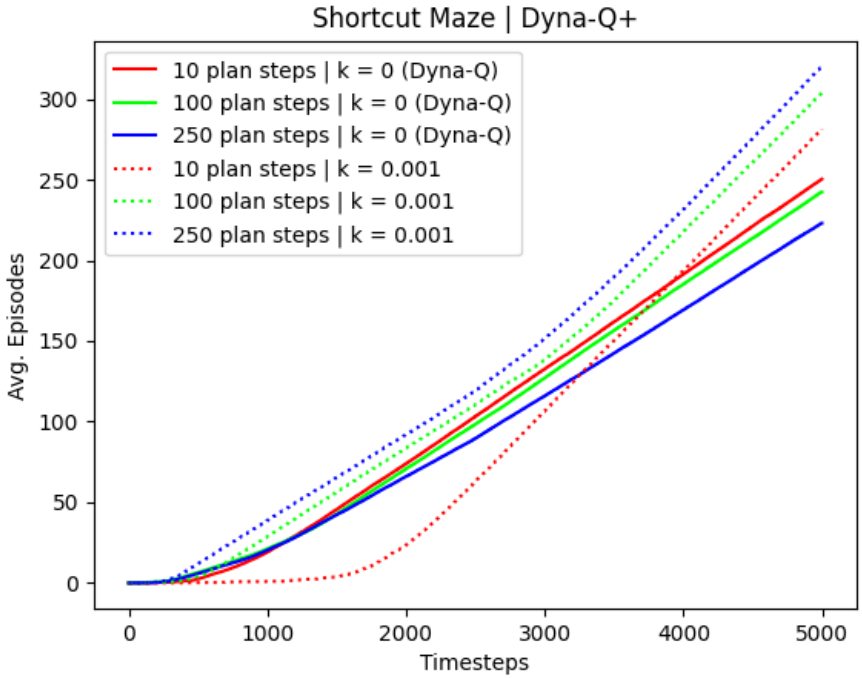
3. (a) The modifications to make to Dyna-Q are:
   - For each state/action pair, track the most-recent timestep at which that state/action pair was taken in the experience phase.
   - In the planning phase, compute $\tau = t - T[s][a]$ where t is the current timestep and $T[s][a]$ is the most recent timestep at which the state action pair (s,a) was executed in the experience phase.
   - Add $k\sqrt{\tau}$ to the model's reward for the state action pair: $r = M[s][a]$; where k is some exploration balancing constant.

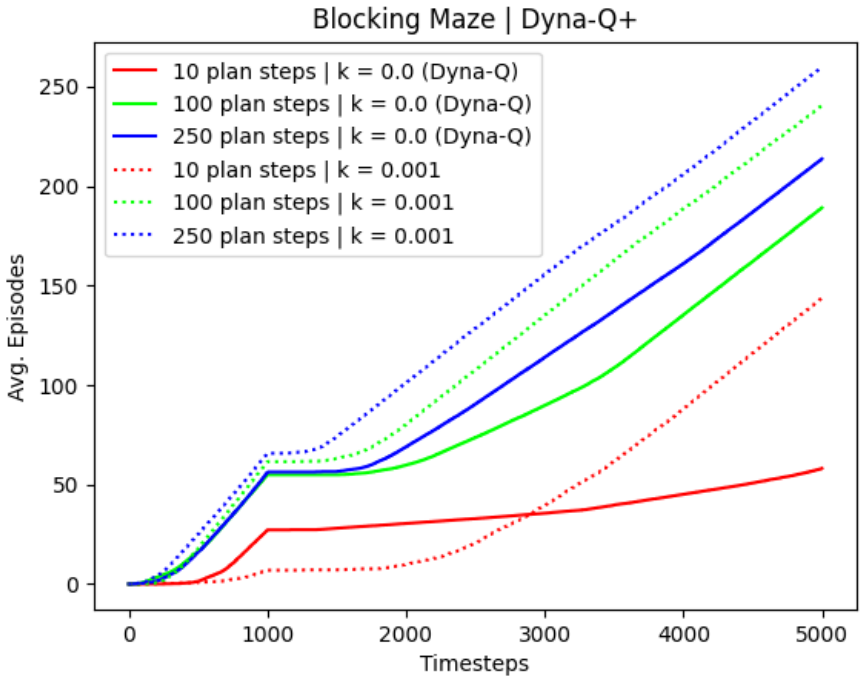
   (b) To run the Dyna-Q+ implementation:

   i. Follow the code setup instructions at the top of this document

   ii. Run 'python src/planning/run_shortcut_maze.py'

   iii. Run 'python src/planning/run_blocking_maze.py'


Shortcut Maze | Dyna-Q+
10 plan steps | k = 0 (Dyna-Q)
100 plan steps | k = 0 (Dyna-Q)
250 plan steps | k = 0 (Dyna-Q)
10 plan steps | k = 0.001
100 plan steps | k = 0.001
250 plan steps | k = 0.001
Avg. Episodes
Timesteps
0
50
100
150
200
250
300
0
1000
2000
3000
4000
5000


Blocking Maze | Dyna-Q+
10 plan steps | k = 0.0 (Dyna-Q)
100 plan steps | k = 0.0 (Dyna-Q)
250 plan steps | k = 0.0 (Dyna-Q)
10 plan steps | k = 0.001
100 plan steps | k = 0.001
250 plan steps | k = 0.001
Avg. Episodes
Timesteps
0
50
100
150
200
250
0
1000
2000
3000
4000
5000

4. (a) In action selection, the bonus is ephemeral; that is: it is used only after a one-time computation and then thrown out. It does not integrate into the value function.

   When the bonus is applied to the is applied to the Q-function, it becomes a "permanent" part of the system and can only be "erased" through more experience at that state/action pair. In this case, we are making our Q-function inaccurate, whereas in action selection we are overriding our Q-function.

   (b) i. Advantage of bonus applied to action selection: Not a permanent change to our system. We aren't making our Q-function inaccurate.

   ii. Disadvantage of bonus applied to action selection: Only doable in small-dim state spaces as we must iterate over all $\{state, action\}$ pairs and compute the bonus for each.

   iii. Advantage of bonus applied to Q-function: can be done for high-dim state space as we embed the bonus in the Q-value.

   iv. Disadvantage of bonus applied to Q-function: bonus becomes embedded in the system and must be overcome through experience. This leads to taking more suboptimal actions overtime.

5. (a) In stochastic environments, the transitions for a given $\{state, action\}$ pair are not stochastic, and sour our model updates would have high variance and make the planning phase very inconsistent and unreliable. Further, if a bad sample is used to update the model, then our subsequent planning phase will push our Q-function in a suboptimal direction.

   (b) To fix this, our model could be an expectation of rewards for a given $\{state, action\}$ pair. That is: we could track $\{s, a, s', r, t\}$; where $\{s, a\}$ is the current state/action pair, s' is the experienced next state, r is the reward at that next state, and t is the number of times $\{s', r\}$ was experience when taking action a at state s. When we execute the planning phase, we compute the reward for the samples $\{state, action\}$ pair as the sum of rewards experienced at each s' from that $\{state, action\}$ pair, multiplied by the probability of entering s' when taking action a at state s.

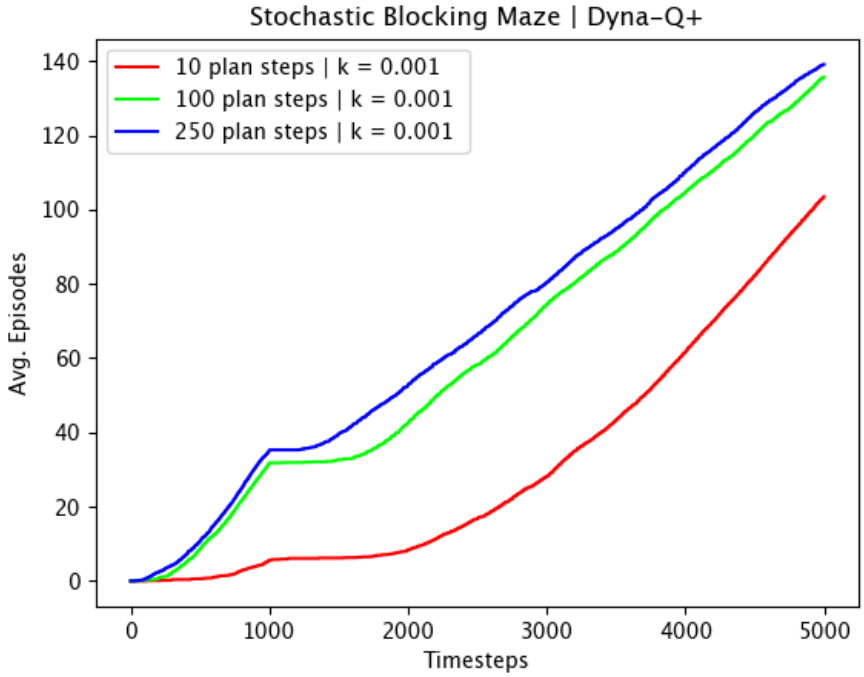
   (c) To deal with non-stationary environments, we can integrate Dyna-Q+ with the above mentioned changes to encourage exploration.

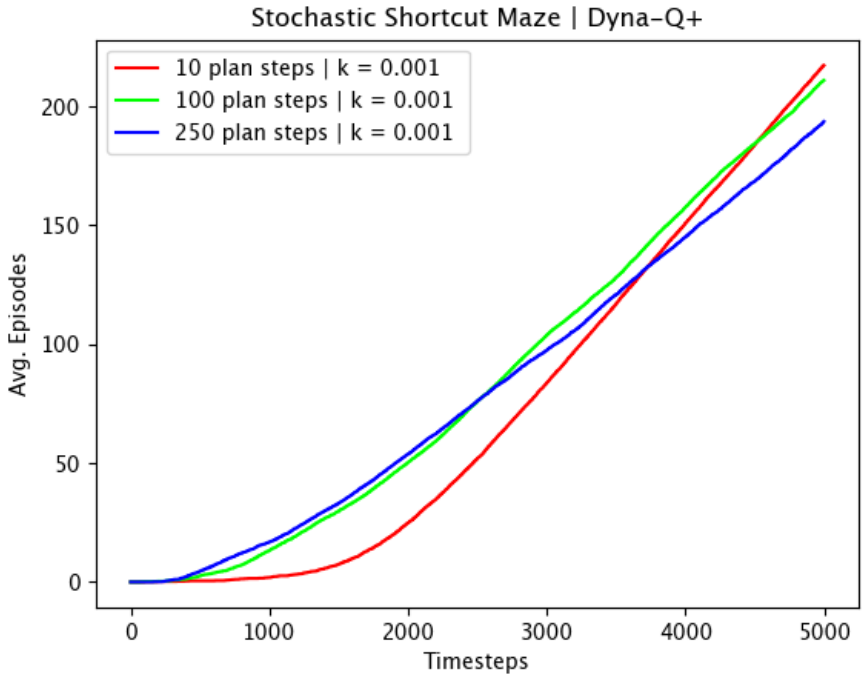
   (d) To run the code implementation of these changes:

   i. Follow the code setup instructions at the top of this document

   ii. Run 'python src/planning/run_stochastic_dynaq.py

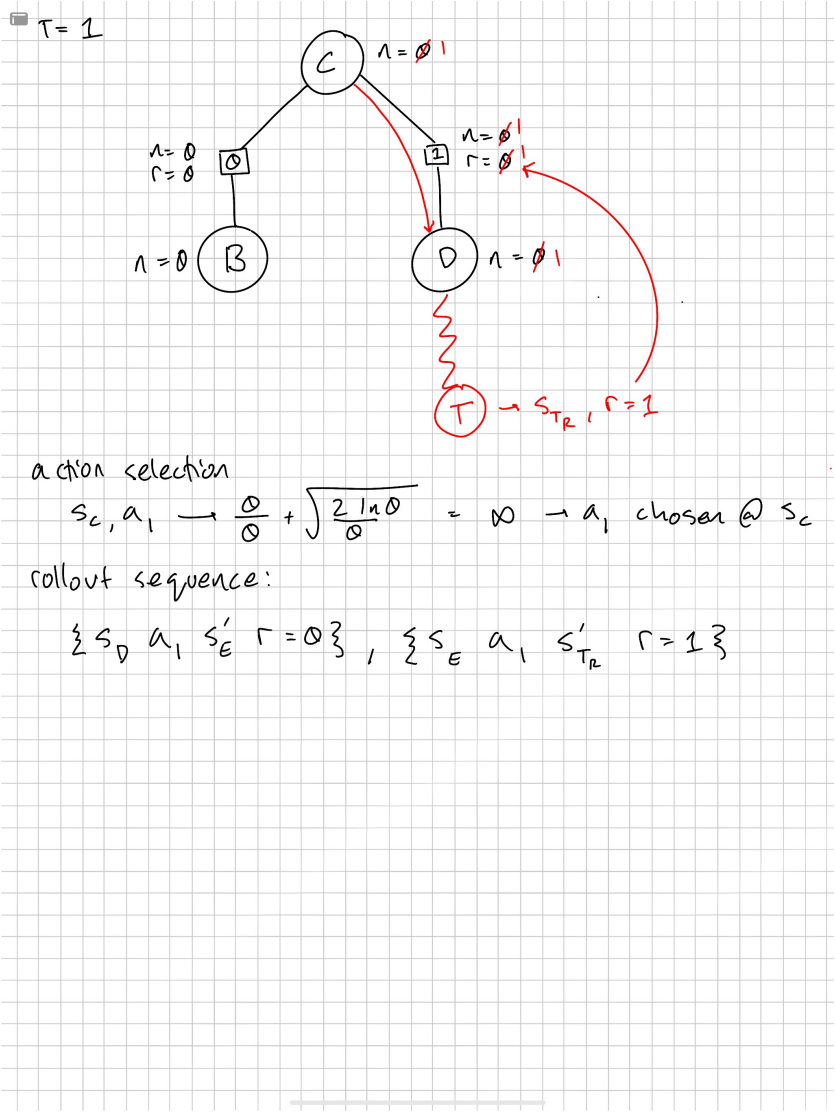
   (e) We see the average number of episodes completed by timestep 5000 in the stochastic environment is much lower than in the deterministic environment (plotted on page 2). This is could be due to slippage


Stochastic Blocking Maze | Dyna-Q+
10 plan steps | k = 0.001
100 plan steps | k = 0.001
250 plan steps | k = 0.001
Avg. Episodes
Timesteps
0
20
40
60
80
100
120
140
0
1000
2000
3000
4000
5000

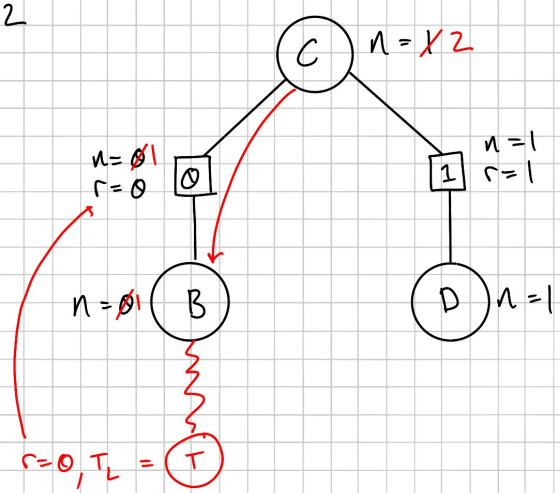

Stochastic Shortcut Maze | Dyna-Q+
10 plan steps | k = 0.001
100 plan steps | k = 0.001
250 plan steps | k = 0.001
Avg. Episodes
Timesteps
0
50
100
150
200
0
1000
2000
3000
4000
5000

causing the agent to take sub-optimal routes. Additionally, the policy may be accounting for the additional risk caused by stochasticity. The optimal path may bring the agent dangerously close to low-value states. Since the domain has slippage, there is the risk of accidentally entering one of these low-value states. The policy will learn to cushion itself from low-value states but in doing so take a safer, less-optimal, route.

6. Manual MCTS (pictures on pages below):



T = 2



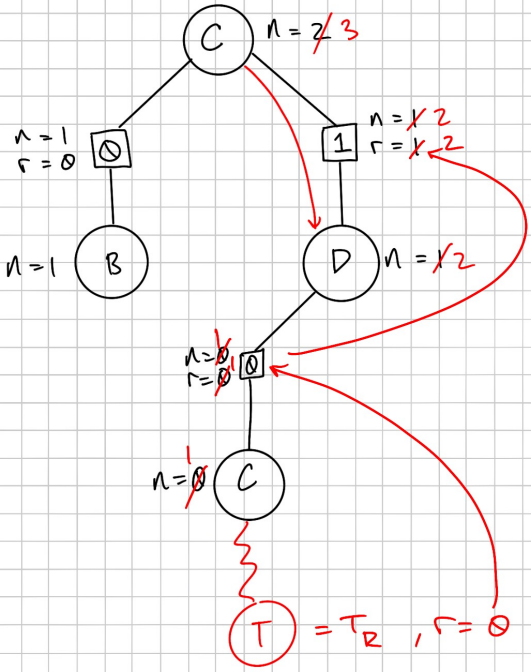
action selection:

$$s_C, a_0 = \frac{0}{0} + \sqrt{\frac{2 \ln 1}{0}} = \infty \rightarrow s_C, a_0 \text{ chosen}$$

rollout sequence:

$$\{ s_B, a_0, s_A, r=0 \}, \{ s_A, a_0, s_{T_L}, r=0 \}$$

T = 3



action selection

$s_C, a_0 = \frac{0}{1} + \sqrt{\frac{2 \ln 2}{1}} = 1.18$

$s_C, a_1 = \frac{1}{1} + \sqrt{\frac{2 \ln 2}{1}} = 1 + 1.18 = 2.18$ → chosen action

$s_D, a_0 = \sqrt{\frac{2 \ln 1}{0}} = \infty$ → chosen action

rollout sequence

$\{s_C\ a_0\ s'_B\ r=0\}\ \{s_B\ a_0\ s'_A\ r=0\}$

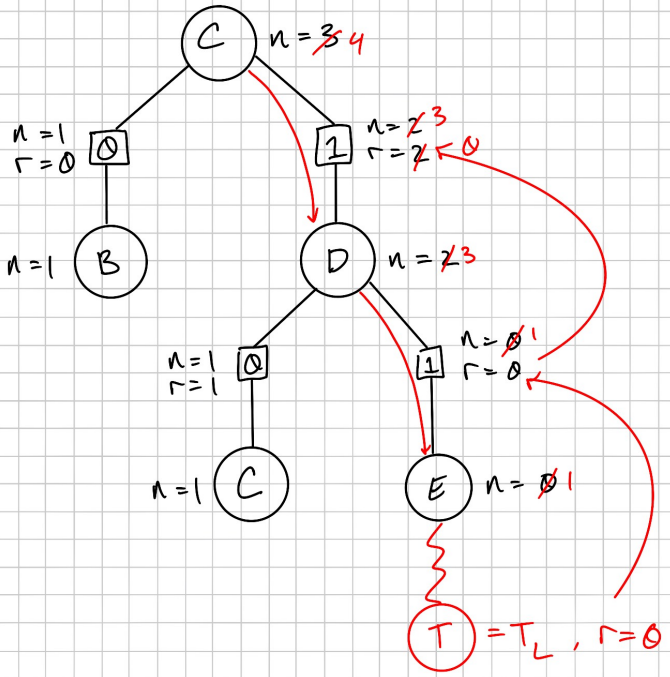
$\{s_A\ a_1\ s'_B\ r=0\}\ \{s_B\ a_0\ s'_A\ r=0\}$

$\{s_A\ a_1\ s'_B\ r=0\}\ \{s_B\ a_1\ s'_C\ r=0\}$

$\{s_C\ a_1\ s'_D\ r=0\}\ \{s_D\ a_1\ s'_E\ r=0\}$

$\{s_E\ a_1\ s'_{T_R}\ r=0\}$

T = 4



action selection:

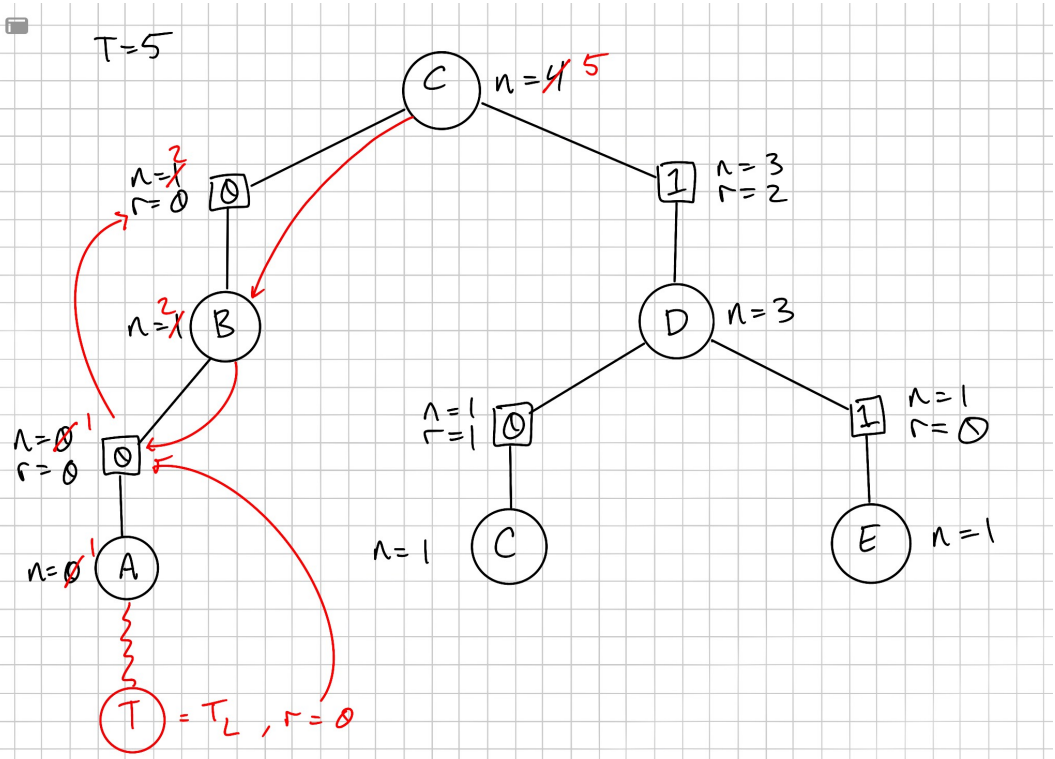
$s_C, a_0 = \frac{0}{1} + \sqrt{\frac{2 \ln 3}{1}} = 1.48$

$s_C, a_1 = \frac{2}{2} + \sqrt{\frac{2 \ln 3}{2}} = 1 + 1.05 = 2.05 \rightarrow$ chosen action

$s_D, a_1 = \frac{0}{0} + \sqrt{\frac{2 \ln 2}{0}} = \infty \rightarrow$ chosen action

rollout sequence:

$\{\{s_D, a_0, s'_C\}, \{\{s_C\ a_0\ s'_B\}\ \{s_B\ a_1\ s'_C\}\{s_C\ a_0\ s'_B\}$
$\{\{s_B\ a_1\ s'_C\}\ \{\{s_C\ a_1\ s'_D\}\ \{\{s_D\ a_0\ s'_C\}\{\{s_C\ a_0\ s'_B\}$
$\{\{s_B\ a_0\ s'_A\}\{\{s_A\ a_1\ s'_B\}\{\{s_B\ a_1\ s_C\}\{\{s_C\ a_0\ s'_B\}$
$\{\{s_B\ a_0\ s'_A\}\{\{\{s_A\ a_1, s'_C\}\{\{s_C\ a_1, s'_D\}\{\{s_D\ a_0\ s'_C\}$
$\{\{s_C\ a_0\ s'_B\}\{\{s_B\ a_0\ s'_A\}\{\{s_A\ a_0, s'_{T_L}, r=0\}$



action selection:

$s_C, a_0 = \frac{0}{1} + \sqrt{\frac{2\ln 4}{1}} = 1.66$ → action chosen

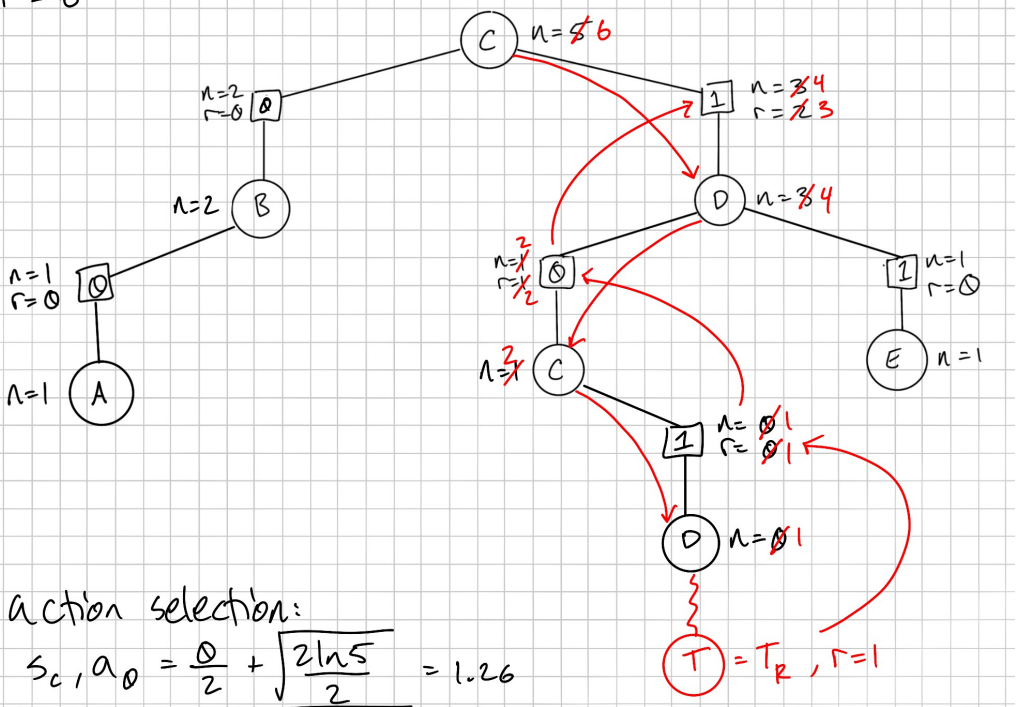
$s_C, a_1 = \frac{2}{3} + \sqrt{\frac{2\ln 4}{3}} = 0.96 + 0.66 = 1.62$

$s_B, a_0 = \frac{0}{0} + \sqrt{\frac{2\ln 4}{0}} = \infty$ → action chosen

rollout sequence:

$\{s_A\ a_0\ s_{T_L}, r=0\}$

T = 6



action selection:

$s_C, a_0 = \frac{0}{2} + \sqrt{\frac{2\ln 5}{2}} = 1.26$

$s_C, a_1 = \frac{2}{3} + \sqrt{\frac{2\ln 5}{3}} = 1.7$ → action chosen

$s_D, a_0 = \frac{1}{1} + \sqrt{\frac{2\ln 3}{1}} = 2.5$ → chosen action

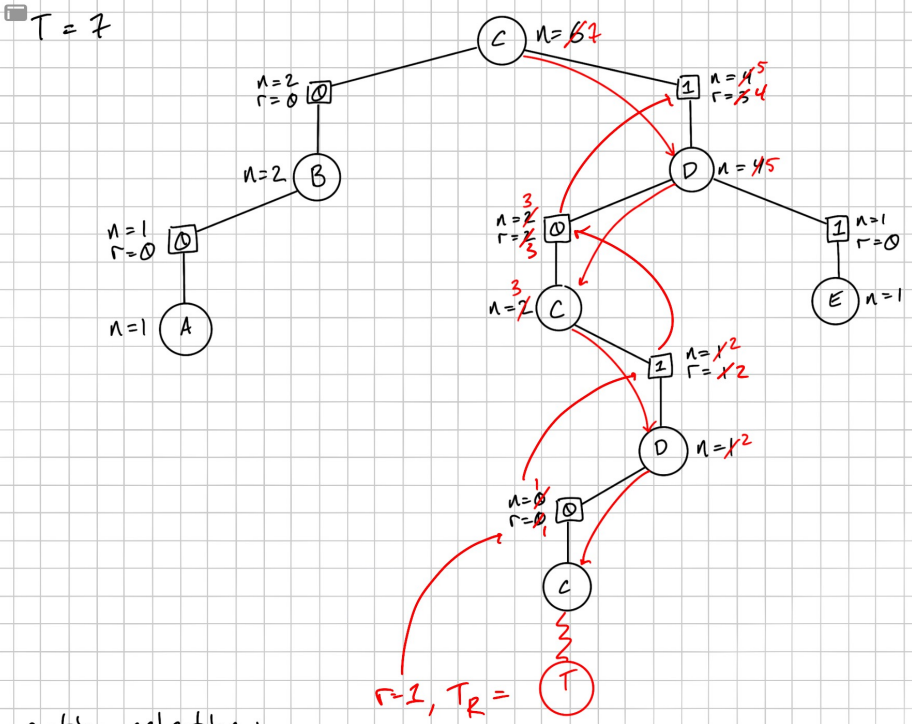
$s_D, a_1 = \frac{0}{1} + \sqrt{\frac{2\ln 3}{1}} = 1.5$

$s_{C_2}, a_1 = \frac{0}{0} + \sqrt{\frac{2\ln 1}{0}} = \infty$ → chosen action

rollout sequence:

$\{s_D\ a_1\ s'_E\}\ \{s_E\ a_0\ s'_D\}\ \{s_D\ a_1\ s'_E\}\ \{s_E\ a_1\ s_{T_R}, r=1\}$

T = 7



action selection:

$s_C, a_0 = \frac{0}{2} + \sqrt{\frac{2\ln 6}{2}} = 1.34$

$s_C, a_1 = \frac{3}{4} + \sqrt{\frac{2\ln 6}{4}} = 1.7$ → action chosen

$s_D, a_0 = \frac{2}{2} + \sqrt{\frac{2\ln 4}{2}} = 2.17$ → action chosen

$s_D, a_1 = \frac{0}{1} + \sqrt{\frac{2\ln 4}{2}} = 1.17$

$s_{C_2}, a_0 = \frac{0}{0} + \sqrt{\frac{2\ln 2}{0}} = \infty$ → action chosen

$s_{D_2}, a_0 = \frac{0}{0} + \sqrt{\frac{2\ln 1}{0}} = \infty$ → action chosen

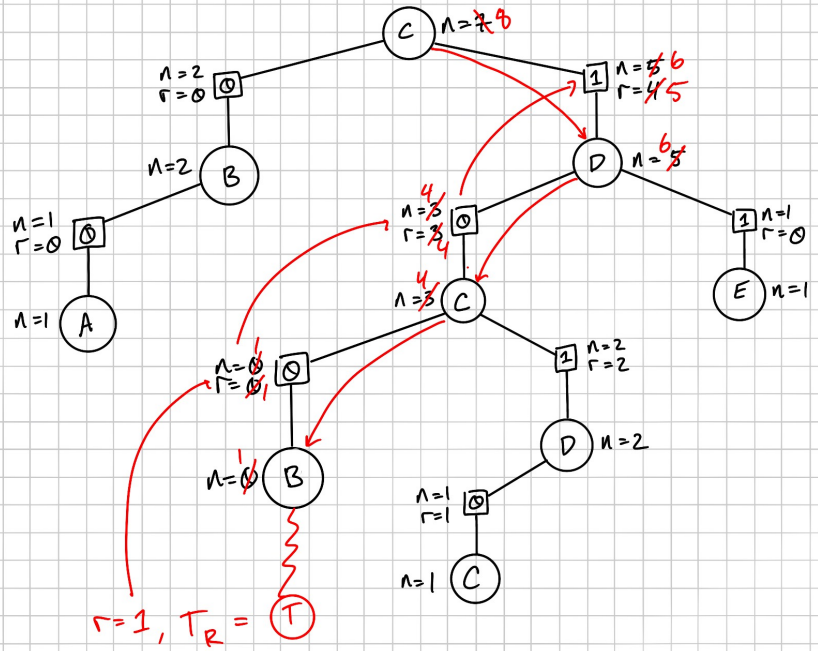
rollout sequence

$\{\{s_C\ a_1\ s'_D\}\{s_D\ a_1\ s'_E\}\{s_E\ a_0\ s'_D\}\}\ \{\{s_D\ a_1\ s'_E\}\{s_E\ a_0\ s'_D\}\}$

$\{\{s_D\ a_0\ s'_C\}\{s_C, a_0, s'_B\}\{s_B\ a_1\ s'_C\}\}\ \{\{s_C\ a_1\ s'_D\}\{s_D\ a_1\ s'_E\}\{s_E\ a_0\ s'_D\}\}$

$\{\{s_D\ a_0\ s'_C\}\{s_C\ a_1\ s'_D\}\{s_D\ a_1\ s'_E\}\{s_E\ a_1\ s_{T_R}, r=1\}\}$

T = 8



action selection

$s_C, a_0 = \frac{0}{2} + \sqrt{\frac{2\ln 7}{2}} = 1.4$

$s_C, a_1 = \frac{4}{5} + \sqrt{\frac{2\ln 7}{5}} = 1.7$ → action chosen

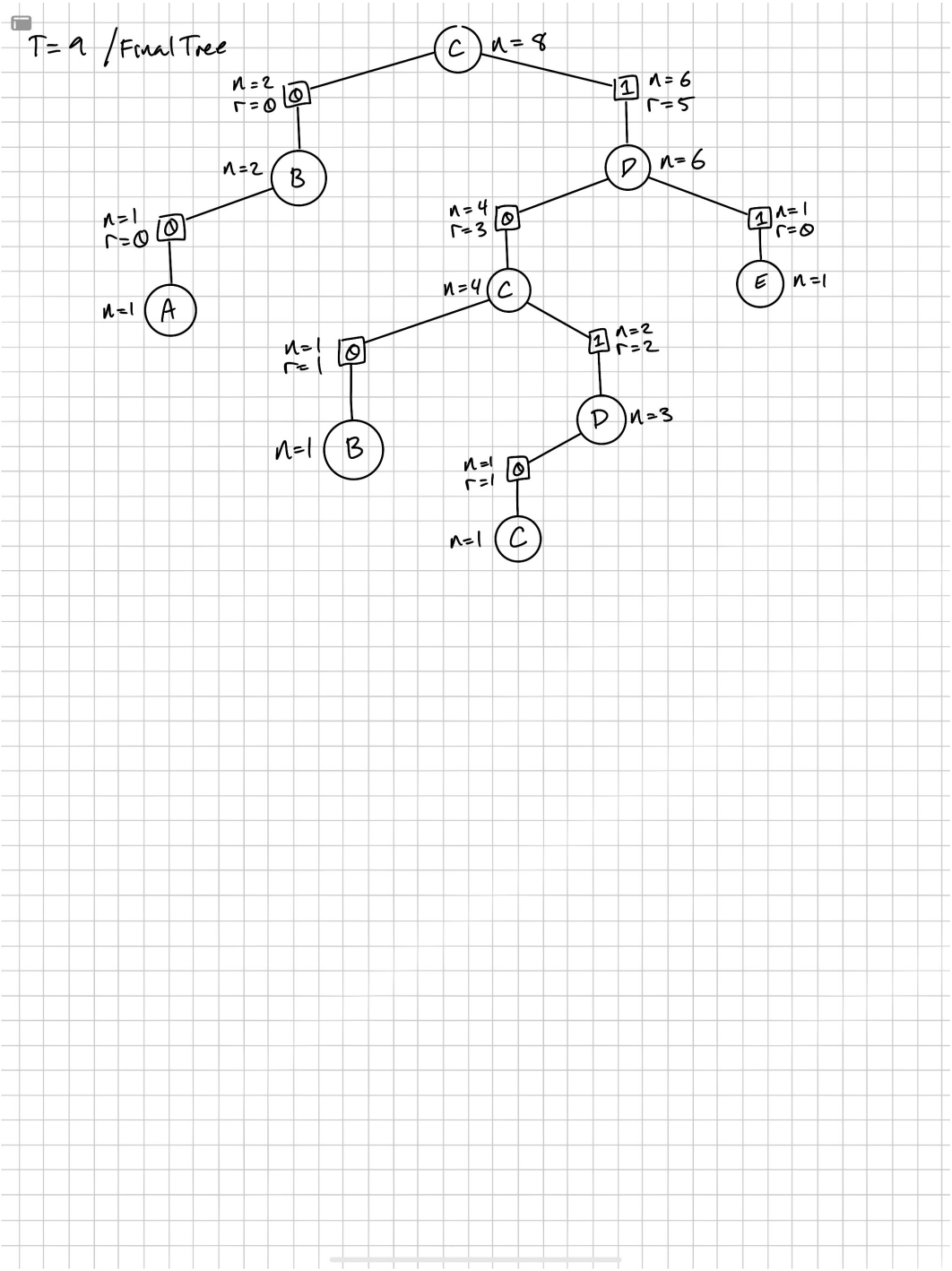
$s_D, a_0 = \frac{3}{3} + \sqrt{\frac{2\ln 5}{3}} = 2.03$ → action chosen

$s_D, a_1 = \frac{0}{1} + \sqrt{\frac{2\ln 5}{1}} = 1.8$

$s_{C_2}, a_0 = \frac{0}{0} + \sqrt{\frac{2\ln 3}{0}} = \infty$ → action chosen

rollout seq:

$\{\{s_B\ a_0\ s'_A\}\}\ \{\{s_A\ a_1\ s'_B\}\}\ \{\{s_B\ a_1\ s'_C\}\}\ \{\{s_C\ a_1\ s'_D\}\}\ \{\{s_D\ a_1\ s'_E\}\}$
$\{\{s_E\ a_0\ s'_D\}\}\ \{\{s_D\ a_0\ s'_C\}\}\ \{\{s_C\ a_1\ s'_D\}\}\ \{\{s_D\ a_1\ s'_E\}\}\ \{\{s_E\ a_0\ s'_D\}\}$
$\{\{s_D\ a_0\ s'_C\}\}\ \{\{s_C\ a_0\ s'_B\}\}\ \{\{s_B\ a_1\ s'_C\}\}\ \{\{s_C\ a_0\ s'_B\}\}\ \{\{s_B\ a_1\ s'_C\}\}$
$\{\{s_C\ a_1\ s'_D\}\}\ \{\{s_D\ a_1\ s'_E\}\}\ \{\{s_E\ a_1\ s_{T_R}, r=1\}\}$


T=9 / Final Tree
C
n=8
0
n=2
r=0
1
n=6
r=5
B
n=2
D
n=6
0
n=1
r=0
0
n=4
r=3
1
n=1
r=0
A
n=1
C
n=4
E
n=1
0
n=1
r=1
1
n=2
r=2
B
n=1
D
n=3
0
n=1
r=1
C
n=1