Garrett Partenza
Advanced Reinforcement Learning
March 20, 2022

**Exercise 5**

**Question 1.** (a) Do you think one of the multi-step bootstrapping methods from Chapter 7 could do as well as the Dyna method? Explain why or why not. (b) What are the advantages and disadvantages of using n-step returns in the planning phase?

(a) I do believe that one of the multi-step bootstrapping methods would do well in this environment, however, I do not believe it would do better than Dyna since Dyna updates q-values more frequently.
(b) N-step returns can be implemented for the planning phase, where advantages include faster propagation of the reward function and disadvantages include increased computational costs per outer-iteration.

**Question 2.** Careful inspection of Figure 8.5 reveals that the difference between Dyna-Q+ and Dyna-Q narrowed slightly over the first part of the experiment. What is the reason for this?

Dyna-Q+ offers a reward bonus for exploring under visited states as to ensure the model of the environment has converged to the correct values. This phenomenon is a result of that desire to explore, where Dyna-Q does not experience this.

**Question 3.** (a) For some unknown reason, the textbook is quite vague about Dyna-Q+ and does not provide pseudo-code for the modification. Read Section 8.3 carefully (specifically p. 168) and reconstruct the pseudocode for Dyna-Q+. Look carefully at the footnote1 on p. 168 as well.

(a) The new psuedocode is as follows.
(1) Initialize $Q(s, a)$ for all $S$ and $A$
(2) **NEW**: Initialize $M(s, a)$ as $(s, 0)$ for all $S$ and $A$
(3) **NEW**: initialize $k$ and $T(s, a)$ for all $S$ and $A$
(4) Loop Forever:
(5) $S \longleftarrow$ current non-terminal state
(6) $A \longleftarrow$ epsilon-greedy choice given $S$ and $Q$
(7) Take action $A$ and observe reward $R$ and new state $S$‘
(8) **NEW**: $T(s, a) =$ current time
(9) $M(s, a) \longleftarrow R$, $S$‘
(10) Loop n-times:
(11) $S \longrightarrow$ random previously observed state
(12) **NEW**: $A \longleftarrow$ random action
(13) $R$, $S$‘ $\longrightarrow M(s, a)$
(14) **NEW**: $R+ = k \cdot \sqrt{\text{current time} - T(s, a)}$
(15) Update Q-Value
(b) *See figures below.*
(c) *See figures below.* In terms of the footer vs no footer trial, we see that the implementation with footer does much better than without due to the footer implementation encouraging exploration by being able to simulate or model un-visited states.

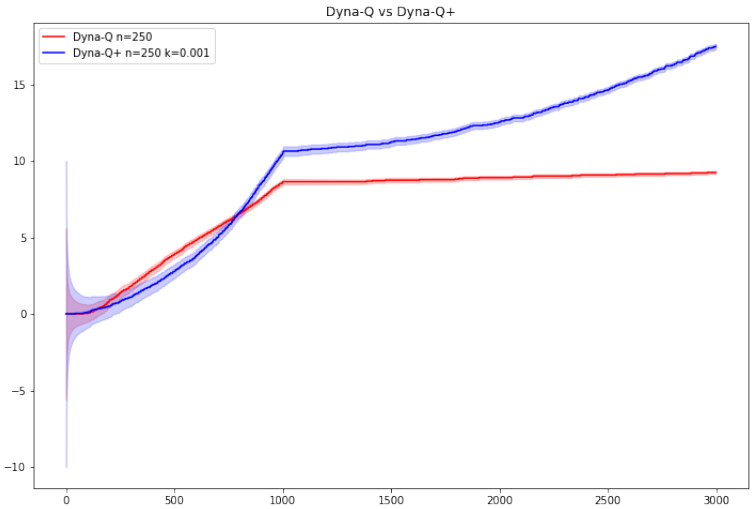



FIGURE 1. Dyna-Q vs Dyna-Q+ (k=0.001, n=250, $\epsilon = 0.1$, lr=0.1, $\gamma = 0.95$)

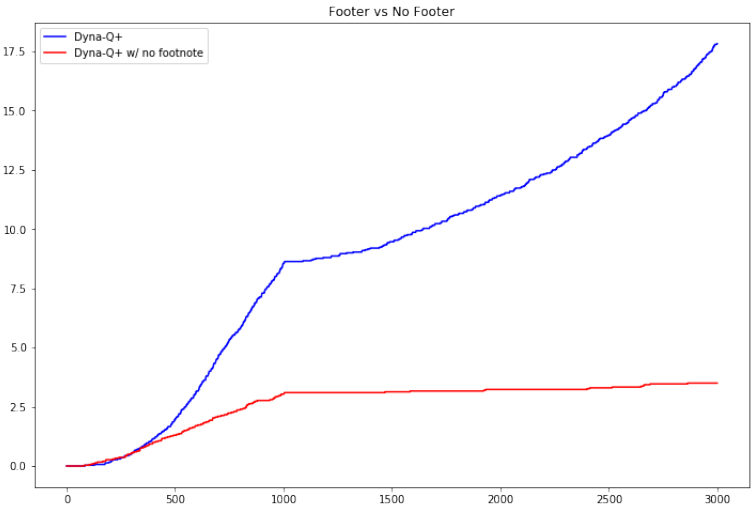



FIGURE 2. Footer vs No Footer (k=0.001, n=250, $\epsilon = 0.1$, lr=0.1, $\gamma = 0.95$)

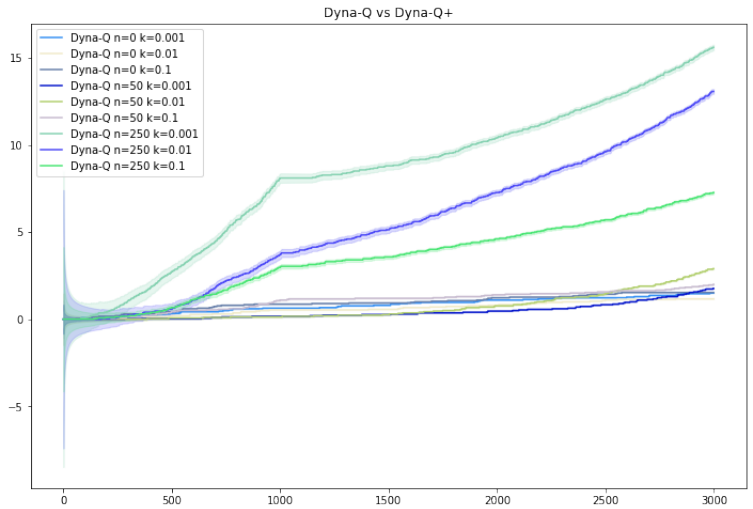



FIGURE 3. Comparison of different values for n and k

**Question 4.** (a) What would generally happen under each approach? How do they differ? (b) What are the advantages and disadvantages of each approach?

(a) Generally under each approach exploration would be greater than other learning methods mentioned in the book, except UCB does this using upper confidence bounds and Dyna-Q+ does this using a psuedo-reward function directly to q-values. One of the benefits of using UCB is that Q values are not influenced by the psuedo-reward, and thus the q-values are empirically correct based on your model of the world. This not true for Dyna-Q+ since q-values are offset by some amount considering the exploration bonus. One of the disadvantages of this is that in very large state spaces, many state-action pairs will be continuously under-explored for long periods of time and may cause arbitrarily large bonuses to be applied q-values.

**Question 5.** (a) How might the tabular Dyna-Q algorithm be modified to handle stochastic environments? Would this modification perform poorly on changing environments such as considered in Section 8.3? How could the algorithm be modified to handle stochastic environments and changing environments?

(a) To handle stochastic environments, we can keep track of the rewards and next states for our model, and when we do the planning phase, select the reward and next state with their respective probabilities. This would not perform well in a changing environment due to previous samples that are no longer relevant influencing our planning phase. To correct this we can weight heavier more recently experienced transitions.

(b) The plot below contains the cumulative reward for the original, stochastic, and stochastic+changing Dyna-Q algorithms. The stochastic implementation assigns probabilities to transitions based on past occurrences, and the changing algorithm applies an exponential recency weight the set of previous 500 observed occurrences. The stochastic implementation definitely improves the performance in changing environments. However, I could not get the changing implementation to perform well by weighting more recently encountered states. I believe this may be a result of my weighting methodology.

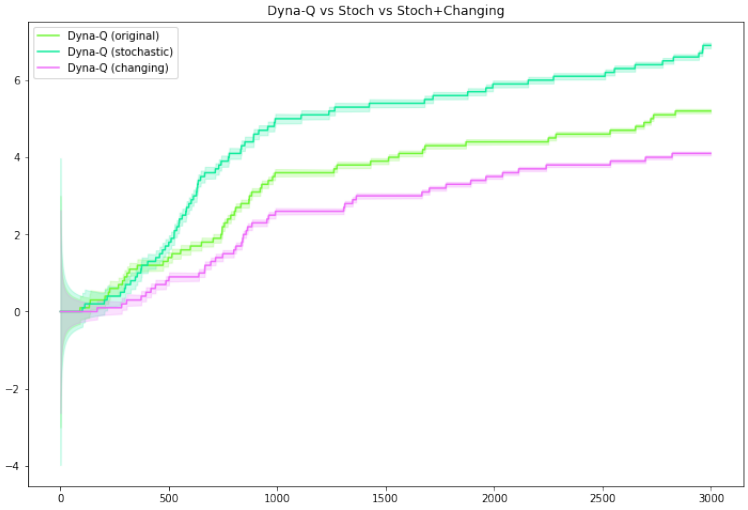



FIGURE 4. Dyna-Q vs. Dyna-Q+Stoch vs. Dyna-Q+Stoch+Change

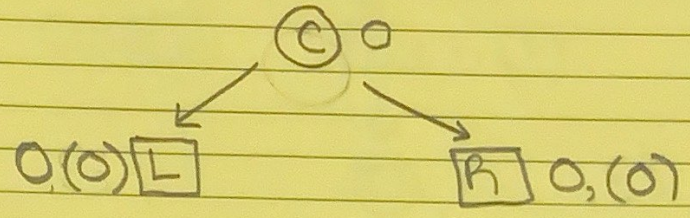
**Question 6.** Manual MCTS

Drawings are on the next 9 pages of this homework submission.

/ /

# Monte Carlo Tree Search

## Iteration N1



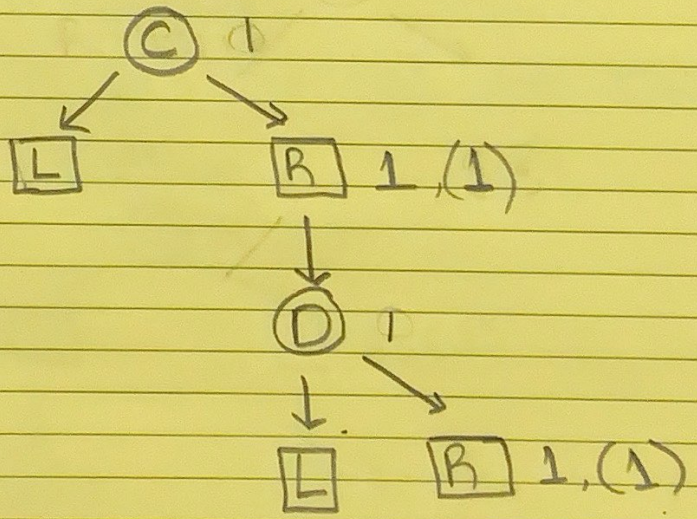

* selection/expansion

C → D *

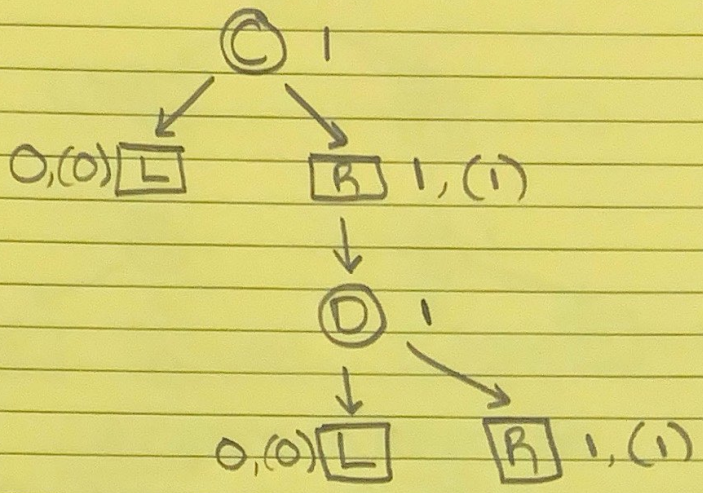
* Rollout

C → D → E → T + 1

* Backup



Iteration N2



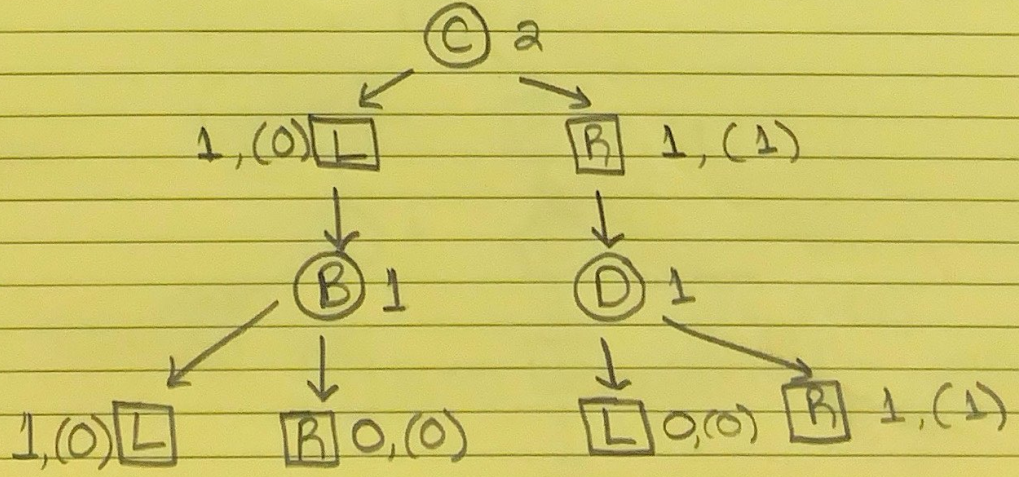

* Selection / expansion

$$C \rightarrow \left\{ \frac{0}{0} + \sqrt{\frac{2\ln(1)}{0}},\ \frac{1}{1} + \sqrt{\frac{2\ln(1)}{1}} \right\} \rightarrow B *$$

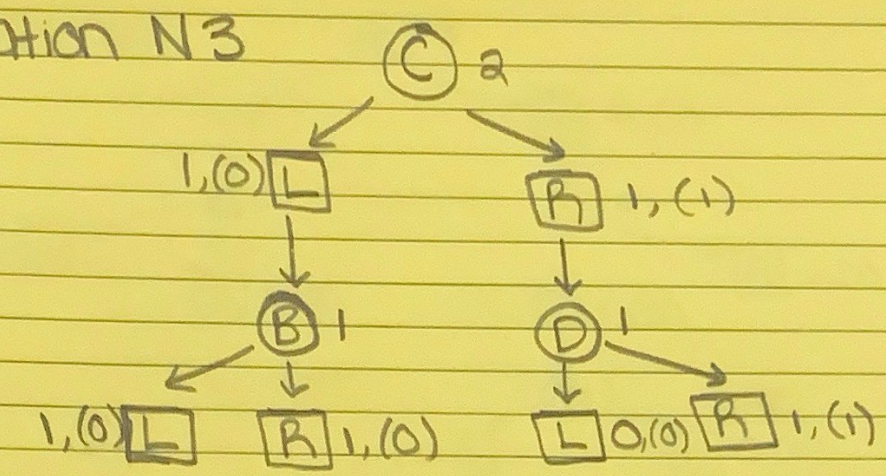
* Rollout

C → B → A → T+O

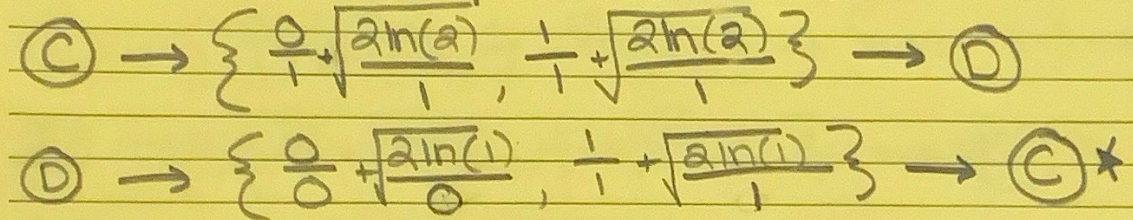
* Backup



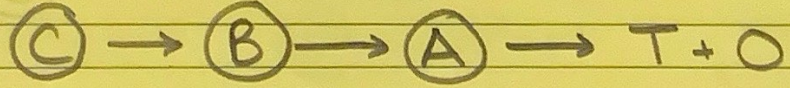
Iteration N3



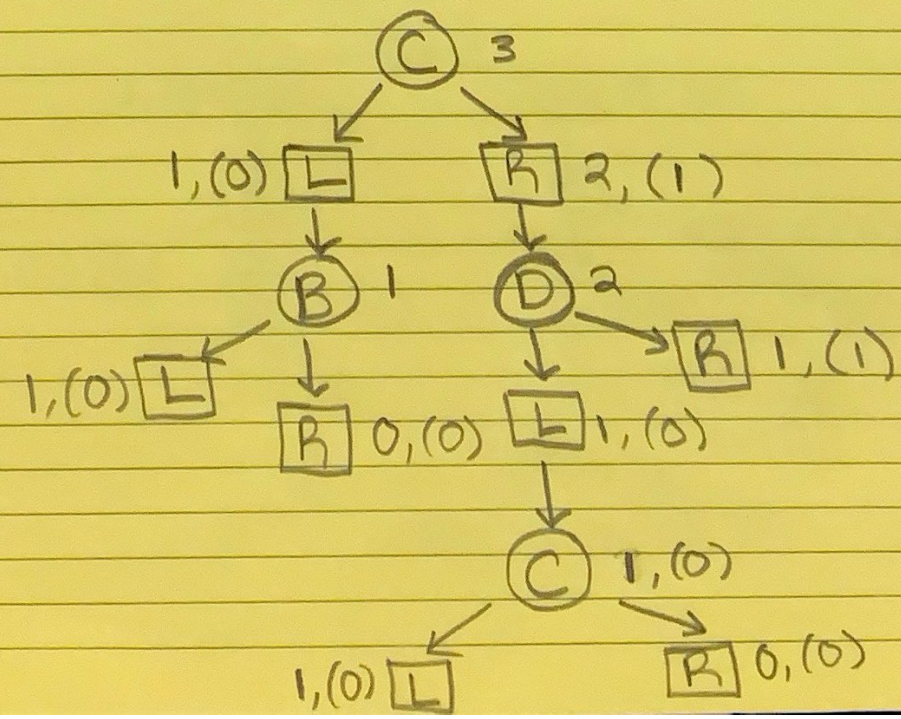

* expansion / selection



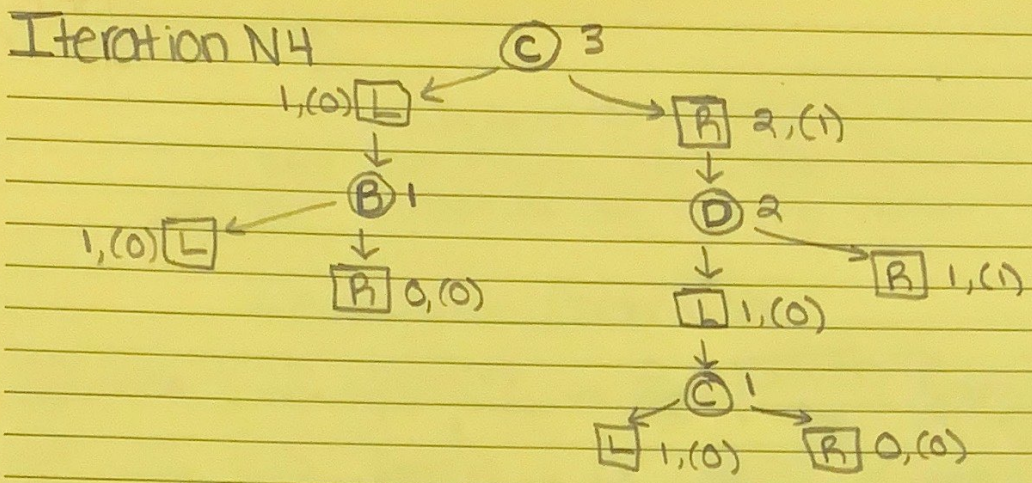

* Rollout




* Backup



Iteration N4




* expansion / selection

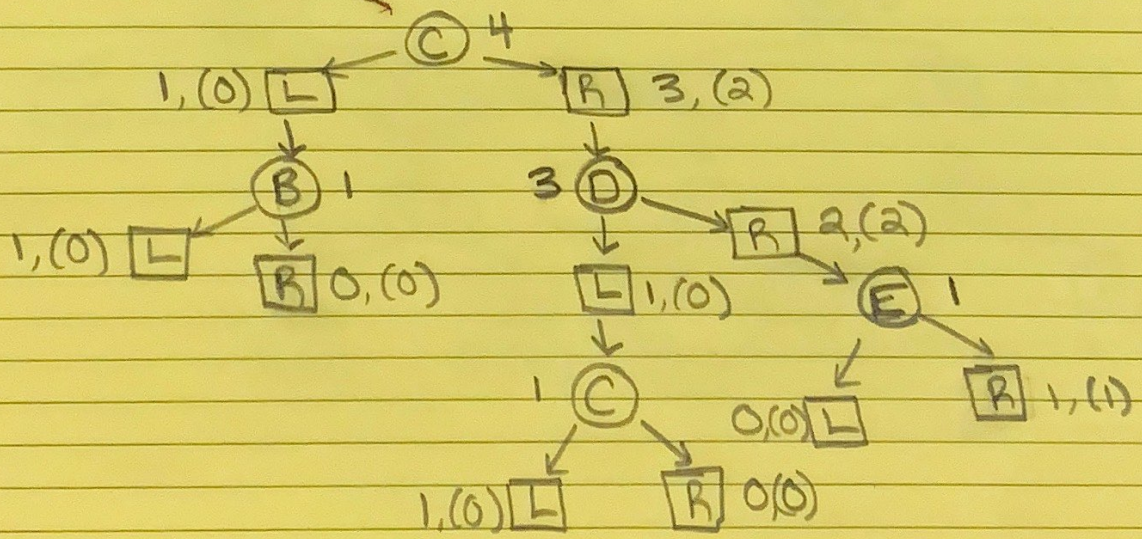
C → $\left\{ \frac{0}{1} + \sqrt{\frac{2\ln(3)}{1}}, \frac{1}{2} + \sqrt{\frac{2\ln(3)}{2}} \right\}$ → {1.48, 1.54} → D

D → $\left\{ \frac{0}{1} + \sqrt{\frac{2\ln(2)}{1}}, \frac{1}{1} + \sqrt{\frac{2\ln(2)}{1}} \right\}$ → E *

* Rollout

E → T+1

* Backup



# Iteration N5

* Tree is getting too big to carry over last result... please just refer to the bottom of the last pg. Thanks!

* expansion / selection

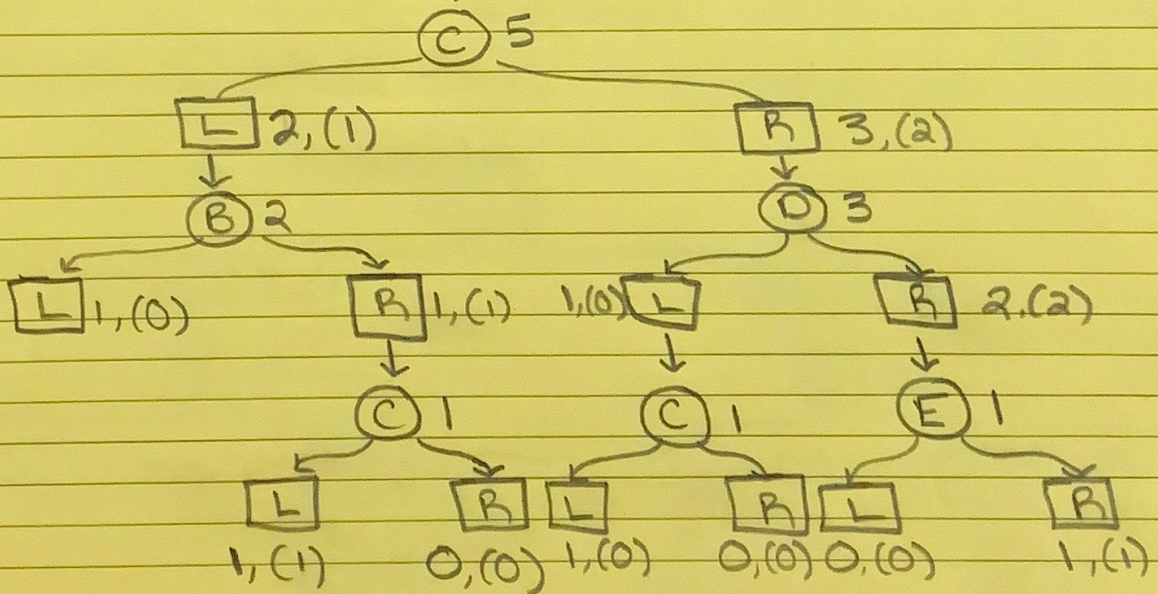
Ⓒ → $\left\{\frac{0}{1} + \sqrt{\frac{2\ln(1)}{1}} \text{ vs } \frac{2}{3} + \sqrt{\frac{2\ln(3)}{3}}\right\}$ → {2.77 vs 1.39} → Ⓑ

Ⓑ → $\left\{\frac{0}{1} + \frac{2\ln(1)}{1} \text{ vs } \frac{0}{0} + \sqrt{\frac{2\ln(1)}{0}}\right\}$ → Ⓒ *

* Rollout

Ⓒ → Ⓑ → Ⓒ → Ⓓ → Ⓔ → T+1

* Backup



# Iteration N6

* Tree is getting too big to carry over last result. Please just refer to the last page. Thanks!

* expansion/selection

Ⓒ → $\{\frac{1}{2}+\sqrt{\frac{2\ln(5)}{2}}$ vs $\frac{2}{3}+\sqrt{\frac{2\ln(5)}{3}}\}$ → {1.76 vs 1.70} → Ⓑ

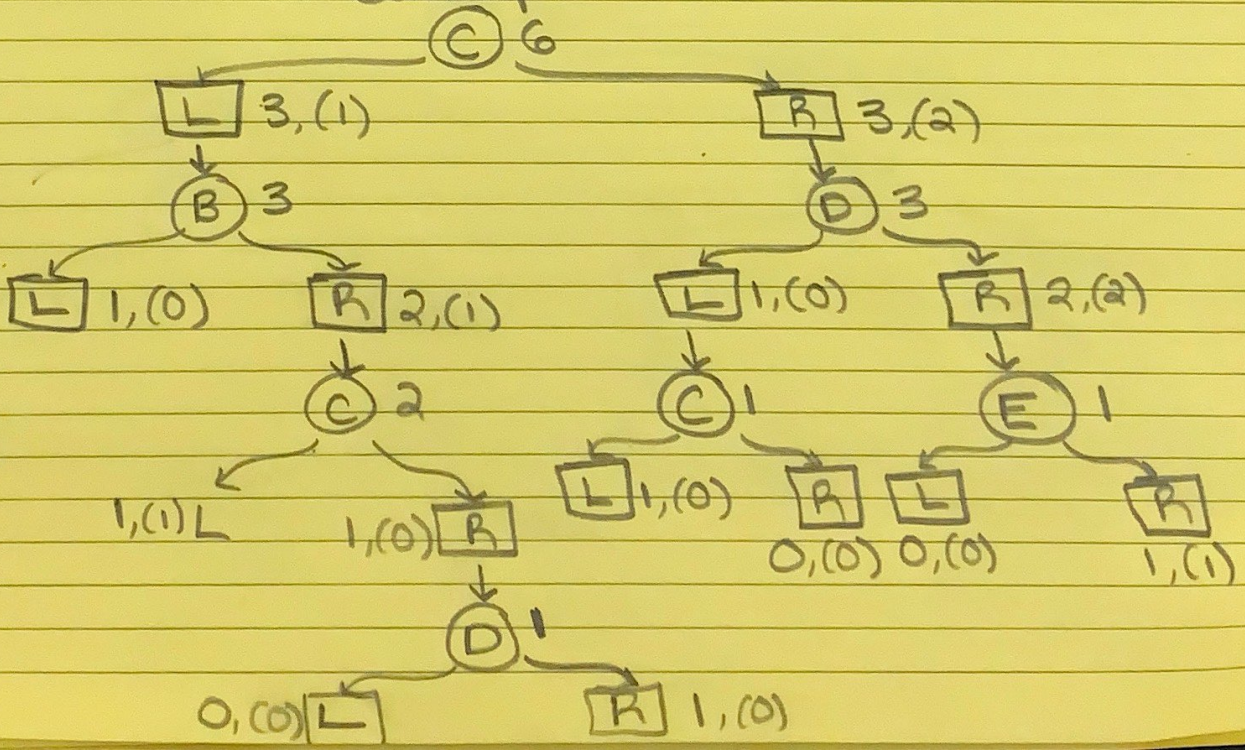
Ⓑ → $\{\frac{0}{1}+\sqrt{\frac{2\ln(2)}{1}}$ vs $\frac{1}{1}+\sqrt{\frac{2\ln(2)}{1}}\}$ → Ⓒ

Ⓒ → $\{\frac{1}{1}+\sqrt{\frac{2\ln(1)}{1}}$ vs $\frac{0}{0}+\sqrt{\frac{2\ln(1)}{0}}\}$ → Ⓓ *

* Rollout

D → E → D → C → D → C → D →
E → D → C → B → C → D → C →
B → C → D → C → B → A → T+0

* Backup



# Iteration N7

* Tree is getting too big to carry over to next page. Please just refer to previous page. Thanks!

* expansion / selection

C → $\frac{1}{3}+\sqrt{\frac{2\ln(6)}{3}}$ vs $\frac{0}{3}+\sqrt{\frac{2\ln(6)}{3}}$ → 1.42 vs 1.70 → D

D → $\frac{0}{1}+\sqrt{\frac{2\ln(3)}{1}}$ vs $\frac{0}{2}+\sqrt{\frac{2\ln(3)}{2}}$ → 1.48 vs 2.04 → E

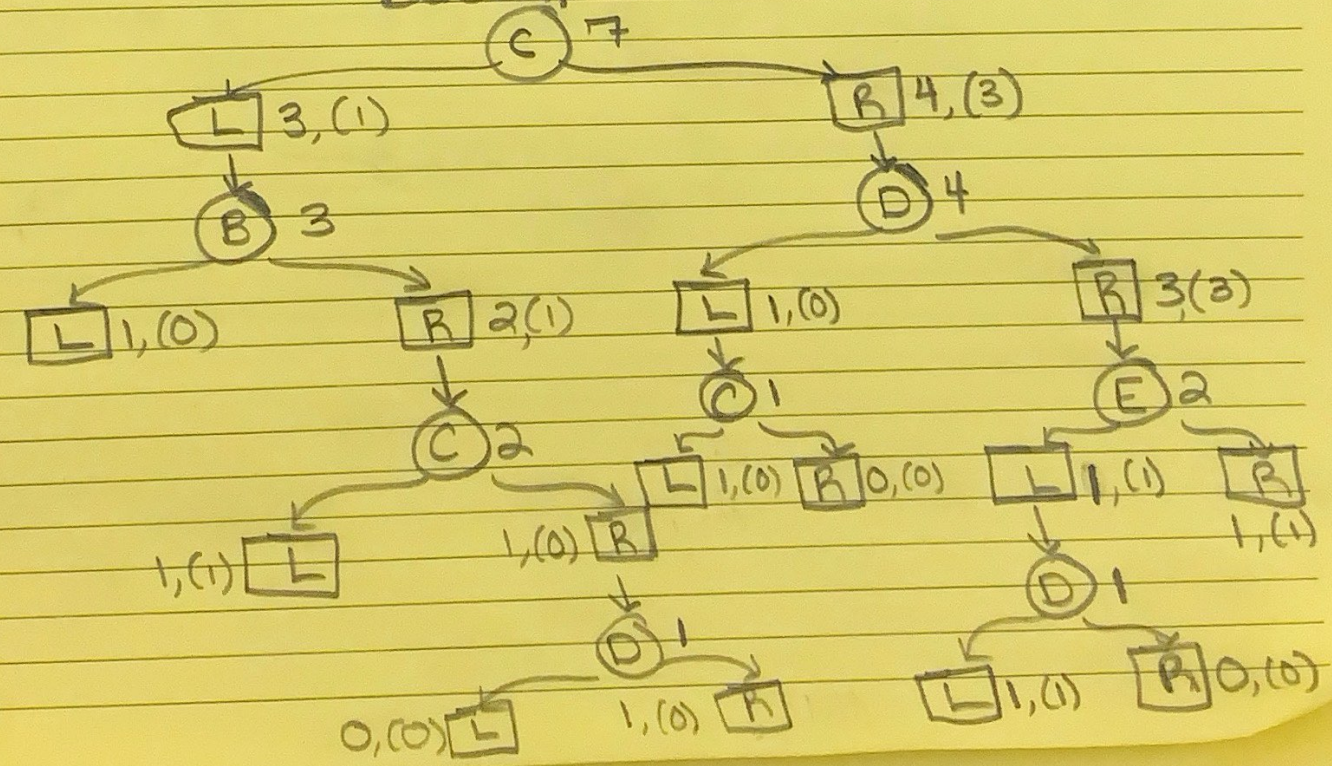
E → $\frac{0}{0}+\sqrt{\frac{2\ln(1)}{0}}$ vs $\frac{1}{1}+\sqrt{\frac{2\ln(1)}{1}}$ → D *

* Rollout

D → C → B → L → D → C → D →

E → D → E → T+1

* Backup



Iteration N8 (thank god)

* Tree is getting too big to carry over. Please just refer to the previous page. Thanks!

* expansion / selection

(C) → $\left\{\frac{1}{3}+\sqrt{\frac{2\ln(7)}{3}} \text{ vs } \frac{3}{4}+\sqrt{\frac{2\ln(7)}{4}}\right\}$ → $\{1.47 \text{ vs } 1.73\}$ → (D)

(D) → $\left\{\frac{0}{1}+\sqrt{\frac{2\ln(4)}{1}} \text{ vs } \frac{3}{3}+\sqrt{\frac{2\ln(4)}{3}}\right\}$ → $\{1.66 \text{ vs } 1.96\}$ → (E)

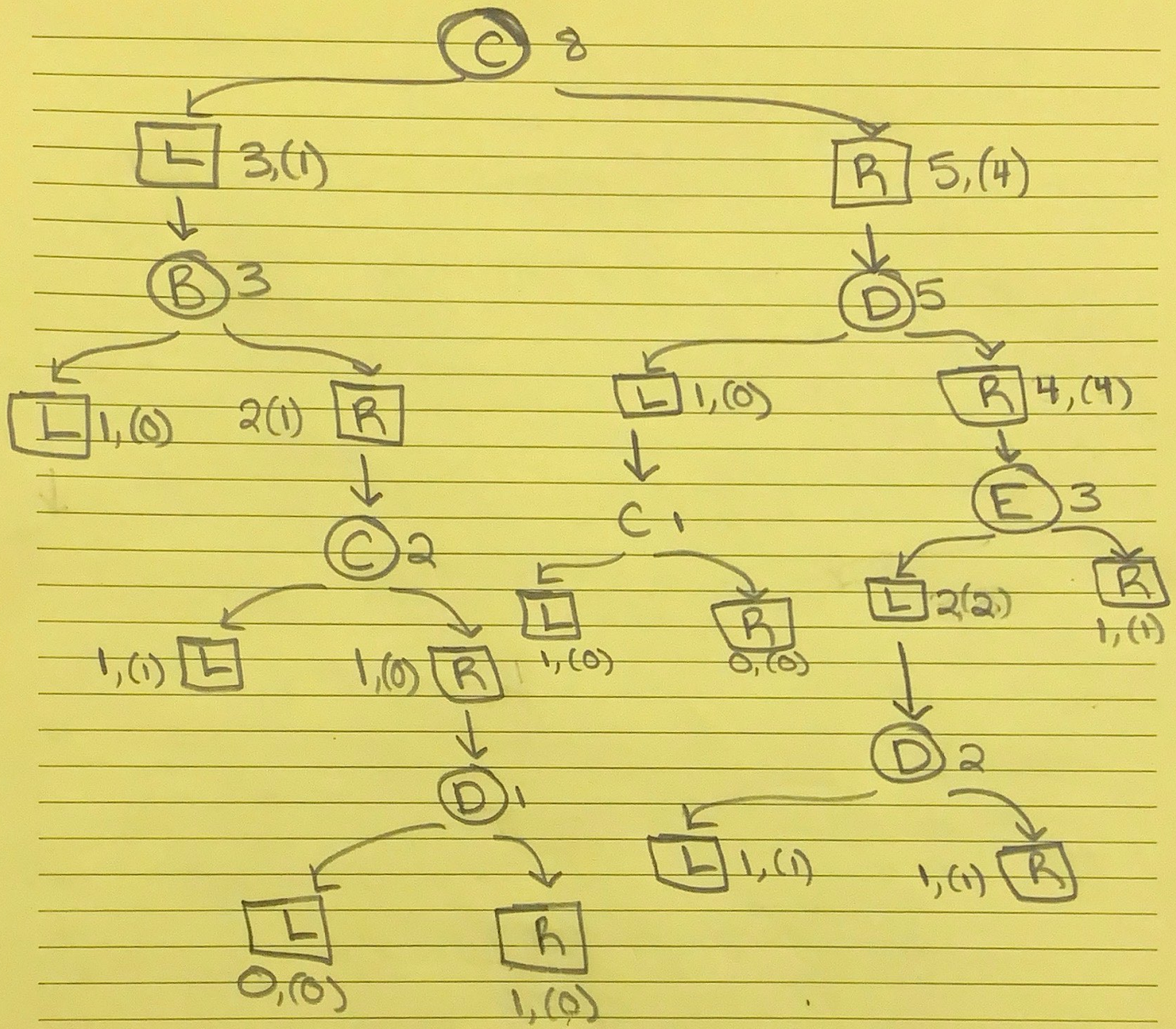
(E) → $\left\{\frac{1}{1}+\sqrt{\frac{2\ln(2)}{1}} \text{ vs } \frac{1}{1}+\frac{2\ln(2)}{1}\right\}$ → random → (D)

(D) → $\left\{\frac{1}{1}+\sqrt{\frac{2\ln(1)}{1}} \text{ vs } \frac{0}{0}+\frac{2\ln(1)}{0}\right\}$ → (E) *

* Rollout

E → D → E → D → C → B → C →
D → E → D → C → D → E → T+1

* Backup on next Page.


C 8
L 3,(1)
R 5,(4)
B 3
D 5
L 1,(0)
2(1) R
L 1,(0)
R 4,(4)
C 2
C 1
E 3
1,(1) L
1,(0) R
L 1,(0)
R 0,(0)
L 2(2)
R 1,(1)
D 1
D 2
L 0,(0)
R 1,(0)
L 1,(1)
1,(1) R