1.
(a) Generally, TD methods are often better than MC because they are bootstrapping. TD don't wait for the final outcome to compute and optimize. Driving home from a new building, only the path to highway entrance ramp is changed. In this case, TD updates will only look one step ahead to optimize the policy, which still utilized the experience it got before. On contrast, MC couldn't learn until it complete the trip. That is less efficient than TD methods.

However, it won't happen in the original scenario because it doesn't have any prior experience. In this case, MC experience all first and updates together within a episode, while TD updates will require more episodes to learn for all states and bootstrap.

(b) MC advantage is episodic learning, while TD advantage is bootstrapping. Therefore, MC methods are better than TD in tasks that are episodic without immediate rewards. For example, Texas hold 'em and Blackjack are board games where the agent only get rewards from final outcome. So learning is only meaningful at the end of each episode. In this case, MC approach is better than TD.

2.
(a) Q-learning updates its Q-values regardless of the policy currently being followed.

(b) No. SARSA is an on-policy algorithm but Q-learning is an off-policy algorithm. Looking at their pseudocode, SARSA choose a' and s' and then updates the Q-function; while Q-learning first updates the Q-function, and the next action to perform is selected in the next iteration, derived from the updated Q-function and not necessarily equal to the a' selected to update Q.

3.
(a) It tells us the first episode terminate on the extreme left. According to Eq(6.2), we know the estimate value of a state is affected by the next state. So only V(A) is changed, whose the next state in 1st episode is a terminate state.

$$V(A) = 0.5 + 0.1*[0 + 0 – 0.5] = 0.45$$

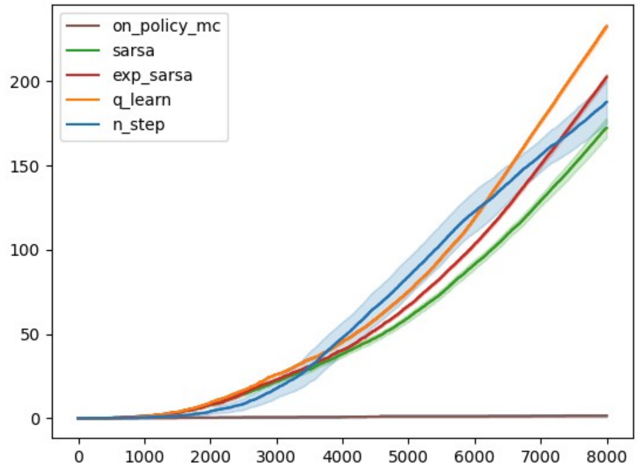
(b) As we learn from the right graph, TD methods have best performance when alpha = 0.05 and MC perform best when alpha = 0.01. TD's α < MC's α. I don't think wider range of alpha will perform better, because higher values of alpha will result in bigger error.
Smaller alpha will more likely to converge to the optimal value. Both alphas are already small, so I don't think that fixed value of alpha would performed significantly better.

(c) It is caused by given a high α, as it given to the TD error will exaggerate small errors. This doesn't always occurs. It might be affected by the initial state value because the state-values are calculated on algorithm basis, which have bias.

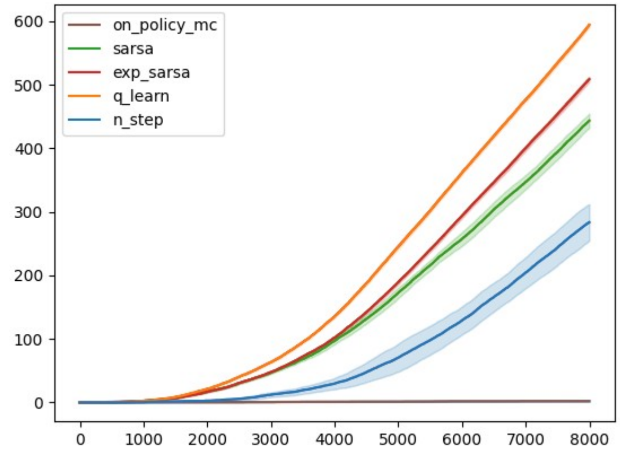
(d) Differences between various n and α would be smaller for 5 states than for 19 states. In Example 7.1, we want to exaggerate the differences so we choose a larger random walk task.
A smaller walk would shifted the advantage to a different value of n, e.g., 5 states task cause better value for n smaller than n=4.

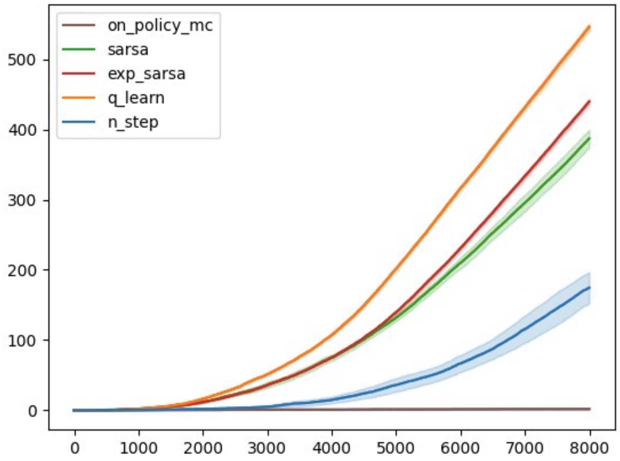
I don't think changing in left-side outcome from 0 to -1 make any difference in the best value of n because only the right-side terminate state have a positive reward. But perhaps this speeds up the learning efficiency for n-step TD.

4.
(b)



(c)



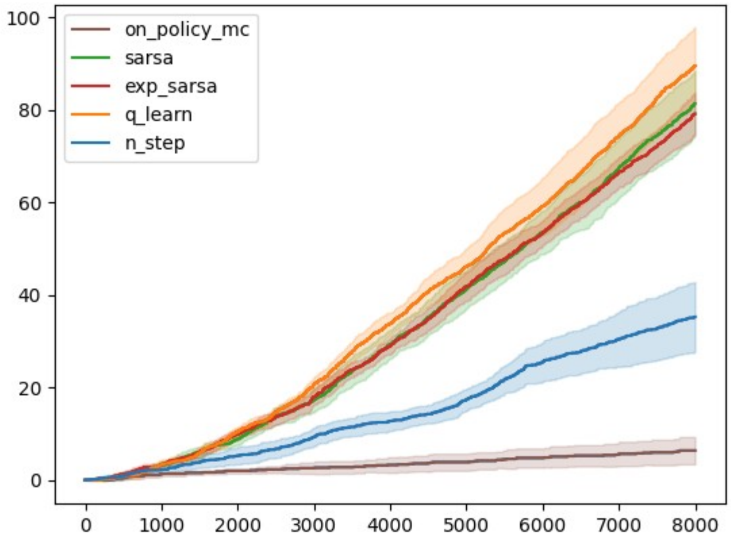


King Move

King + Ninth Movement

After converting to King’s move, all the methods got a better performance. According to the number of episodes they completed, most of the methods achieved twice the completion rate compared to their performance in four-move task. Adding a ninth action won’t be better than King’s move task.

(d)
Changing the wind to be stochastic resulted in an increase in variance and a drop in performance across all methods.



5.
(a) Plots are in .ipynb file

(b) Monte-Carlo methods are known for having low bias because they do not bootstrap. TD(0) method typically has higher bias because it uses the estimated value of the subsequent state to update the value of the current state. N-step TD methods strike a balance between Monte-Carlo and TD(0) by considering N future steps when updating value estimates.

As the training amount N increases, MC methods have higher variance. Looking at MC methods' histograms, we know the true value is around -20. The state-value given by TD(0) is -15 so, from here, we can also tell that TD(0) have higher bias. As N increases from 20 to 50, the bias is increased for both TD(0) and n-step TD.