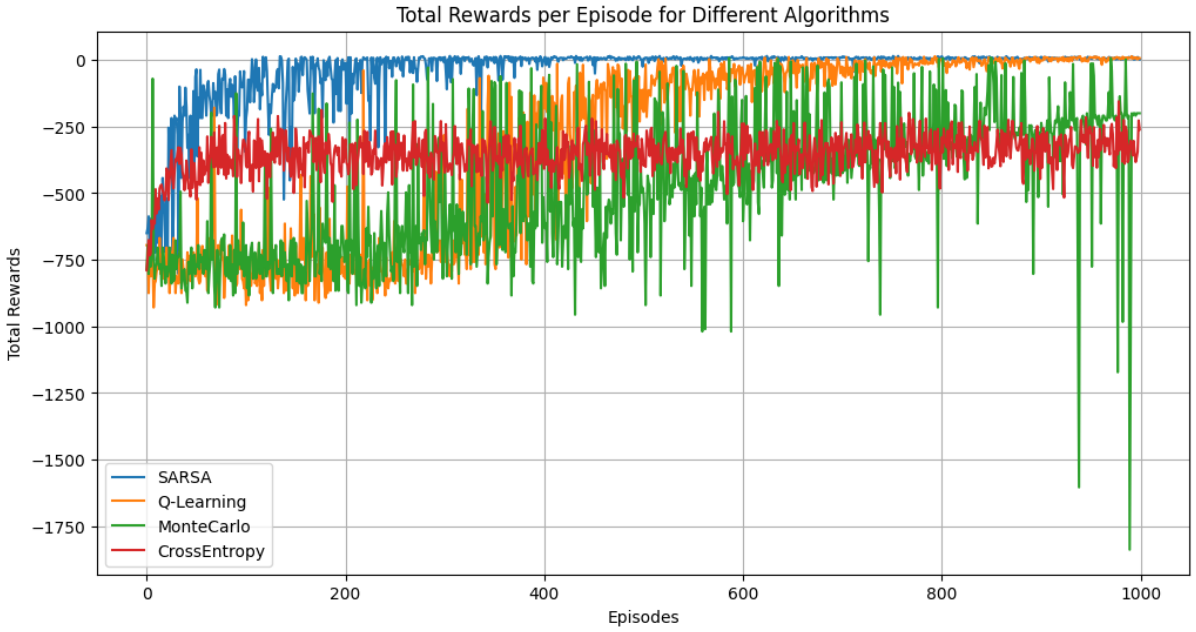
Домашняя работа #4 RL

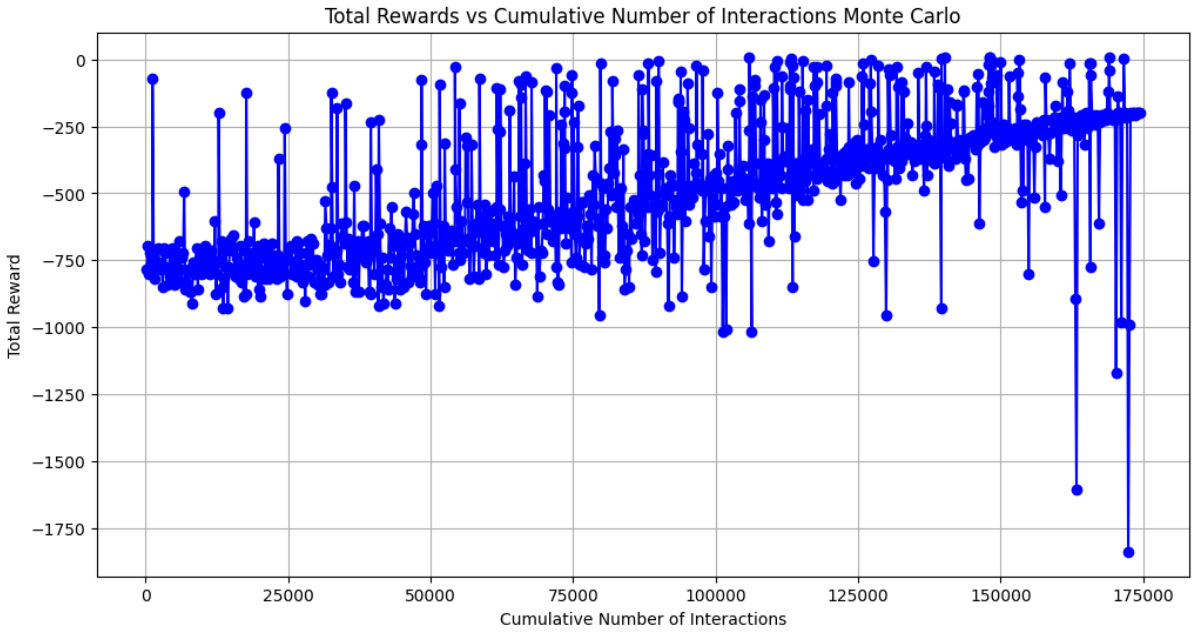
Первое задание

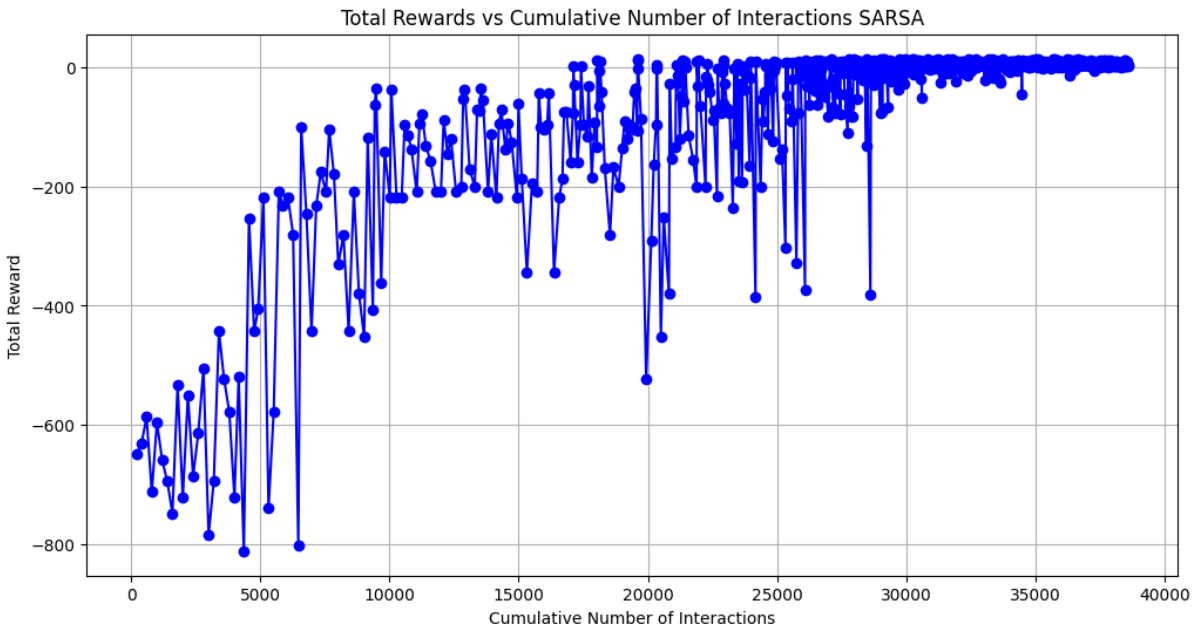
Обучение

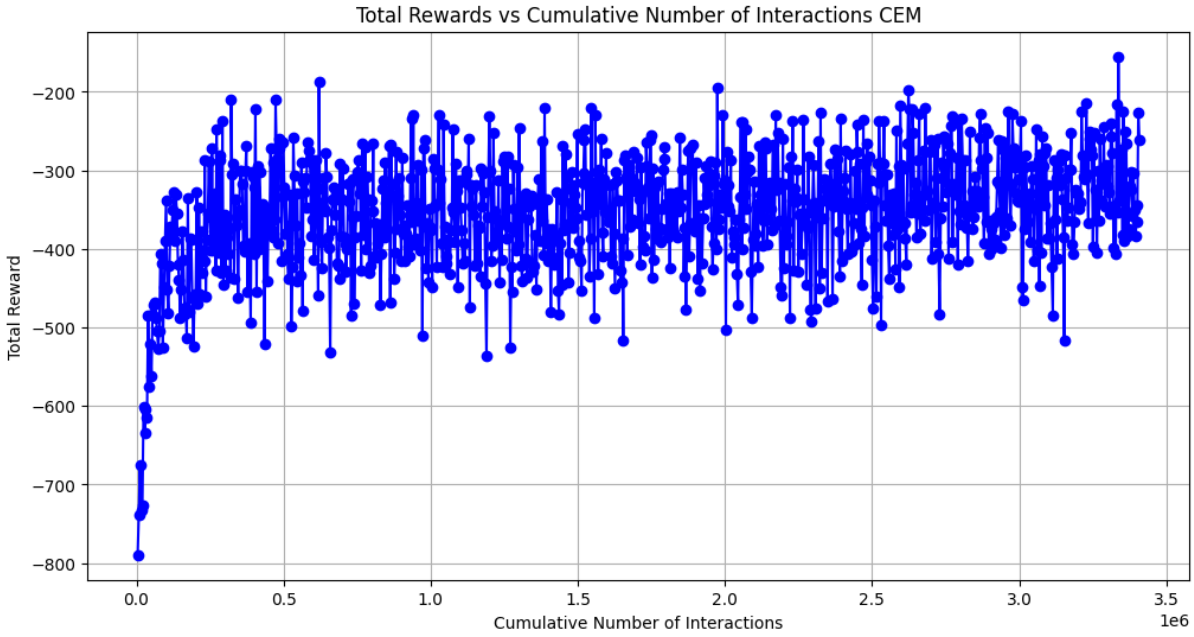
Эксперимент 1

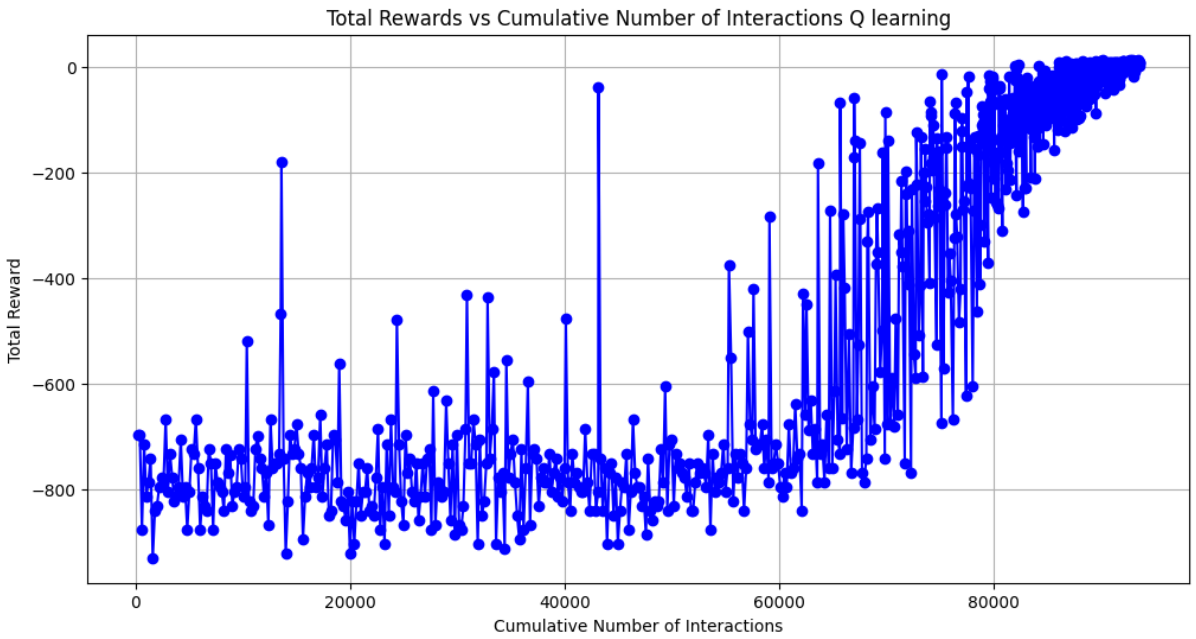
Был написан код для Cross-entropy и q learning. Кросс-энтропию пришлось переписать с нуля, потому что предыдущий код долго и мучительно отказывался работать.

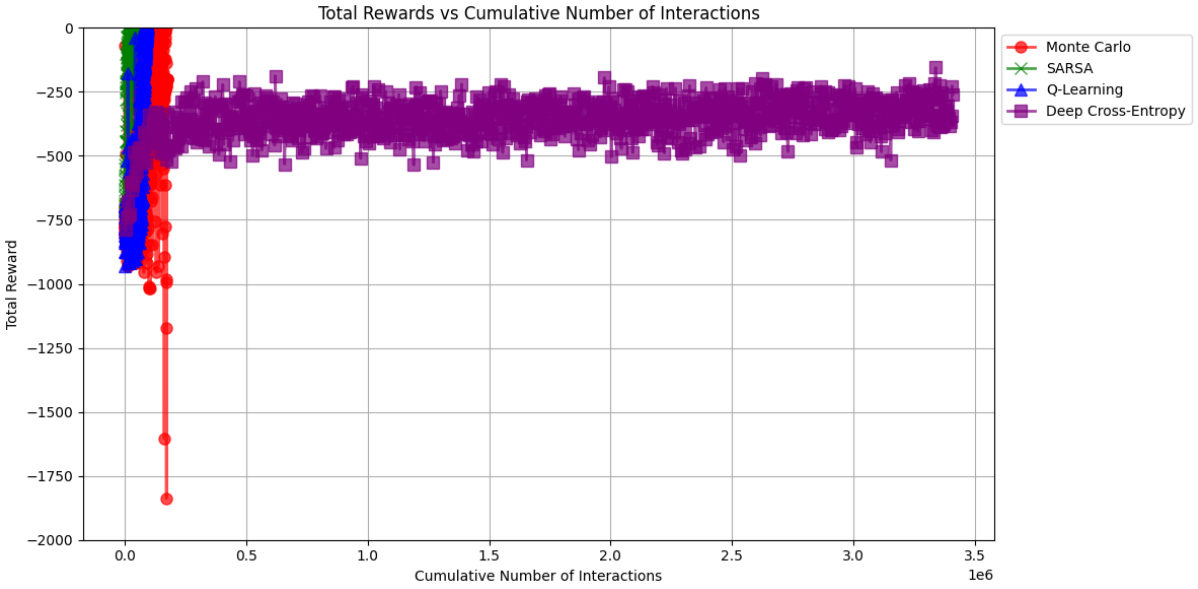





Total Rewards vs Cumulative Number of Interactions SARSA
Total Reward
0
−200
−400
−600
−800
0
5000
10000
15000
20000
25000
30000
35000
40000
Cumulative Number of Interactions


Total Rewards vs Cumulative Number of Interactions CEM
Total Reward
−200
−300
−400
−500
−600
−700
−800
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1e6
Cumulative Number of Interactions





Вывод

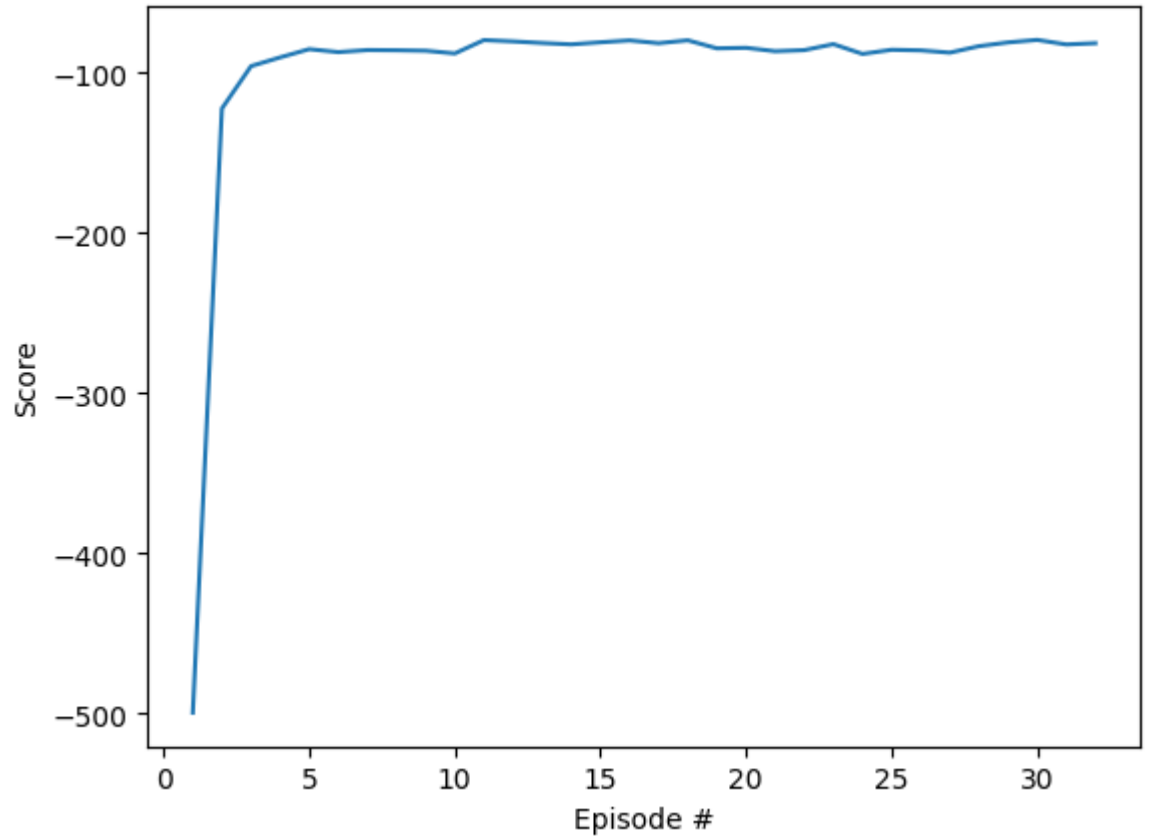
Быстрее и стабильнее всего обучились Sarsa, и Q-learning. Кросс энтропия вообще отказалась нормально обучаться.

Второе задание

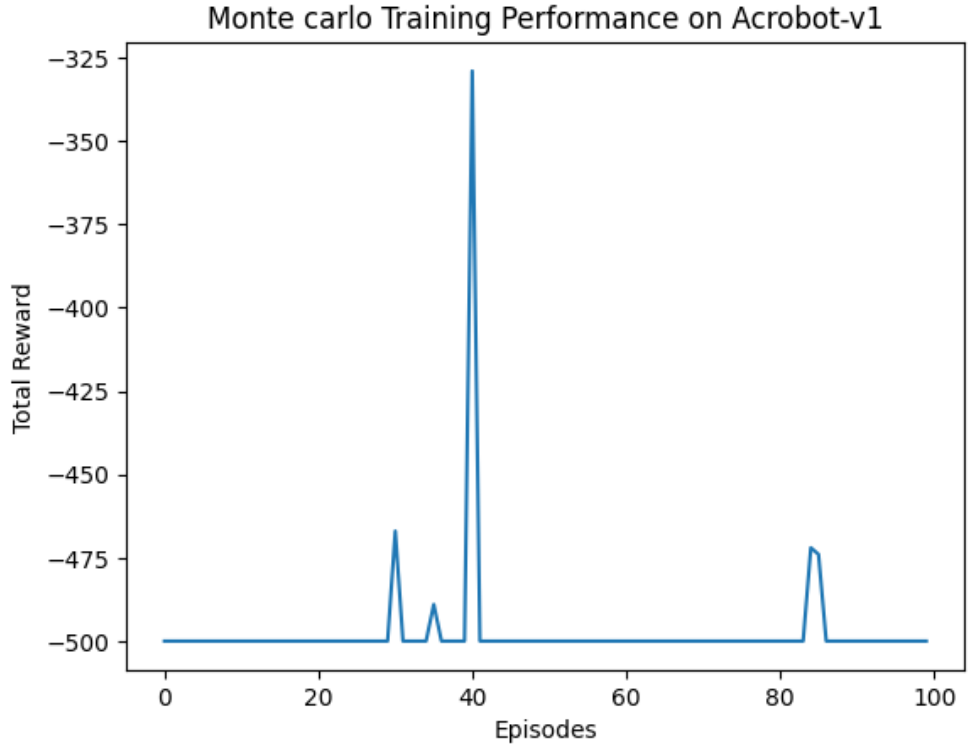
Обучение

Эксперимент 1

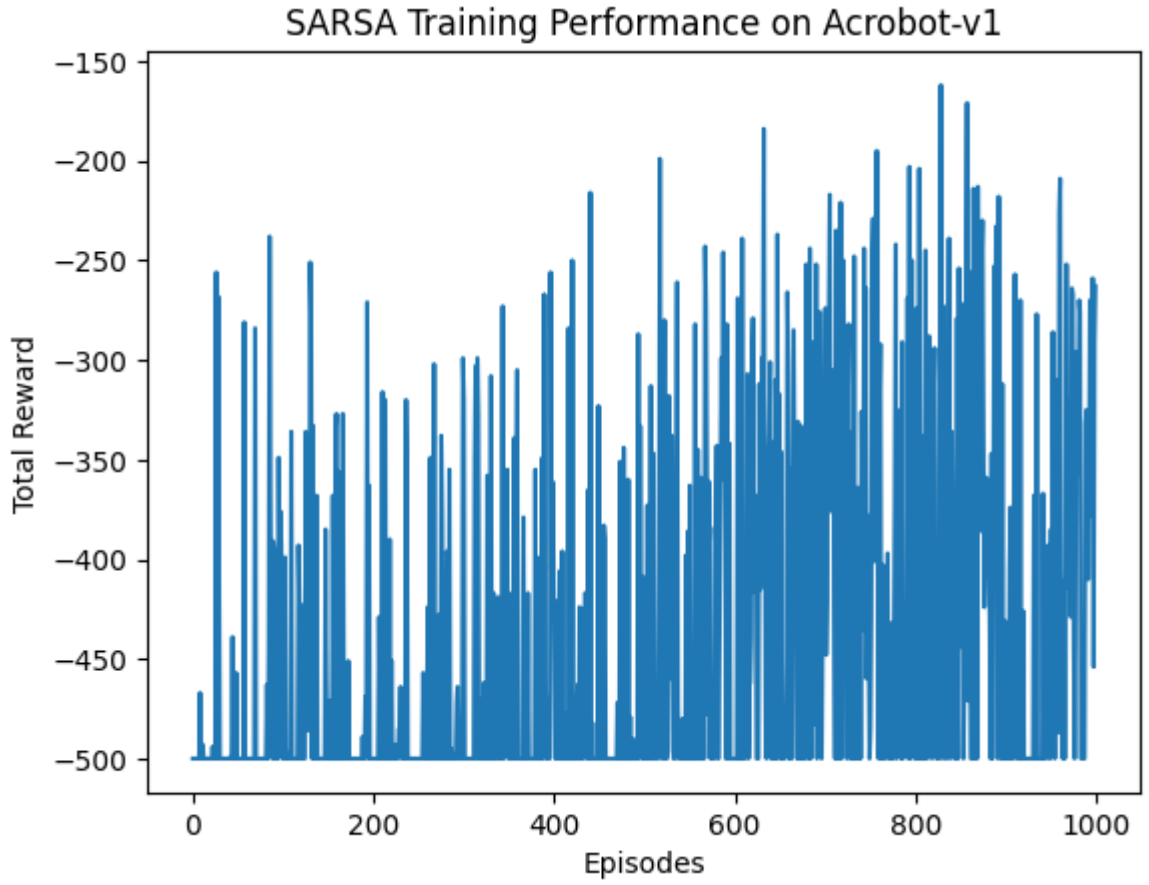
Результат обучения кросс энтропии. Она очень быстро обучилась до неплохих для Acrobot значений



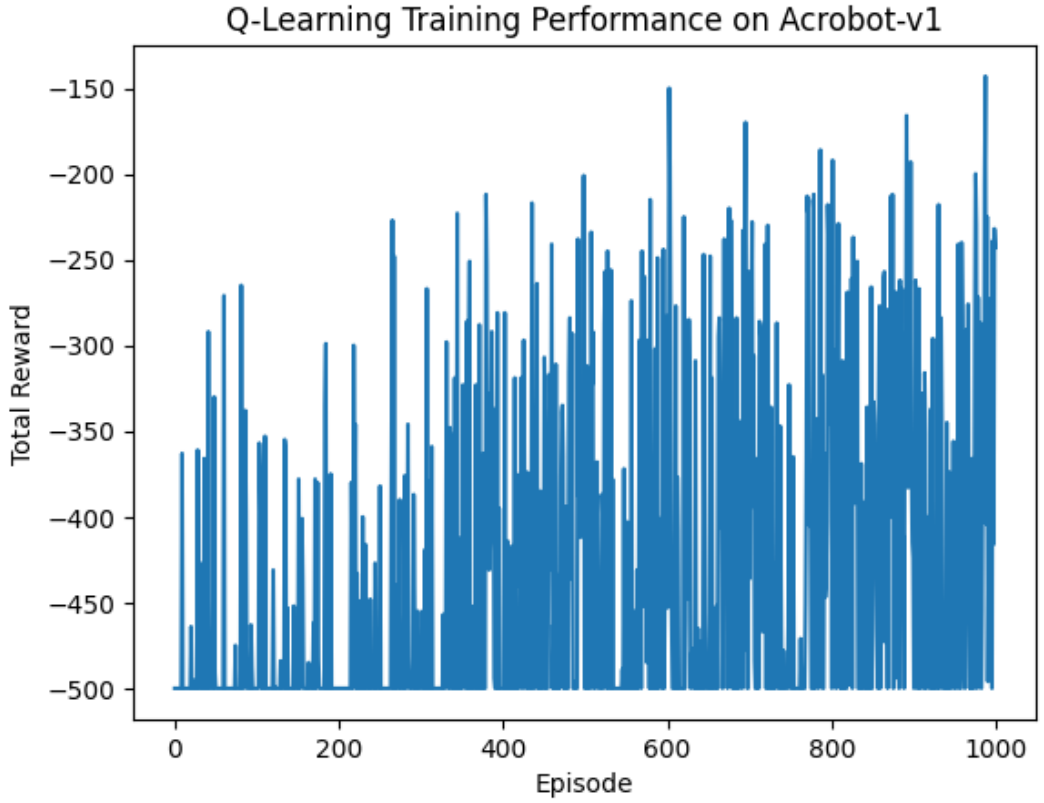
Монте карло просто отказалась обучаться:



Сарса оказался получше, но тоже значительно хуже кросс энтропии.



Более менее обучился q learning



Вывод

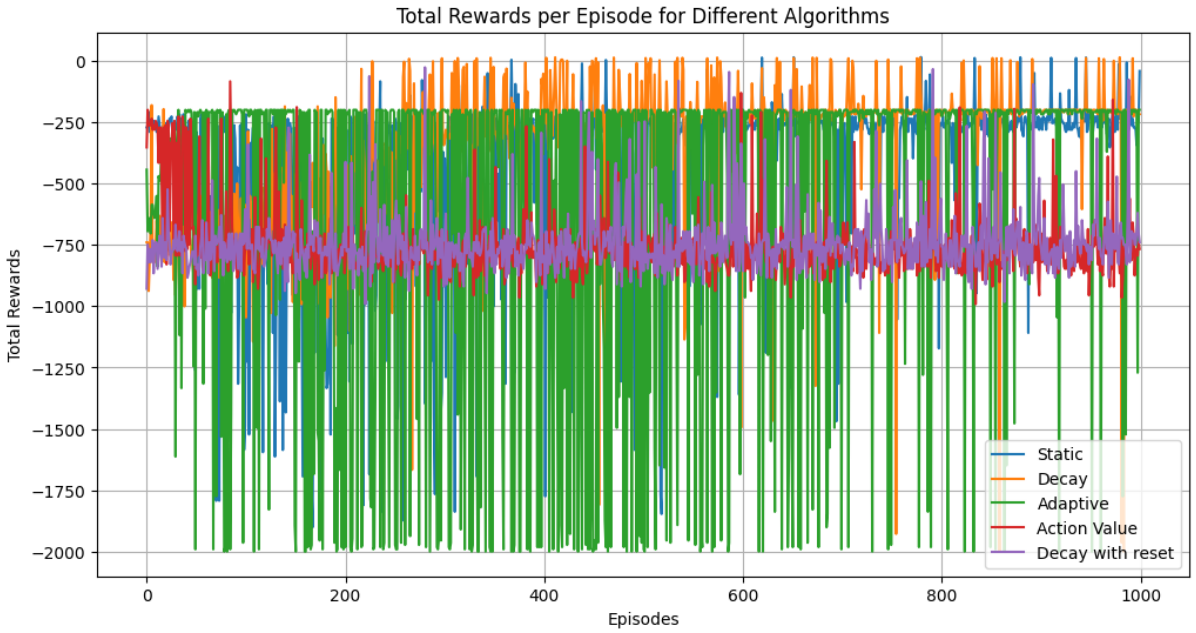
На этой среде montecarlo, qlearning, и sarsa работают значительно хуже кроссэнтропии. Как я понял это происходит из-за задержки награды у акробота. Из-за этого этим алгоритмам сложно оценить и улучшить свои политики. Кросс энтропия же перебирает большое количество шумных траекторий и учится уже на них.

Третье задание

Обучение

Эксперимент 1

Написал разные способы выбора epsilon для монтекарло. Проверил на задаче taxi-v3. Лучше всего себе показали static и decay.



Вывод

Не могу точно сказать причину почему static и decay оказались значительно лучше всех остальных. Возможно у них получился больший баланс между exploration/exploitation. В частности Decay скорее всего изначально очень много рандомно обучался, а потом чаще всего показывал хорошие результаты. Возможно с другими гиперпараметрами этот график выглядел бы по другому