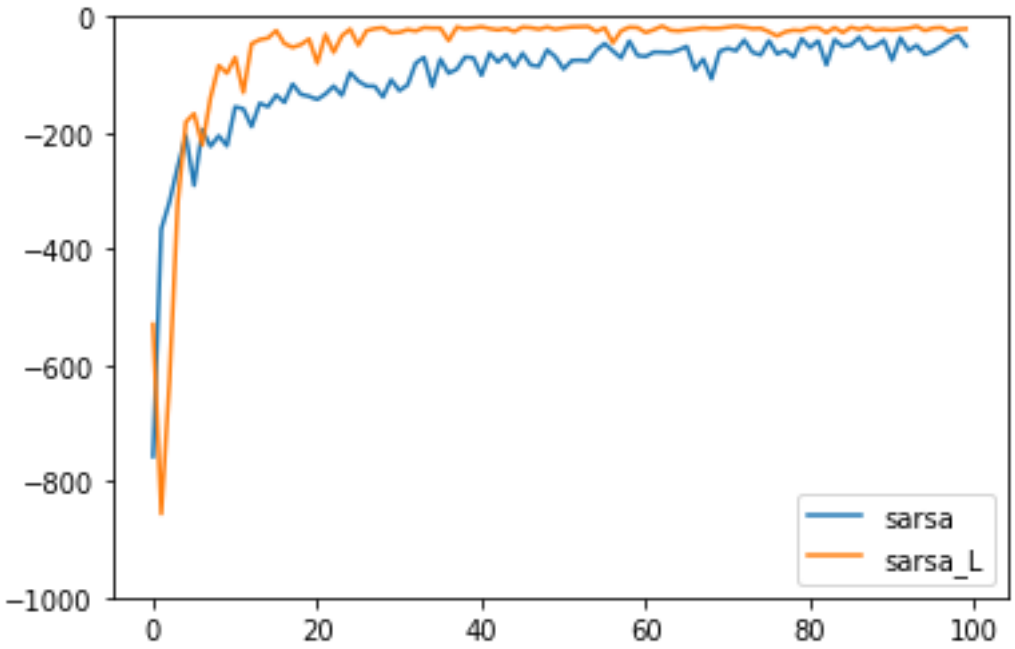
108061217 鍾永桓 HW8

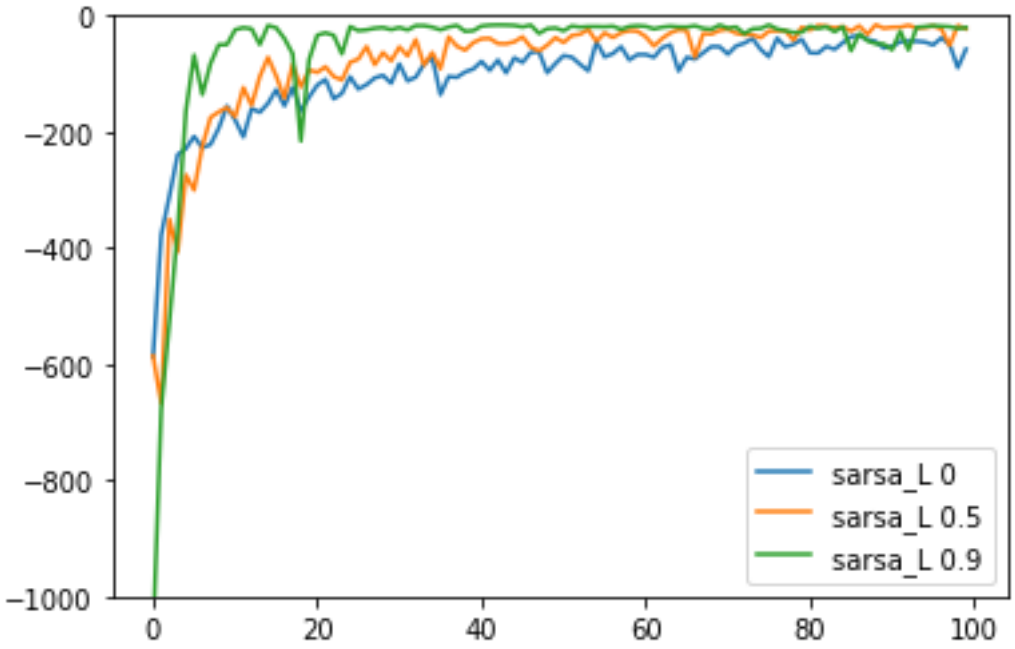
Result (sarsa and sarsa_L(0.9))



Experiment1

Compare the result of sarsa(0.0),sarsa(0.5),sarsa(0.9),sarsa,in one graph.

雖然和時間差分法的實現方法有所不同，但當 lambda 接近 1 時，會有接近於蒙地卡羅法的表現，而當 lambda 接近 0 表現會近似於 1 步時間差分法，而 0.5 和 0.9 在 1 和 0 之間，他們的表現有點接近於 n 步時間差分法，當 lamda 越大時會使資格跡衰退越慢，也表示越多步以前拜訪的經驗都能造成影響，而由本次結果來看 lambda 較大時收斂較快，也可知道在本題中每次更新時能使用較多過去的經驗可以使其收斂加快。

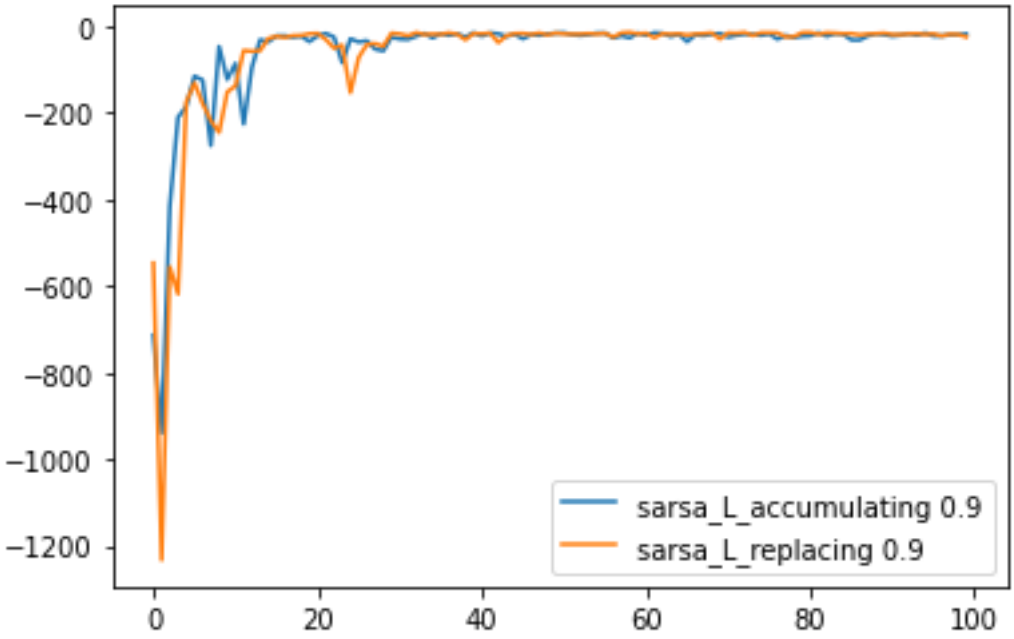


Experiment2 (20%)

In Sarsa lambda algorithm line 12, we can use both accumulating traces or replacing trace.

If we use accumulating traces, In which condiction $\alpha * z(s,a)$ in line 14 will greater than 1? And what happen if $\alpha * z(s,a)$ greater than 1, show your result.

α *z(s,a)>1 可能會發生在同一個 state-action pair 被拜訪超過一次的情況，因為在 accumulating traces 時過去所累積的拜訪經驗不會被重置，所以當一個 state-action pair 過去被拜訪過，並且還未衰退到接近 0 時又再度被拜訪就會使 z(s,a)出現大於 1 的情況，而 z(s,a)累積夠大就會使 α *z(s,a)>1。在本題中因為狀態很多，只會偶爾出現 α *z(s,a)>1，而且通常不會累積到超過 1 太多，所以從圖上其實 accumulating traces 和 replacing traces 最終都能夠成功地收斂，但是如果某個 state-action pair 被頻繁拜訪導致 α *z(s,a)持續大於 1 且越累積越大將導致每次更新的幅度都很大，可能無法穩定收斂。


0
−200
−400
−600
−800
−1000
−1200
0
20
40
60
80
100
sarsa_L_accumulating 0.9
sarsa_L_replacing 0.9