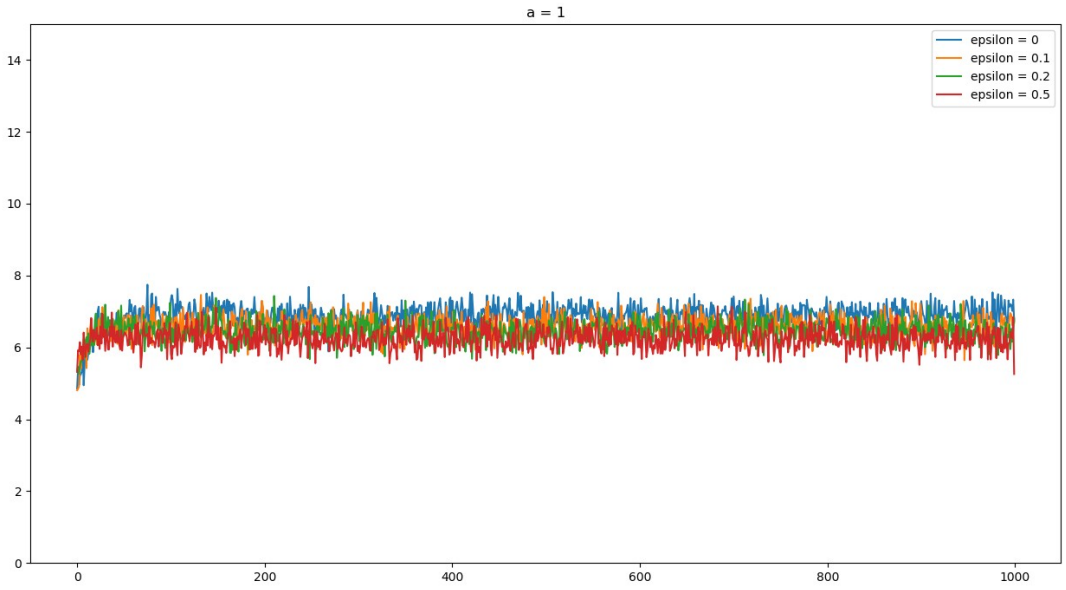
HW 1 Write-up – Isaiah King

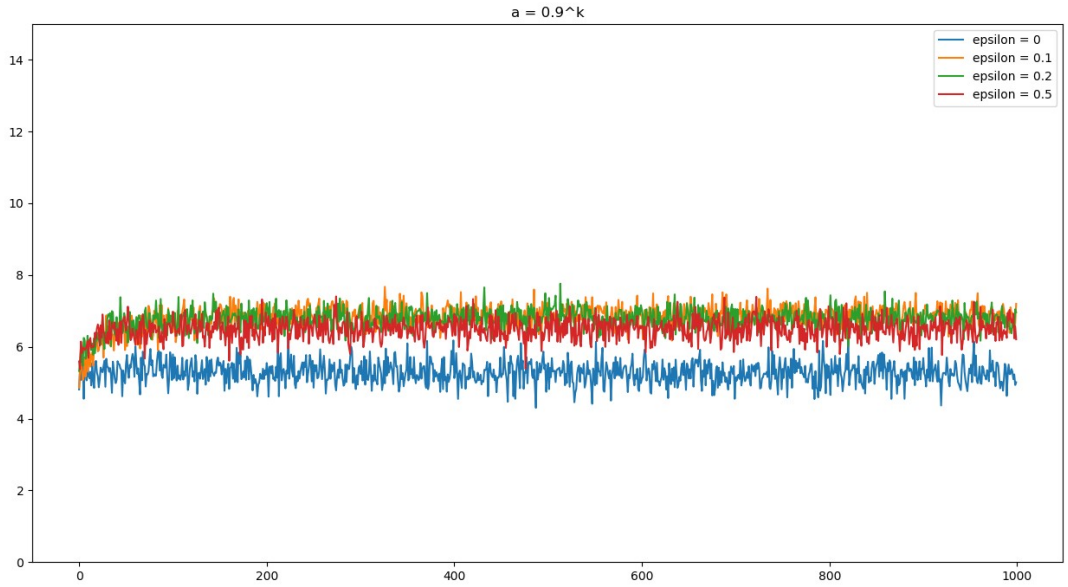
Alpha = 1

Epsilon	Final Average	Std Dev Q1	Std Dev Q2	Avg Q1	Avg Q2
0	6890.89065885812	1.63923246187441	2.66881454167894	-2.27337832296826	6.64927976279132
0.1	6581.33219751088	2.70537655075664	3.10000694657346	1.92350443407093	6.53528076221983
0.2	6446.93035920948	3.0911386062395	2.6528759718471	2.91765318802312	6.73771918921912
0.5	6244.92786178747	2.78779866140883	2.60456564835427	3.97563338426418	6.11340138453677



When alpha is set to 1 the model is able to get decent results with reasonable epsilon values (0.1, and 0.2) however, this was not the most optimal configuration. When compared with other values of alpha, it is clear that the model continually gives the intermediate, new gradient for Q too much weight in later stages. As such, it has a harder time selecting the most optimal option at a point where it should have learned it, and the greedy method is the top performer.

Alpha = 0.9^k

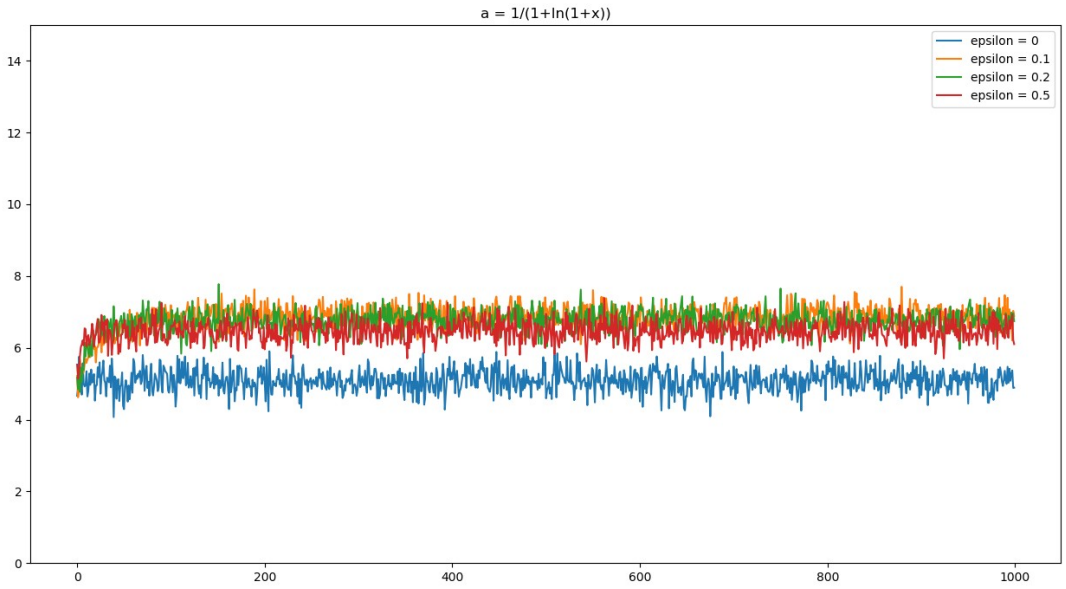


Epsilon	Final Average	Std Dev Q1	Std Dev Q2	Avg Q1	Avg Q2
0	5283.18910779607	2.2123007679894	2.49518184745718	4.0748988902056	1.04670079243224
0.1	6821.21113840801	0.525005066782709	0.401335993164101	4.87216934371625	7.01812005642147
0.2	6771.5291410943	0.54570717185915	0.39134648920484	5.05728336826527	6.97986018266568
0.5	6491.51051763507	0.527889934804553	0.417946412546868	5.00501077413459	6.99927755363212

This was one of the highest performing alpha-values for the model. Though, we observe that its optimization depends heavily on the value selected for epsilon. This is abundantly clear when we observe the performance of the greedy action, which performed very poorly. We also observe that with a higher performing alpha value, we trade off efficiency, as the best performing epsilon (e=0.1) takes significantly longer to achieve peak performance. This situation is prevalent in all subsequent models.

Alpha = 1/(1+ln(1+k))

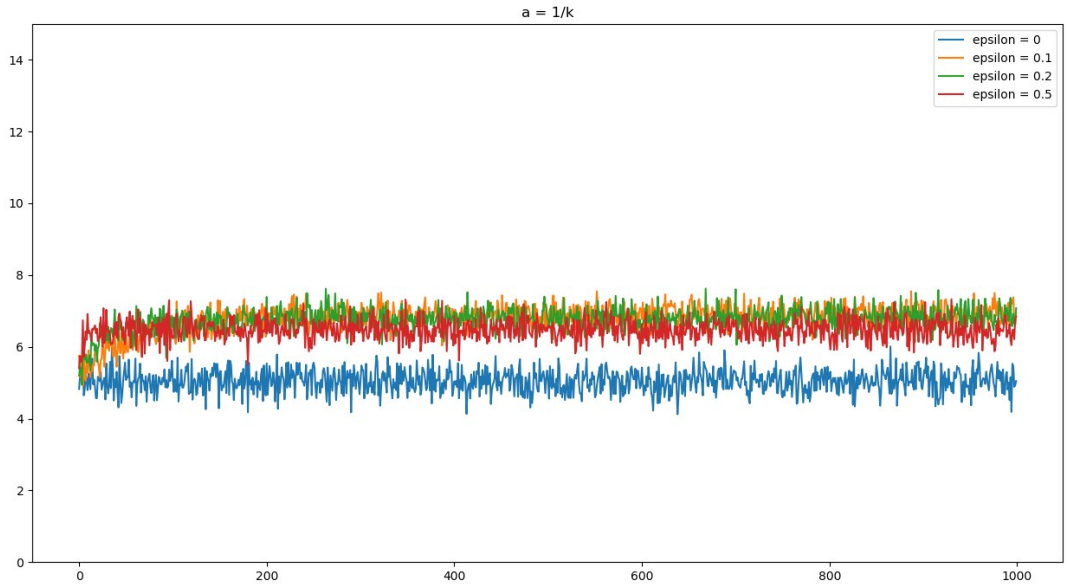
Epsilon	Final Average	Std Dev Q1	Std Dev Q2	Avg Q1	Avg Q2
0	5085.2375556765	1.54383008187956	1.37604452767924	4.75841087400683	0.314520859937151
0.1	6839.71094195909	0.846417170346802	0.618298915684755	4.79792112891301	6.99410727769342
0.2	6746.9647976458	0.875392754337057	0.65421277586557	4.69838744013088	7.00410093433458
0.5	6466.64331186991	0.814993724004388	0.748293382106656	5.04060559606491	6.99952308819145



This model performed in a very similar manner to the a=0.9^k model. However, we note that the descending weight placed on delta-Q in this model produced the highest results of any non-optimistic initial valued model (where epsilon=0.1). We observe that this model learned the optimal parameters faster than any, other than the greedy and optimistic initial valued models, as is evident from the graph, and attribute its success to this.

Alpha = 1/k

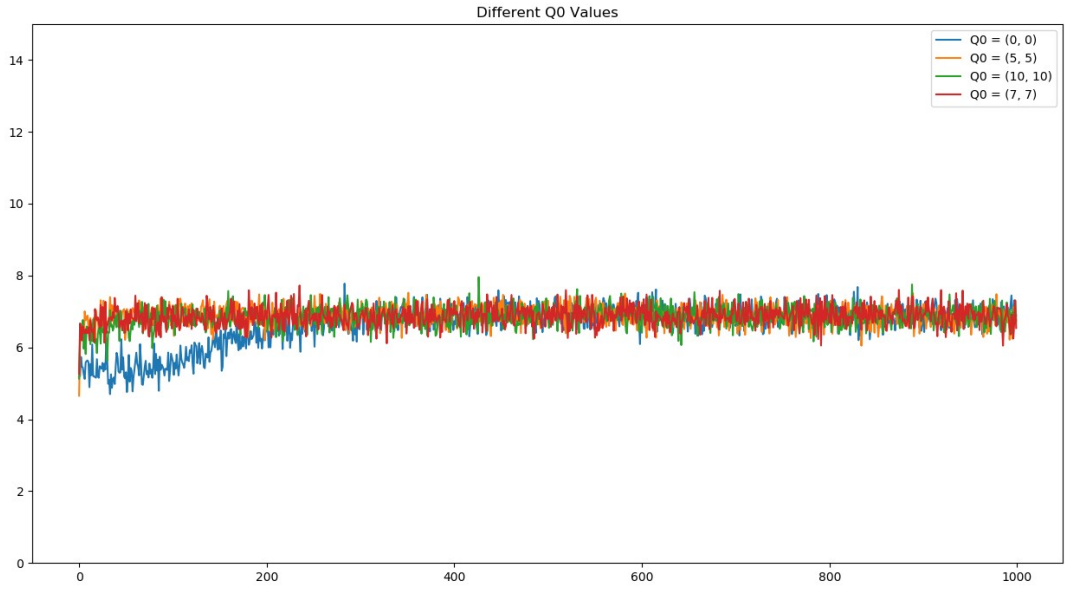
Epsilon	Final Average	Std Dev Q1	Std Dev Q2	Avg Q1	Avg Q2
0	5046.11049819148	0.981685227313918	1.20040024689029	4.80854453006885	0.211078235807005
0.1	6793.05925851736	0.36824256802529	0.078070816285189	4.88346345533684	6.99199422424852
0.2	6755.74944015264	0.261142239748391	0.087161929242341	4.96771316615296	6.99483653629758
0.5	6480.63050611807	0.198741969027699	0.095967148549437	4.97365831164037	6.98445231682338



This was another somewhat effective model, however it is not the best. Again e=0.1 performs best, however, it takes too long to learn the optimal parameters and as a result, performs only slightly better than more exploratory models such as e=0.2 and even e=0.5

Optimistic Initial Values

Q0	Final Average	Std Dev Q1	Std Dev Q2	Avg Q1	Avg Q2
(0, 0)	6631.82009349832	0.65694125989213	0.54063547215291	4.91698263950481	7.08502559979904
(5, 5)	6887.41875102592	0.706052906538489	0.616373871219345	4.89175541674389	6.94043348284073
(10, 10)	6845.88944888071	0.628156005370831	0.556002385098637	4.88027010771613	7.00842706344274
(7, 7)	6870.29849657478	0.648197992511812	0.593883148424607	4.82893973509639	6.91620226395745



By using optimistic initial values, the time it took for the model to find the best Q values was drastically reduced. The blue line for optimizing at 0,0 (which is what previous models did) is a clear indicator of this. Predictably, 5,5 was the highest performing model. This is likely because it was the closest to the average optimal value of Q2, which is consistently just under 5.