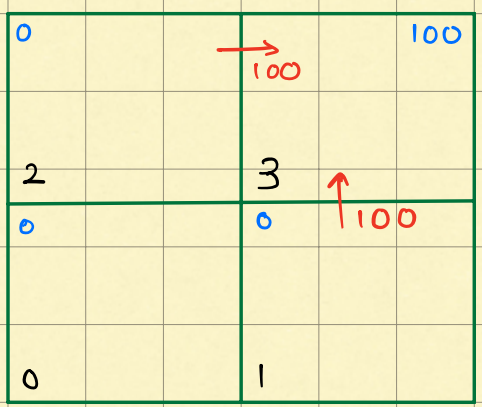
Q-learning



① Q-table state action that lead to the nex step reward state are filled first.

It's referred as back propagation of rewards or bootstrapping.

0	0	0	100
0	1	2	3

② With random series of action say, state 1 and state 2 are filled first.

ⓘ If we take bottom action for 2, the q-table for state 2 action bottom won't be updated.

Reason: (Refer to Algorithm below)

ⓐ Immediate reward for state 0 = 0, delayed reward i.e, max reward available in any direction from 0 is also zero.

ⓑ Basically the reward is propagated (core logic) When agent first jumps into a state adjacent to the final state, its q-table (state, action)

is filled with reward, here it's 100.

③ When the agent accidentally stumbles upon the reward state without intentionally taking action, it doesn't execute Q-learning in the traditional sense.

Algorithm

While n < 200 ← user defined

① Pick a random state

② While state != reward state:

No. of times executed will for sure.

ⓐ Select random action

ⓑ Get the state the action leads to.

ⓒ get immediate reward for this state.

ⓓ get delayed reward for this state

delayed reward is retrieving maximum reward value from Q-table (for a state) meaning highest reward from a set of action.

get delayed reward * decay (maybe 0.9)

looking back, we get immediate reward value for taken action-state, get most-rewarding value for an action from that state * decay.

We sum, total reward
= immediate reward +
delayed reward.

Save this snapshot, i.e
the state-action with the total reward to the
q-table.

Q-table has reward values for state-action.