KI Project B

Martijn Dekker 6013368
Egor Dmitriev 6100120
Cody Bloemhard 6231888

December 6, 2018

1 Exercise 1

1.a ex1-a

The game state holds all the data that represents the game in the current time frame. bla bla bla.

1.b ex1-b

The agent state holds all data that describes an agent in the current time frame. bla bla bla.

1.c ex1-c

- A: II, Stack: Pile of dishes
- B: III, Queue: Roller coaster waiting line
- C: I, PriorityQueue: Emergency room waiting line

2 Exercise 2

2.a

3 Exercise 3

3.a ex3-a

That means that if there exist a path to the goal you are searching for, it will return that goal.

3.b ex3-b

It is complete because the algorithm will keep searching till it found the path to the goal location or till there are no more possible paths to stroll, which haven't been visited before. This would happen when the stack is empty and that can only occur if it has tried every possible path.

3.c ex3-c

It will not always be the least cost solution. This is because it tries out a path and continues it till it ends, which can lead to the goal location. This first path it finds does not have to be the path with the least cost. And because it stops after finding a path it doesn't always find the least cost solution.

3.d ex3-d

Yes, the order is as we expected. Pacman does not visit all the explored squares on his way to the goal. This is because the first path he explores may or may not be a dead end. If this is the case then he will not go there, because his objective is to reach the goal.

4 Exercise 4

4.a ex4-a

It is complete because the algorithm will keep searching till it found the path to the goal location or till there are no more possible paths to stroll, which haven't been visited before.

4.b ex4-b

Yes, because every step pacman can take is equal in cost the algorithm needs to find the path with the least steps. And because it looks into every path of depth k before looking into paths of depth $>$ k it will always find the least cost solution.

4.c ex4-c

Yes, it returns a solution, because the algorithm was implemented so well that it does not matter what problem it is used on.

5 Exercise 5

5.a ex5-a

For the first agent: it's intended behaviour is to find the shortest path to the goal, because there is no reason not to cause there are no obstacles, therefore every path costs the same. It is achieved by sorting the currently visible paths based on their cost so far, with the cheapest sorted at the front. That way the algoritm will constantly pick the cheapest path to further explore. And because every path costs the same it basically works like breadth first search. For the second agent: it's intended behaviour is to take the east route because that path has some food on it. The food makes for a cheaper path than a normal path. It is achieved almost the same way as the first agent, only now not every path costs the same. The path with the food costs less, so that path will be explored first instead of the other paths. For the third agent: it's intended behaviour is to take the most west path, because the other paths contain ghosts which drive up the cost of a path. This is because the paths with the ghosts have a high chance of death. It is achieved the same way as with the other agents, only now the paths with the ghosts have a high cost and thus they will be sorted at the end. This way the algorithm explores the path without ghosts first.

When the algorithm finds a path to the goal it stops looking because it chose the cheapest path at all time, therefore the first path must be the cheapest.

5.b ex5-b

6 Exercise 6

7 Exercise 7

7.a

Both files contain a class which can be extended to implement an agent. Both of these classes implement a different base class / interface.
The one in file *valueIterationAgents.py* is meant to be implemented for agents which estimate the Q-values and values for an environment using a Markov Decision Process, which is done before acting. The other in file *qlearningAgents.py* is meant to be implemented for agents that estimate Q-values from policies rather than from a model.

7.b

- In reinforcement learning transition probabilities are unknown. Which is one of the reasons why MDP can't be used in such cases.
- In reinforcement learning transition rewards are unknown. These are learned with experience and can differ every iteration.

7.c

GridWorld implementation differs in how terminal state and rewards is handled. In contrary to slides the rewards are set per grid cell basis. Which means a cell can give a reward and not a be a terminal state (although only action possible in a cell with a reward is a terminal state).
Terminal states do not belong to any cell, nor do they give any rewards. They are implemented as actions. These are only to indicate when the iteration should break. In the slides they behave as a cell, but a final one.

7.d

"A terminal state is a state that once reached causes all further action of the agent to cease."
Source: `http://burlap.cs.brown.edu/tutorials/bd/p1.html`.

Both slides and implementation use this definition. Since no actions can be taken once a terminal state is reached.

8 Exercise 8

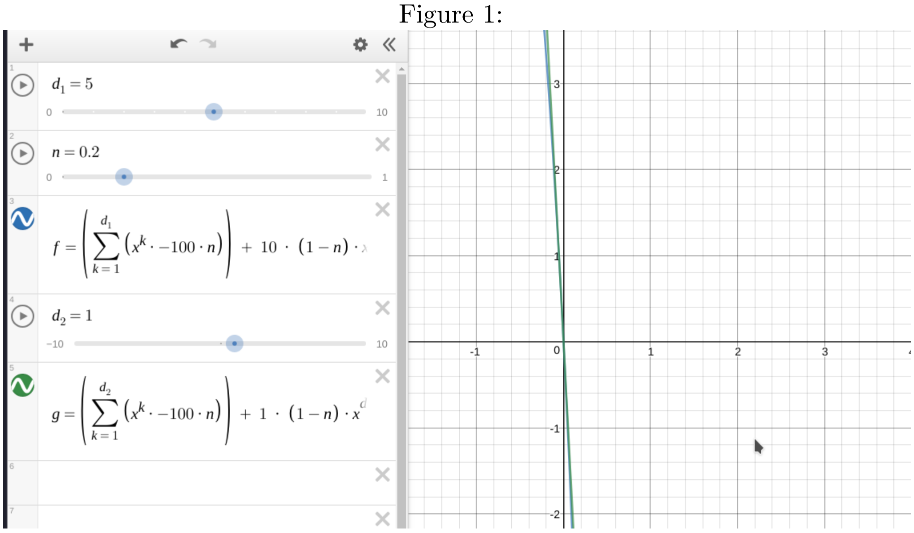
9 Exercise 9

9.a

To solve these questions we have defined a few functions which represent the values of the 2 given options. Go left and don't cross the bridge and cross the bridge. We have chosen X axis to represent the discount and Y axis to represent value / score of the given function / option.

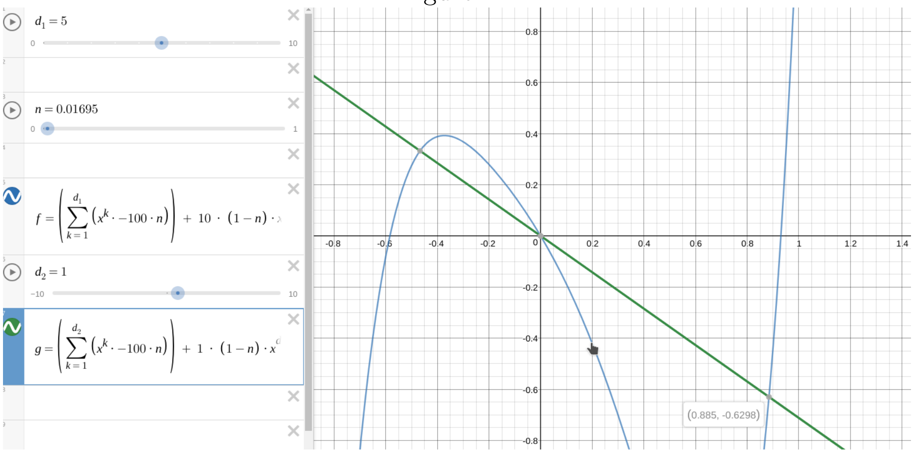
- Agent will never cross the bridge is only discount is changed. See Figure 1. The green lines is always on top.
- AgentAgent is able to cross the bridge when discount is 0.9 and noise is lower than 0.01695. This is because the the value functions cross below discount of 0.9 around this value. Figure 2

Figure 1:



Blue = Cross the bridge values
Red = Take reward on the left
Noise = 0.2

Figure 2:



Blue = Cross the bridge values
Red = Take reward on the left
Noise = 0.01695

9.b

We have changed the noise parameter, since it is not possible to do this by only changing the discount. See explanation above. We have chosen 0.01695 as noise value since it lies right on the border of possible noise values which allow agent to cross the bridge. Values lower than this one will give same results and values above* won't.

10 Exercise 10

10.a

a (γ = 0.2, n = 0, r = 0)

b (γ = 0.5, n = 0.1, r = -1)

c (γ = 0.5, n = 0, r = 0)

d (γ = 0.5, n = 0.25, r = 0)

e (γ = 0, n = 0, r = 0)

10.b

For this exercise we have plotted formulae to compute the value / score of different path options. We have chosen X axis to represent the discount and Y axis to represent value / score of the given path option.
Legend for the following figures:

n Noise parameter

l Living reward parameter

Red line Prefer the close exit (+1), risking the cliff (-10)

Blue line Prefer the distant exit (+10), risking the cliff (-10)

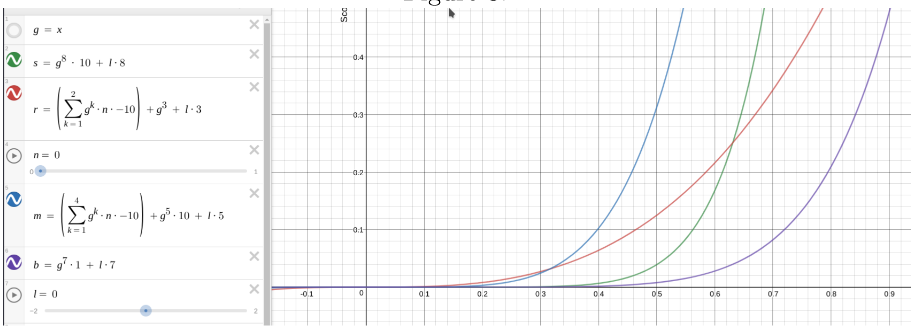
Green line Prefer the distant exit (+10), avoiding the cliff (-10)

Purple line Prefer the close exit (+1), but avoiding the cliff (-10)

10.b.1 a

Because noise is 0 this path is considered safe. Therefore agent would go along the bridge. If discount is low enough the shorter path will be preferred. This preference is amplified if living reward is negative.
We have chosen for for discount of 0.2 since that allows other parameters be unchanged. See Figure 4 the red line.

Figure 3:



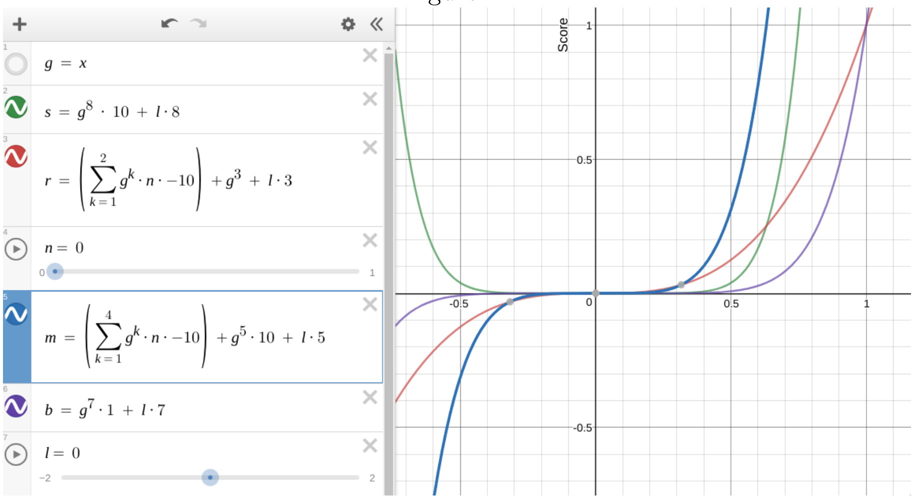
10.b.2 b

Here we make living reward negative to discover long routes. Along with this we make noise big to discourage risk. With this we see that purple line has a higher value then other paths with discount of 0.5, noise of 0.1 and living reward of -1.

10.b.3 c

Here we set noise to 0 to remove any risks and we make long term reward (discount) tempting enough to pick the longest path along the bridge. We can see with noise and living reward set to 0 that the blue line has the higher score than other paths at discount of 0.5. See Figure 5 the blue line.

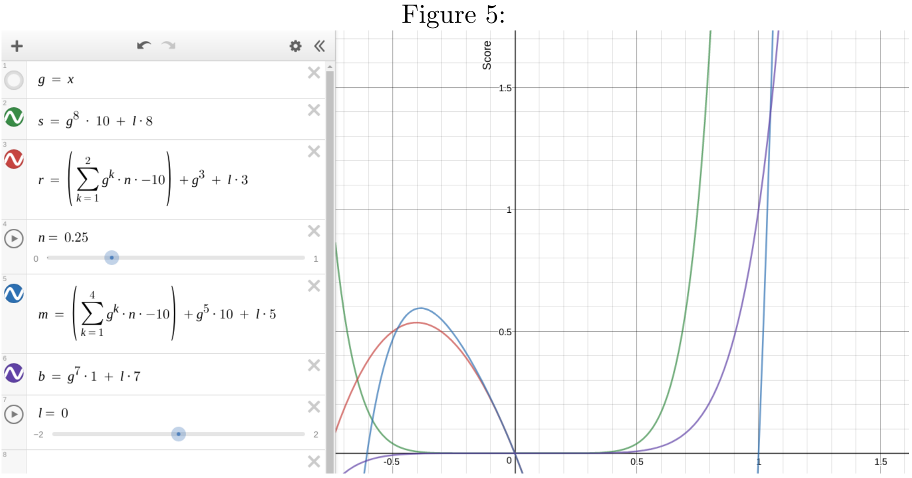
Figure 4:



10.b.4 d

Here we set noise high enough to discourage risk and pick the right discount to encourage long term reward. At discount of 0.5 green line has the highest value. See Figure 6 the green line.

Figure 5:



10.b.5 e

Here we put everything at zero. Because reward is getting multiplied by gamma and gamma is set to 0, every Q-value will also be zero and thus the episode would never terminate.

11 Exercise 11

12 Exercise 12

13 Exercise 13