Enter here your first and last name and your e-mail adress as regiistered in Canvas.

* Name, e-mail: Shab pompeiano , guspomsh@student.gu.se
* Name, e-mail: Haider Ali , gushaial@student.gu.se

Problem 1

$$Q(s,a) = \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V(s')\right]$$

Problem 4

$$V_{k+1}(s) \ <- \ max_a \sum_{s'} T(s,a,s')[R(s,a,s') + V_k(s')]$$

Problem 6

python gridworld.py -d 0.9 -n 0.0 -a value -i 100 -g BridgeGrid

By setting the noise option to 0, the agent always takes the action that it decides to take by becoming deterministic. By doing so it always takes a step to the right and reaches the end of the bridge.

Problem 10

a) the crawler learns far faster when the epsilon is really small, so when random actions are almost never taken and in this case the crawler uses the stick making really small but frequent steps. This may be because it is faster to make this kind of steps rather than making more natural and longer but less frequent steps.
b) The reason for this to be NOT POSSIBLE could be because even though we try to set the epsilon to 1, which means that we are going to take random actions all the time, the furthest part of the bridge to the right will not be reached because there are too many steps in between and by taking random actions it is far more possible to fall off the bridge than cross the whole 5-steps long bridge. Out of 5 random values all five of them have to be taken in the direction to the right (5 times in a row), and this is highly unlikely in only 50 iterations.
   By having 50 iterations, we have 5 actions to take and that is 10 chances of getting the random combination of actions right. If we think in a probabilistic way we have 4 (number of possible actions per state) multiplied by 5 (number of actions to take to reach the end of the bridge) which gives 20. This means that out of 20 possible combinations of actions, one would lead to the end of the bridge. Because we only

have 10 chances to cross the bridge when we have 50 iterations as written before, mathematically we would at least need 100 iterations to reach the end of the bridge.

c) When using the e-greedy policy, random actions are taken which causes the algorithm to find the best approach but slower because these random actions cause the agent to try paths that could lead nowhere. On the other hand using the best-policy approach will cause the agent to find a path really fast but the agent might not find the best path in absolute because the best path could instead be in an unexplored path.

d) We tried to use a larger epsilon during the first iterations and slowly decreasing epsilon through the iterations to reach epsilon = 0. We decided to take this approach because the agent would find with a large probability the best path during the first iterations also by taking random steps and exploring different paths. As more iterations pass it becomes less necessary to try random actions, once different paths are explored the agent should have an idea of the best path.