**Question 1: This question is based on the following scenario:**

**UQ Soda is a very small soft drink seller at UQ. It sells two types of soft drinks: Coke and Sprite. It buys its stock each morning, but can only buy at most 2 cans of soda per day and can only store and sell 3 cans of soda per day. You can assume that:**

* **Coke and Sprite are the only two types of sodas that the customer may want to buy**
* **The customer will not change its preference and will leave UQSoda without buying anything if his/her choice is not available**
* **The sodas will always be in a good condition**
* **The customers preference depends on the amount of each type of sodas available at the beginning of the day (right after UQSoda buys its stock)**

**Despite of its size and lack of information about the behaviour of its consumer, UQSoda wants to stock the sodas such that it can minimize the number of customers who could not get their choice of sodas**

1. **The stocking problem of UQ Soda in Question 3 Tutorial 8 is an MDP problem. However, the stocking problem in the above scenario is a reinforcement learning problem, even though the problem is similar to the scenario in Question 3 of Tutorial 8. Please explain the differences and similarity between MDP and Reinforcement Learning. In your explanation, please provide a comparison example based on the UQSoda stocking problems as described above and as described in Tutorial 8.**
2. **Suppose the owner of UQSoda wants to solve the reinforcement learning stocking problem they face using model based Bayesian Reinforcement Learning problem (yes, he knows there’s such an approach). The owner knows that this means he has to frame the problem as a POMDP problem, but does not really know what a POMDP is. To help him, your task is to define the POMDP problem that represents the UQSoda stocking problem (as described in this assignment)**

*Answer:*

1. The following are the similarities and differences between MDP and Reinforcement Learning
   1. Similarities
      1. Have a State and Action space
      2. Still need Transition and Reward Function
      3. Still need to generate an optimal policy
      4. In both Tutorial 8 and the question above, we are given enough information to create the State and Action space.
         1. The different types of soda we sell, the total number of cans we can purchase and the total number of cans we can store and sell help us generate the State and Action space. These are both available in the question above and the stocking problem given in Tutorial 8.
   2. Differences
      1. In MDP, we have enough information to create the Transition and Reward Function. In Tutorial 8, we are given the probabilities for user consumption for each type of soda. This helps us generate the Transition and Reward Function. Because of this, we can use MDP to solve the problem.
      2. In Reinforcement Learning, we do not have information to create the Transition and Reward Function. In this question, we are not given the probabilities for user consumption of each type of soda. This meant that we do not know the user’s behaviour, so we cannot create these Transition and Reward Functions. Because of this, we need Reinforcement Learning.
2. POMDP Problem Definition
   1. State Space (S)
      1. State Space is a tuple of 2 elements representing the amount of cans for a particular type of soda currently stocked for a day before purchasing for additional stock
      2. Formally: S = <S0, S1> where S0 is the number of Coke cans stored and S1 is the number of Sprite cans stored and C0 + C1 <= 3.
   2. Action Space (A)
      1. Action Space is a tuple of 2 elements representing the amount of cans for a particular type of soda that was purchased for stock in a day.
      2. Formally: A = <P0, P1> where P0 is the number of Coke cans purchased and P1 is the number of Sprite cans purchased and P0 + P1 <= 2
   3. Observation Space (Ω)
      1. The observation space is the amount of cans consumed for each soda in a day
      2. Formally: O = <C0, C1> where C0 is the number of Coke cans consumed by users and C1 is the number of Sprite cans consumed by users and C0 + C1 <= 3. This is assuming that the number of cans users can consume in a day is equal to the number of cans we can purchase
   4. Transition Function (T)
      1. Assuming that the MDP model is as described by POMDP state s S
      2. Given the previous state st, the action taken at and the new state st + 1, where t = current time slice, st, st + 1 S and at A the transition function will return the probability of ending up at st + 1 after doing at at state st.
      3. Informally, it will be the probability of ending up with a particular number of cans for each soda type after purchasing soda.
   5. Observation Function (Z)
      1. Given the new state st + t, the action taken at and the observation ot, where t = current time slice, st + 1 S and ot O, the observation function will give the probability of ending up at st + 1 in relation to the amount of soda consumed by users
   6. Reward Function (R)
      1. Assuming that the MDP model is as described by POMDP state s S
      2. Given the new state s and the action taken a, where s S and a A, the reward function will give us the reward for making it to s after doing a.
      3. Informally, we will be rewarded depending on how much sodas we have left at the end of the day. For this example, we can assume that if we have 0 cans of soda that we possibly failed with providing for the user.
      4. Therefore, our rewards will be as follows:
         1. Neither of our sodas are out of stock => +10
         2. Out of stock for either Coke or Sprite => 0
         3. Out of stock for both => -5

**Question 2:**

**Let’s consider Q-learning and SARSA with greedy algorithm for action selection. Please explain the effect of different values on the performance of the learning methods. In particular, please explain the effect when = 0, = 1 and as increases from 0 to 1**

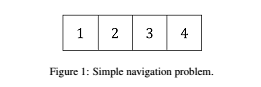
*Answer:*

1. As different as my butt is from Tom’s

**Question 3:**

**This question is based on the following simple navigation scenario.**

**A robot is navigating in an environment that has been discretised into 4 grid cells, as shown in Figure 1. We know the robot’s action space is {left, right} and that effect of performing an action is non-deterministic. However, we do not have a stochastic model of this non-deterministic behaviour of the robot. Worse, we do not know the cost of each action that the robot takes nor the reward for reaching the goal. Despite this lack of information, we want to find a policy for the robot to move from cell-1 to cell-4 with minimum cost.**



**Suppose we want to solve this problem using model-free reinforcement learning approaches. We set the initial values of all states to be zero, generated an initial policy and used this policy to generate the following two episodes.**

**Episode-1: cell1-right- -1, cell1-right- -1, cell2-right- -1, cell3-right-9, cell4**

**Episode-2: cell1-right- -1, cell2-right- -1, cell2-right- -1, cell3-right-9, cell4**

**Now we want to evaluate the policy . Please write down the value functions of the first two iterations of the following model-free reinforcement learning methods:**

1. **TD(0)**
2. **TD(2)**
3. **Monte Carlo**

*Answers:*

1. double myButtValue(string comment)
2. double tomsButtValue(string comment)
3. double buttComp(Butt a = Mybutt, Butt b = Tomsbutt)