Question 1

a) - Explore then commit: The total number of pulls, N, is the sum of wins and losses which is 20. Since this is less then k=30, we are still in the random exploration phase, so the probabilities are

$$\boxed{\begin{matrix} 1: \frac{1}{3} \\ 2: \frac{1}{3} \\ 3: \frac{1}{3} \end{matrix}}$$

- ε-greedy

The best arm is 2, so it will be pull w.p. $1-\varepsilon+\frac{\varepsilon}{3}$

$$\boxed{\begin{matrix} 1: 0.033 \\ 2: 0.933 \\ 3: 0.033 \end{matrix}}$$

- UCB, c=2

N=20, $\log N \approx 3$

$$\mu_a + c\sqrt{\frac{\log N}{N(a)}} = \begin{bmatrix} 0.5 \\ 0.8 \\ 0.5 \end{bmatrix} + 2\begin{bmatrix} \sqrt{\frac{3}{2}} \\ \sqrt{\frac{3}{10}} \\ \sqrt{\frac{3}{8}} \end{bmatrix} = \begin{bmatrix} 2.9 \\ 1.9 \\ 1.7 \end{bmatrix}$$

Antenna 1 has the highest UCB value, so it will be used w.p. 1. $\boxed{1: 1.0,\ 2: 0,\ 3: 0.}$

b) There are a total of N=10 tries of antennes 1 and 3; N=10, $\log N = 2.3$

$$\mu_a + c\sqrt{\frac{\log N}{N(a)}} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} + 2\begin{bmatrix} \sqrt{\frac{2.3}{2}} \\ \sqrt{\frac{2.3}{8}} \end{bmatrix} = \begin{bmatrix} 2.64 \\ 1.57 \end{bmatrix}$$

$\boxed{\text{Antenna 1 will be used.}}$

Question 2

a) Q-learning update

$$Q(1,1) \leftarrow Q(1,1) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(1,1)\right)$$

$$8 + 0.1(1 + 0.9 \cdot 8 - 8)$$

$$\boxed{\begin{array}{ll} Q(1,1) = 8.02 & \\ Q(1,2) = 6 & \text{(unchanged)} \end{array}}$$

b) SARSA update

$$Q(1,1) \leftarrow Q(1,1) + \alpha\left(r + \gamma Q(1,2) - Q(1,1)\right)$$

$$8 + 0.1(1 + 0.9 \cdot 6 - 8)$$

$$\boxed{\begin{array}{ll} Q(1,1) = 7.84 & \\ Q(1,2) = 6 & \text{(unchanged)} \end{array}}$$

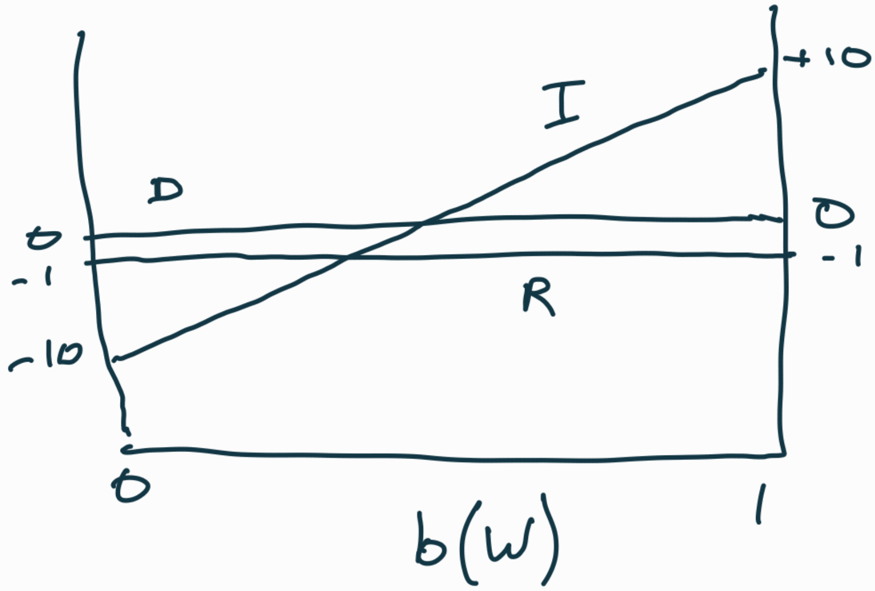
Question 3

a) $\alpha_D = [0, 0]$

$\alpha_I = [10, -10]$

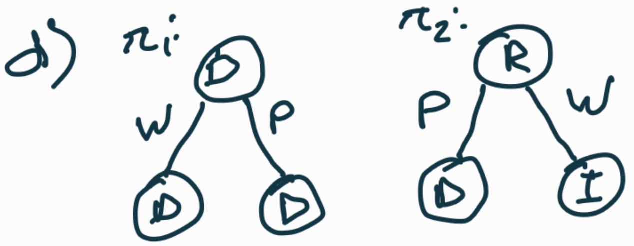
$\alpha_R = [-1, -1]$

b)



c) R will never be chosen because its α-vector is dominated over the entire belief space.

d) π_1: π_2:



e) $U^{\pi}(s) = R(s, \pi()) + \gamma \left[\sum_{s'} T(s' \mid s, \pi()) \sum_{o} Z(o \mid \pi(), s') \, U^{\pi(o)}(s') \right]$

For π_1 = always decline, $s' = s$, and $\pi_1(o) = D$

$U^{\pi_1}(s) = \underbrace{\cancel{R(s, D)}}_{0} + \gamma \underbrace{\cancel{\left[\sum_{o} Z(o \mid D, s) \, U^{D}(o) \right]}}_{0}$

$= 0 \;\therefore\; \boxed{\alpha_{\pi_1} = [0, 0]}$

can ignore this sum since $s = s'$

For π_2:

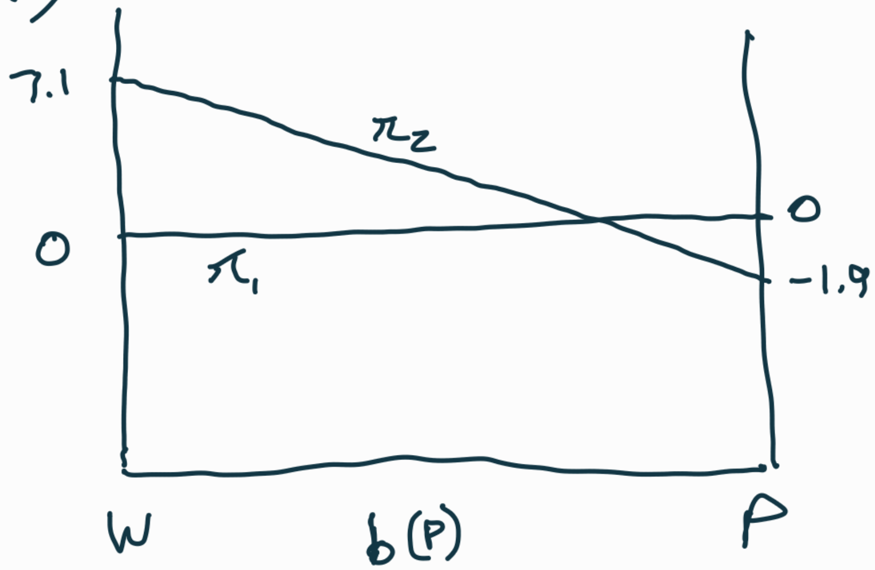
$U^{\pi_2}(s) = R(s, R) + \gamma \left[\cancel{\sum_{s'} T(s' \mid s, a)} \left(Z(P \mid R, s) \underbrace{\cancel{U^{D}(s)}}_{0} + Z(W \mid R, s) \, U^{I}(s) \right) \right]$

$U^{\pi}(P) = -1 + \gamma [0.1 \cdot -10] = -1.9$

$U^{\pi}(W) = -1 + \gamma [0.9 \cdot 10] = 7.1$

$\boxed{\alpha_{\pi_2} = [7.1, -1.9]}$

f)



g) $b = [0.5, 0.5]$

$$\pi^* = \underset{\pi \in \Gamma}{\operatorname{argmax}} \; \alpha_\pi \cdot b$$

π_1: $[0, 0] \cdot [0.5, 0.5] = 0$

π_2: $[7.1, -1.9] \cdot [0.5, 0.5] = 2.6$

$\pi^* = \pi_2$

$\pi^*() = \boxed{R}$ would be selected.

Question 4

Q-learning is more appropriate because it is an off-policy algorithm so it can learn from data collected with any policy.

Policy Gradient is on-policy, so it can only learn from data consistent with the current policy.

Thus Q-learning is easier to use.

(Note: it may be possible to adapt policy gradient or the data to be used together, but this will not necessarily be easy)

Question 5

This function is less expressive than a neural network because it can only approximate linear functions of x. Neural networks have nonlinearities that the inputs are passed through so that they can approximate nonlinear functions.