Q 3.1

Examples of MDP,

- Trading bot - State - Current Holdings of an asset
  - Reward - Money gained from a trade or lost
  - Actions - Buy / sell

- Sport Coach - State - Current score, team stamina & chemistry, of
  - Actions - Playing strategies
  - Rewards - Goals scored.

- Maze Bot - State - Current Position
  - Actions - Directions to move
  - Rewards - Find exit to the maze

Q.3.2

The MDP must not violate the Markov Property

Eg - In FPS games, the agent has no direct info about the opponents unless they are in sight but the state is influenced by both teammates & opponents, making it impossible to figure out effect of your former action on the current actn

Eg - A simpler example would be poker. The previous states determine what is in the deck & what is not, thus violating the markov property.

Q.3.3

- The natural distinction depends on the task. If the task is to go from one location to another, the actions might be in terms of directing the car & altering the speed
- Another distinction is the decision making part here. If we consider the decisions to be made by the brain of a human driving, then their physical ~~feet~~ body will form part of the environment

-

Q 3.4

| s | a | s' | r | p(s', r \| s, a) |
|---|---|---|---|---|
| high | search | high | $r_{search}$ | $\alpha$ |
| high | search | low | $r_{search}$ | $1-\alpha$ |
| low | search | high | $-3$ | $1-\beta$ |
| low | search | low | $r_{search}$ | $\beta$ |
| high | wait | high | $r_{wait}$ | 1 |
| high | wait | low | - | 0 |
| low | wait | high | - | 0 |
| low | wait | low | $r_{wait}$ | 1 |
| low | recharge | high | 0 | 1 |
| low | recharge | low | - | 0 |

Q.3.5

(original) $\sum_{s' \in S} \sum_{r \in R} p(s', r \mid s, a) = 1$ for all $s \in S$, $a \in A(s)$

(modified) $\sum_{s' \in S} \sum_{r \in R} p(s', r \mid s, a) = 1$ for all $s \in S$, $a \in A(s)$

$S$ = {Non terminal states}

$S^+$ = {All state}

Q.3.6

For episodic tasks $G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots + R_T$

If we use discounting $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots + \gamma^{T-t-1} R_T$

$$= \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}$$

The reward for success is set to 0 & -1 for failure.

$\therefore G_t = -\gamma^{T-t-1}$

This is the same return as the continuing ~~task~~ setting where we have return as $-\gamma^k$, where $k$ is the timestep before the failure.

Q3.7

If the agent keeps going randomly, it will reach the end of the maze eventually with probability 1, so the value of G under most strategies is 1. What you actually want is for the agent to leave the maze ASAP.

Also, $\gamma$ in some cases, the agent may get stuck in a infinite loop.

The better way would be add -1 reward to each time step before the escape.

Q3.8

$G_5 = 0$

$G_4 = 2$

$G_3 = 0.5 G_4 + 4 = 4$

$G_2 = 0.5 G_3 + 6 = 8$

$G_1 = 0.5 G_2 + 2 = 6$

$G_0 = 0.5 G_1 + (-1) = 2$

Q3.9

$$G_1 = 7 \frac{\gamma}{1-\gamma} = 63$$

$$G_0 = 2 + 7 \frac{\gamma}{1-\gamma} = 2 + \frac{0.9 \times 7}{1-0.1} = 65$$

Q3.10

$|r| < 1$

$$S_N = \sum_{i=0}^{N} r^i$$

$$r S_N - S_N = r^{N+1} - 1$$

$$S_N = \frac{1 - r^{N+1}}{1 - r}$$

$$S = \lim_{N \to \infty} S_N = \frac{1}{1-r}$$