Laboratorio 2

Task 1

1. ¿Qué es un Markov Decision Process (MDP)?

Un Markov decision process es un proceso en el cual se toman decisiones estocásticas que utiliza un marco matemático para modelar la toma de decisiones de un sistema dinámico en escenarios donde los resultados son aleatorios o controlados por un tomador de decisiones, este es el quien toma decisiones secuenciales a lo largo del tiempo.

2. ¿Cuáles son los componentes principales de un MDP?

A. Estados (S): Son un conjunto de todos los estados posibles en los que el entorno puede encontrarse.
B. Acciones (A): Son un conjunto de todas las acciones posibles que el agente puede tomar.
C. Función de transición de estados (P): Describe la probabilidad de transición de un estado a otro dado una acción, formalmente es la probabilidad de que el entorno transiciones al estado s' cuando el agente realiza la acción a en el estado s.
D. Recompensa (R): Esta es una función que asigna una recompensa inmediata a cada par de estados y acción.
E. Descuento (y): Este es un factor de descuento entre 0 y 1 que representa la importancia de las recompensas futuras en comparación de las recompensas inmediatas.

3. ¿Cuál es el objetivo principal del aprendizaje por refuerzo con MDPs?

El objetivo principal del aprendizaje por refuerzo con los MDP es encontrar una política optima que maximice la recompensa acumulada a lo largo del plazo para el agente, también esta política se puede definir como la estrategia o conjunto de reglas que el agente sigue para decidir que acciones tomar en cada estado, la política optima es aquella que maximiza la recompensa acumulada a largo plazo.

Task 2

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import random
```

Defina los estados

In [ ]:
```
S = {x for x in range(0, 9)}
```

Defina las acciones

In [ ]:
```
A = ['up', 'down', 'left', 'right']
```

In [ ]:
```
goal_state = 2
obstacles = [4, 8]
```

Probabilidades de transición

In [ ]:
```
# Inicialización de las probabilidades de transición y recompensas
P = {s: {a: np.zeros(len(S)) for a in A} for s in S}
R = {s: {a: np.zeros(len(S)) for a in A} for s in S}
```

In [ ]:
```
P[0]['up'] = [0, 0, 0, 1, 0, 0, 0, 0, 0]
P[0]['down'] = [0, 0, 0, 0, 1, 0, 0, 0, 0]
P[0]['left'] = [0, 0, 0, 0, 0, 0, 1, 0, 0]
P[0]['right'] = [0, 0, 0, 0, 0, 1, 0, 0, 0]
# Estado 1
P[1]['up'] = [0, 0, 0, 0, 0, 0, 0, 1, 0]
P[1]['down'] = [0, 0, 0, 0, 0, 0, 0, 0, 1]
P[1]['left'] = [0, 0, 0, 1, 0, 0, 0, 0, 0]
P[1]['right'] = [0, 0, 0, 0, 1, 0, 0, 0, 0]
# Estado 2
P[2]['up'] = [0, 0, 0, 0, 0, 0, 0, 1, 0]
P[2]['down'] = [0, 0, 0, 0, 0, 0, 0, 0, 1]
P[2]['left'] = [0, 0, 0, 0, 0, 0, 1, 0, 0]
P[2]['right'] = [0, 0, 0, 0, 0, 1, 0, 0, 0]
# Estado 3
P[3]['up'] = [0, 0, 0, 0, 0, 0, 0, 1, 0]
P[3]['down'] = [0, 0, 0, 0, 0, 0, 0, 0, 1]
P[3]['left'] = [0, 0, 0, 0, 0, 0, 1, 0, 0]
P[3]['right'] = [0, 0, 0, 0, 0, 1, 0, 0, 0]
# Estado 4
P[4]['up'] = [0, 0, 0, 0, 0, 0, 0, 1, 0]
P[4]['down'] = [0, 0, 0, 0, 0, 0, 0, 0, 1]
P[4]['left'] = [0, 0, 0, 1, 0, 0, 0, 0, 0]
P[4]['right'] = [0, 0, 0, 0, 1, 0, 0, 0, 0]
# Estado 5
P[5]['up'] = [0, 0, 0, 0, 0, 0, 0, 1, 0]
P[5]['down'] = [0, 0, 0, 0, 0, 0, 0, 0, 1]
P[5]['left'] = [0, 0, 0, 0, 0, 0, 1, 0, 0]
P[5]['right'] = [0, 0, 0, 0, 0, 1, 0, 0, 0]
# Estado 6
P[6]['up'] = [0, 0, 0, 0, 0, 0, 0, 1, 0]
P[6]['down'] = [0, 0, 0, 0, 0, 0, 0, 0, 1]
P[6]['left'] = [0, 0, 0, 1, 0, 0, 0, 0, 0]
P[6]['right'] = [0, 0, 0, 0, 1, 0, 0, 0, 0]
# Estado 7
P[7]['up'] = [0, 0, 0, 0, 0, 0, 0, 1, 0]
P[7]['down'] = [0, 0, 0, 0, 0, 0, 0, 0, 1]
P[7]['left'] = [0, 0, 0, 0, 0, 0, 1, 0, 0]
P[7]['right'] = [0, 0, 0, 0, 0, 1, 0, 0, 0]
# Estado 8
```

```
P[8]['up'] = [0, 0, 0, 0, 0, 0, 0, 1, 0]
P[8]['down'] = [0, 0, 0, 0, 0, 0, 0, 0, 1]
P[8]['left'] = [0, 0, 0, 1, 0, 0, 0, 0, 0]
P[8]['right'] = [0, 0, 0, 0, 1, 0, 0, 0, 0]
```

In [ ]:
```
print(P)
```

```
{0: {'up': [0, 0, 0, 1, 0, 0, 0, 0, 0], 'down': [0, 0, 0, 0, 1, 0, 0, 0, 0], 'lef
t': [0, 0, 0, 0, 0, 0, 1, 0, 0], 'right': [0, 0, 0, 0, 0, 1, 0, 0, 0]}, 1: {'up':
[0, 0, 0, 0, 0, 0, 0, 1, 0], 'down': [0, 0, 0, 0, 0, 0, 0, 0, 1], 'left': [0, 0,
0, 1, 0, 0, 0, 0, 0], 'right': [0, 0, 0, 0, 1, 0, 0, 0, 0]}, 2: {'up': [0, 0, 0,
0, 0, 0, 0, 1, 0], 'down': [0, 0, 0, 0, 0, 0, 0, 0, 1], 'left': [0, 0, 0, 0, 0, 0,
1, 0, 0], 'right': [0, 0, 0, 0, 0, 1, 0, 0, 0]}, 3: {'up': [0, 0, 0, 0, 0, 0, 0,
1, 0], 'down': [0, 0, 0, 0, 0, 0, 0, 0, 1], 'left': [0, 0, 0, 0, 0, 0, 1, 0, 0],
'right': [0, 0, 0, 0, 0, 1, 0, 0, 0]}, 4: {'up': [0, 0, 0, 0, 0, 0, 0, 1, 0], 'dow
n': [0, 0, 0, 0, 0, 0, 0, 0, 1], 'left': [0, 0, 0, 1, 0, 0, 0, 0, 0], 'right': [0,
0, 0, 0, 1, 0, 0, 0, 0]}, 5: {'up': [0, 0, 0, 0, 0, 0, 0, 1, 0], 'down': [0, 0, 0,
0, 0, 0, 0, 0, 1], 'left': [0, 0, 0, 0, 0, 0, 1, 0, 0], 'right': [0, 0, 0, 0, 0,
1, 0, 0, 0]}, 6: {'up': [0, 0, 0, 0, 0, 0, 0, 1, 0], 'down': [0, 0, 0, 0, 0, 0, 0,
0, 1], 'left': [0, 0, 0, 1, 0, 0, 0, 0, 0], 'right': [0, 0, 0, 0, 1, 0, 0, 0, 0]},
7: {'up': [0, 0, 0, 0, 0, 0, 0, 1, 0], 'down': [0, 0, 0, 0, 0, 0, 0, 0, 1], 'lef
t': [0, 0, 0, 0, 0, 0, 1, 0, 0], 'right': [0, 0, 0, 0, 0, 1, 0, 0, 0]}, 8: {'up':
[0, 0, 0, 0, 0, 0, 0, 1, 0], 'down': [0, 0, 0, 0, 0, 0, 0, 0, 1], 'left': [0, 0,
0, 1, 0, 0, 0, 0, 0], 'right': [0, 0, 0, 0, 1, 0, 0, 0, 0]}}
```

In [ ]:
```
print(R)
```

```
{0: {'up': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'down': array([0., 0., 0.,
0., 0., 0., 0., 0., 0.]), 'left': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'ri
ght': array([0., 0., 0., 0., 0., 0., 0., 0., 0.])}, 1: {'up': array([0., 0., 0.,
0., 0., 0., 0., 0., 0.]), 'down': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'le
ft': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'right': array([0., 0., 0., 0.,
0., 0., 0., 0., 0.])}, 2: {'up': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'dow
n': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'left': array([0., 0., 0., 0.,
0., 0., 0., 0., 0.]), 'right': array([0., 0., 0., 0., 0., 0., 0., 0., 0.])}, 3:
{'up': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'down': array([0., 0., 0., 0.,
0., 0., 0., 0., 0.]), 'left': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'righ
t': array([0., 0., 0., 0., 0., 0., 0., 0., 0.])}, 4: {'up': array([0., 0., 0., 0.,
0., 0., 0., 0., 0.]), 'down': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'left':
array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'right': array([0., 0., 0., 0., 0.,
0., 0., 0., 0.])}, 5: {'up': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'down':
array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'left': array([0., 0., 0., 0., 0.,
0., 0., 0., 0.]), 'right': array([0., 0., 0., 0., 0., 0., 0., 0., 0.])}, 6: {'up':
array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'down': array([0., 0., 0., 0., 0.,
0., 0., 0., 0.]), 'left': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'right': ar
ray([0., 0., 0., 0., 0., 0., 0., 0., 0.])}, 7: {'up': array([0., 0., 0., 0., 0.,
0., 0., 0., 0.]), 'down': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'left': arr
ay([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'right': array([0., 0., 0., 0., 0., 0.,
0., 0., 0.])}, 8: {'up': array([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'down': arra
y([0., 0., 0., 0., 0., 0., 0., 0., 0.]), 'left': array([0., 0., 0., 0., 0., 0.,
0., 0., 0.]), 'right': array([0., 0., 0., 0., 0., 0., 0., 0., 0.])}}
```

[1 2 3]

[4 5 6]

[7 8 9]

In [ ]:
```python
def move(s, a):
    if a == 'up':
        return s - 3 if s >= 3 else s
    elif a == 'down':
        return s + 3 if s <= 5 else s
    elif a == 'left':
        return s - 1 if s % 3 != 0 else s
    elif a == 'right':
        return s + 1 if s % 3 != 2 else s
    else:
        return s
```

Función de recompensa

In [ ]:
```python
for s in S:
    for a in A:
        s_prime = move(s, a)
        # P[s][a][s_prime] = 1  # Probabilidad de transición determinística
        if s_prime == goal_state:
            R[s][a][s_prime] = 10  # Recompensa por alcanzar la meta
        elif s_prime in obstacles:
            R[s][a][s_prime] = -10  # Penalización por chocar con un obstáculo
        elif s != s_prime:
            R[s][a][s_prime] = -1  # Penalización por moverse a una celda normal
```

In [ ]:
```python
# Definición de una política
policy = {s: random.choice(A) for s in S}
```

In [ ]:
```python
def simulate_policy(policy, steps=100):
    total_reward = 0
    state = 0  # Estado inicial
    for _ in range(steps):
        action = policy[state]
        next_state = move(state, action)
        reward = R[state][action][next_state]
        total_reward += reward
        state = next_state
        if state == goal_state:
            break
    return total_reward
```

In [ ]:
```python
# Evaluar la política
def evaluate_policy(policy, num_simulations=100, steps_per_simulation=100):
    rewards = [simulate_policy(policy, steps_per_simulation) for _ in range(num_sim
    avg_reward = np.mean(rewards)
    return avg_reward
```

In [ ]:
```python
# Imprimir la política y la recompensa promedio
print("Política:")
for s in S:
    print(f"Estado {s}: {policy[s]}")

avg_reward = evaluate_policy(policy, num_simulations=100, steps_per_simulation=100)
print(f"\nRecompensa acumulada promedio: {avg_reward}")
```

```
Política:
Estado 0: right
Estado 1: left
Estado 2: up
Estado 3: left
Estado 4: down
Estado 5: down
Estado 6: left
Estado 7: down
Estado 8: down

Recompensa acumulada promedio: -100.0
```

In [ ]:
```python
for s in P:
    for a in P[s]:
        print(f"Estado {s}, Acción {a}:")
        for s_prime in range(len(P[s][a])):
            print(f"  -> Estado {s_prime} P={P[s][a][s_prime]}, R={R[s][a][s_prime]
```

```
Estado 0, Acción up:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=1, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 0, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=-1.0
  -> Estado 4 P=1, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 0, Acción left:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=1, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 0, Acción right:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=-1.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=1, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 1, Acción up:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=1, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 1, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=-10.0
```

```
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=1, R=0.0
Estado 1, Acción left:
  -> Estado 0 P=0, R=-1.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=1, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 1, Acción right:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=10.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=1, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 2, Acción up:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=10.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=1, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 2, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=-1.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=1, R=0.0
Estado 2, Acción left:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=-1.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=1, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 2, Acción right:
  -> Estado 0 P=0, R=0.0
```

```
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=10.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=1, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 3, Acción up:
  -> Estado 0 P=0, R=-1.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=1, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 3, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=-1.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=1, R=0.0
Estado 3, Acción left:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=1, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 3, Acción right:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=-10.0
  -> Estado 5 P=1, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 4, Acción up:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=-1.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
```

```
  -> Estado 7 P=1, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 4, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=-1.0
  -> Estado 8 P=1, R=0.0
Estado 4, Acción left:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=1, R=-1.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 4, Acción right:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=1, R=0.0
  -> Estado 5 P=0, R=-1.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 5, Acción up:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=10.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=1, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 5, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=1, R=-10.0
Estado 5, Acción left:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
```

```
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=-10.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=1, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 5, Acción right:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=1, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 6, Acción up:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=-1.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=1, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 6, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=1, R=0.0
Estado 6, Acción left:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=1, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 6, Acción right:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=1, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=-1.0
  -> Estado 8 P=0, R=0.0
```

```
Estado 7, Acción up:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=-10.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=1, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 7, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=1, R=0.0
Estado 7, Acción left:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=1, R=-1.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 7, Acción right:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=1, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=-10.0
Estado 8, Acción up:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=-1.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=1, R=0.0
  -> Estado 8 P=0, R=0.0
Estado 8, Acción down:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=0, R=0.0
```

```
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=1, R=-10.0
Estado 8, Acción left:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=1, R=0.0
  -> Estado 4 P=0, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=-1.0
  -> Estado 8 P=0, R=0.0
Estado 8, Acción right:
  -> Estado 0 P=0, R=0.0
  -> Estado 1 P=0, R=0.0
  -> Estado 2 P=0, R=0.0
  -> Estado 3 P=0, R=0.0
  -> Estado 4 P=1, R=0.0
  -> Estado 5 P=0, R=0.0
  -> Estado 6 P=0, R=0.0
  -> Estado 7 P=0, R=0.0
  -> Estado 8 P=0, R=-10.0
```

Task 3

En clase hemos dicho que una vez tengamos v* o q* sabemos la póliza óptima π* ¿Por qué?

Si tenemos v* las mejores acciones que se pueden tomar en cada estado son las que maximizan la recompensa esperada en el tiempo inmediato. En otras palabras, la política óptima es aquella que es greedy con respecto a v. *Esto quiere decir, que si se evalúan las consecuencias de tomar una acción acorde a v* en el corto plazo, sin considerar el largo plazo, especificamente al siguiente timestep, entonces la política greedy, que elige las acciones que maximizan el reward en el corto plazo, es óptima.

Si tenemos q* solo es necesario buscar cualquier acción que maximice q*(s, a). La función acción-valor nos dice cuál es la mejor acción a tomar en cada estado, por lo que la política óptima es aquella que sigue las acciones que maximizan q*(s, a) en cada estado.