```
import numpy as np
import random


# Parameters
n_states = 6  # Maze with 6 states (0 to 5)
actions = [0, 1]  # 0: left, 1: right
q_table = np.zeros((n_states, len(actions)))
alpha = 0.1  # learning rate
gamma = 0.9  # discount
epsilon = 0.9  # exploration



# Rewards
rewards = [0, 0, 0, 0, 1, -1]  # 4 is goal, 5 is fire


# Step function
def step(state, action):
    if state == 4 or state == 5:
        return state, 0
    if action == 0:  # left
        next_state = max(0, state - 1)
    else:  # right
        next_state = min(5, state + 1)
    return next_state, rewards[next_state]


# Training
for episode in range(200):
    state = 0
    while state not in [4, 5]:
        if random.uniform(0, 1) < epsilon:
            action = random.choice(actions)
        else:
            action = np.argmax(q_table[state])
        next_state, reward = step(state, action)
        q_table[state, action] = q_table[state, action] + alpha * (
            reward + gamma * np.max(q_table[next_state]) - q_table[state, action]
        )
        state = next_state


# Test
state = 0
path = [state]
while state not in [4, 5]:
    action = np.argmax(q_table[state])
    state, _ = step(state, action)
    path.append(state)

print("Learned Path:", path)
```

```
Learned Path: [0, 1, 2, 3, 4]
```

Start coding or generate with AI.