



The Abdus Salam

**International Centre
for Theoretical Physics**

QUANTITATIVE
LIFE SCIENCES



The Abdus Salam

**International Centre
for Theoretical Physics**

**QUANTITATIVE
LIFE SCIENCES**

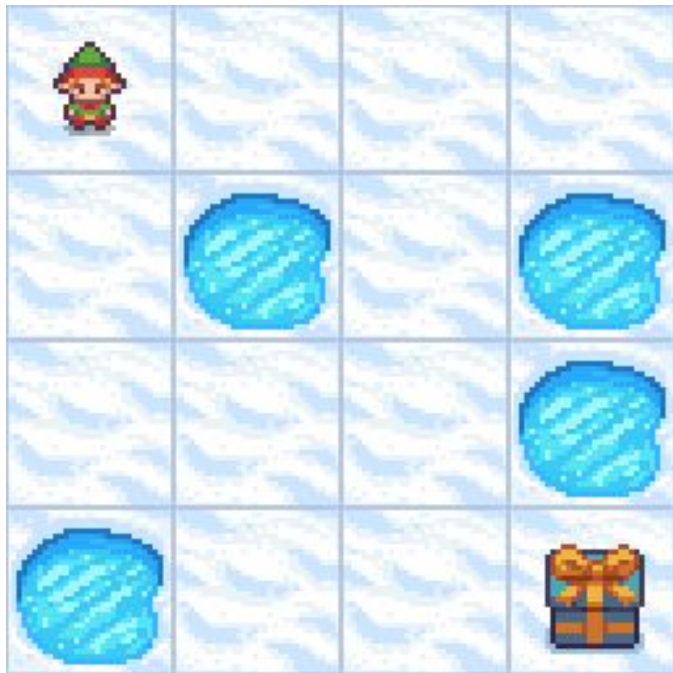
Frozen Lake

SA Torres Orozco

Description of the problem



Description of the problem



Motivation for the problem

- Simplicity and Clarity
- Introduction to Key Concepts
- Educational Value

In summary, **my motivation** stems from the desire to **learn and understand reinforcement learning** in a clear, practical, and educational manner.

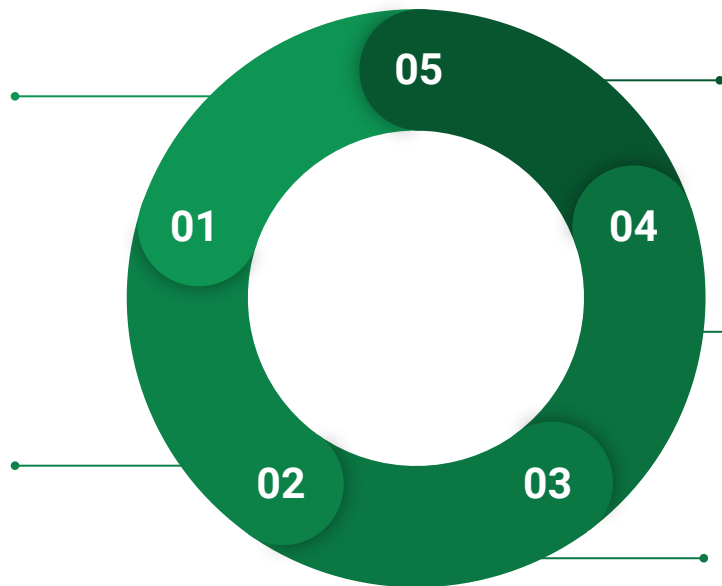
Outline

Objectives

Define the primary goals of the project

Reinforcement Learning Framework

Translating the Frozen Lake problem into an RL framework



Conclusions

Justification for the chosen methods and unique contributions to the Frozen Lake problem

Results

Analysis of the outcomes and their interpretation

Approach and Methods

Explanation of the methods

OBJECTIVES

- Determine **Optimal Policies** Using **Different Methods:**
Policy Iteration.
Value Iteration.
Q-Learning.
- Compare and **Analyze Policies.**
- **Evaluate the performance** of each method in terms of convergence speed, policy quality, and robustness in deterministic and stochastic environments.

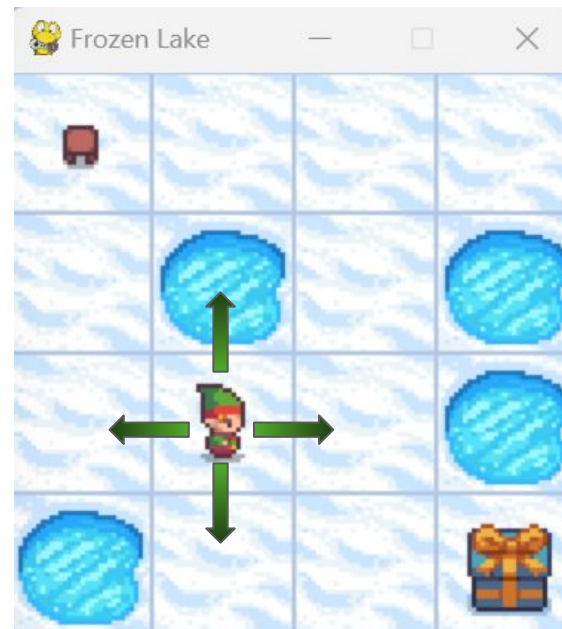
Reinforcement learning framework

- **States;** coordinates in the grid $\{(0,0), \dots, (3,3)\}$
 $S = \{0, 1, 2, \dots, 15\}$



Reinforcement learning framework

- **States;** coordinates in the grid $\{(0,0), \dots, (3,3)\}$
 $S = \{0, 1, 2, \dots, 15\}$
- **Actions;** $A = \{\text{left, down, right, up}\}$
 $A = \{0, 1, 2, 3\}$



Reinforcement learning framework

- **States;** coordinates in the grid $\{(0,0), \dots, (3,3)\}$

$$S = \{0, 1, 2, \dots, 15\}$$

- **Actions;** $A = \{\text{left, down, right, up}\}$

$$A = \{0, 1, 2, 3\}$$



- **Transitions/Dynamics**

Deterministic \longrightarrow 100 % probability

Stochastic \longrightarrow 33 % probability

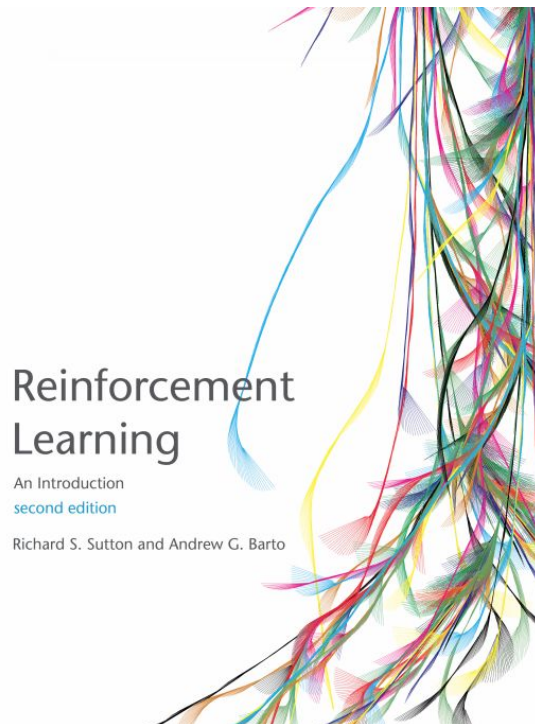


Reinforcement learning framework

- **States;** coordinates in the grid $\{(0,0), \dots, (3,3)\}$
 $S = \{0, 1, 2, \dots, 15\}$
- **Actions;** $A = \{\text{left, down, right, up}\}$
 $A = \{0, 1, 2, 3\}$
- **Transitions/Dynamics**
Deterministic  *100 % probability*
Stochastic  *33 % probability*
- **Rewards**
+1 (Goal) ; 0 (Hole) ; 0 (Frozen)

How I tried to solve it?

How I tried to solve it?



How I tried to solve it?

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization

$V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in \mathcal{S}$:

$v \leftarrow V(s)$

$V(s) \leftarrow \sum_{s',r} p(s', r | s, \pi(s)) [r + \gamma V(s')]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement

policy-stable \leftarrow true

For each $s \in \mathcal{S}$:

old-action $\leftarrow \pi(s)$

$\pi(s) \leftarrow \arg \max_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$

If *old-action* $\neq \pi(s)$, then *policy-stable* \leftarrow false

If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

How I tried to solve it?

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation

Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\text{terminal}) = 0$

Loop:

```
|  $\Delta \leftarrow 0$   
|   Loop for each  $s \in \mathcal{S}$ :  
|      $v \leftarrow V(s)$   
|      $V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$   
|      $\Delta \leftarrow \max(\Delta, |v - V(s)|)$ 
```

until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that

$$\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$$

How I tried to solve it?

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

 Initialize S

 Loop for each step of episode:

 Choose A from S using policy derived from Q (e.g., ε -greedy)

 Take action A , observe R, S'

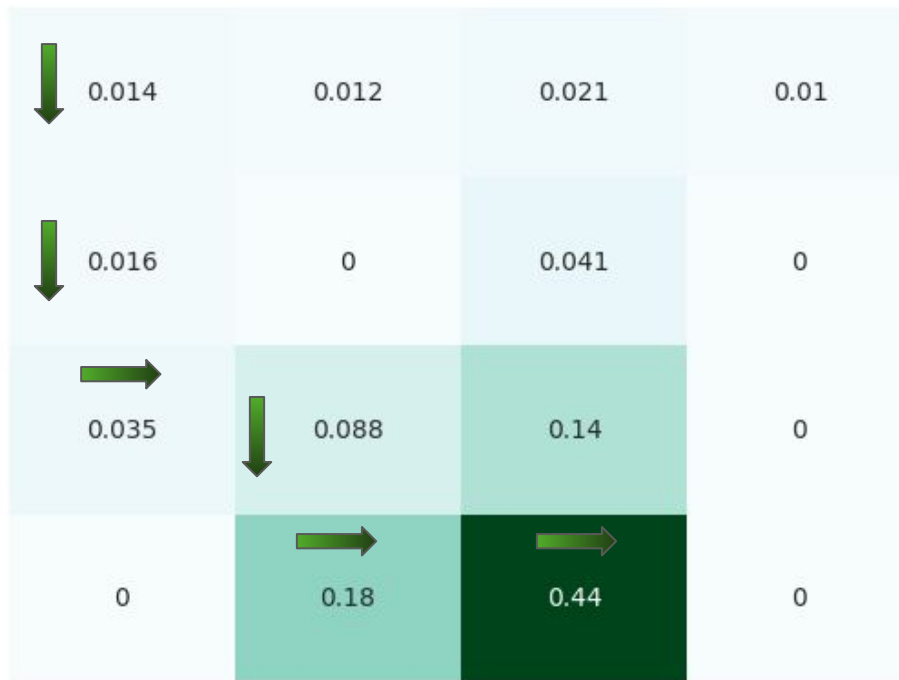
$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S, A)]$

$S \leftarrow S'$

 until S is terminal

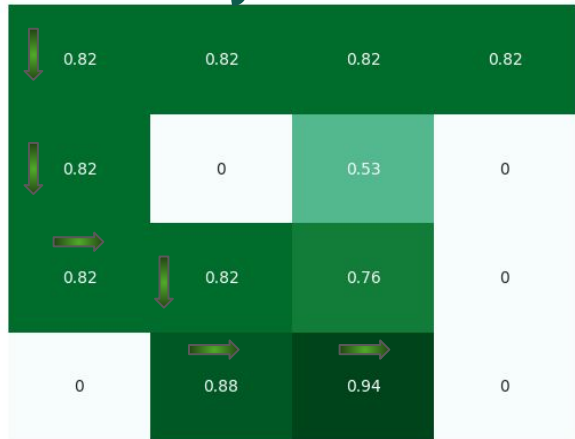
Results Slippery

Policy Evaluation

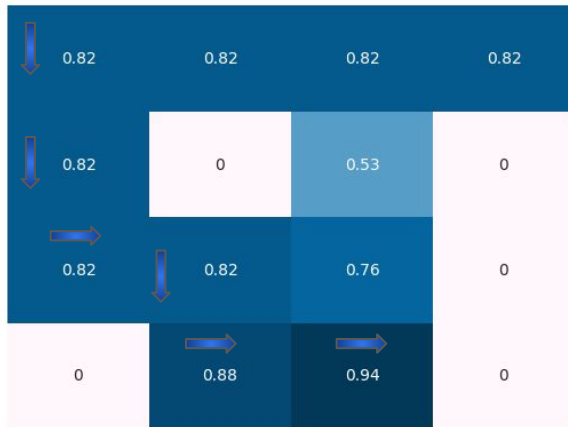


Optimal policies for 3 (4x4)

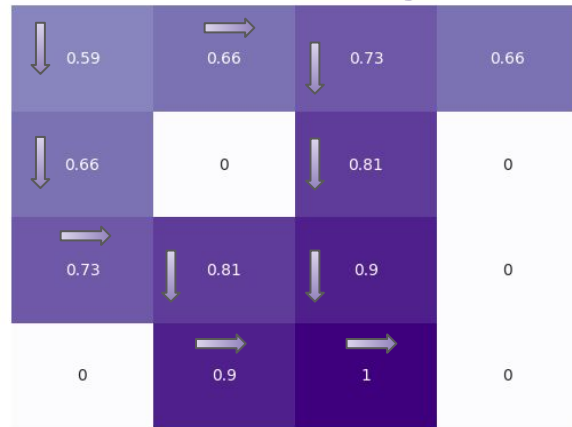
Policy Iteration



Value Iteration



Q-Learning



Optimal policies for 3 (8x8)

Policy Iteration

→ 1	→ 1	→ 1	→ 1	→ 1	→ 1	→ 1	→ 1
1	1	1	1	1	1	1	↓ 1
1	0.98	0.93	0	0.86	0.95	0.98	↓ 1
1	0.93	0.8	0.47	0.62	0	0.94	↓ 1
1	0.83	0.54	0	0.54	0.61	0.85	↓ 1
1	0	0	0.17	0.38	0.44	0	↓ 1
1	0	0.19	0.12	0	0.33	0	↓ 1
1	0.73	0.46	0	0.28	0.55	0.78	0

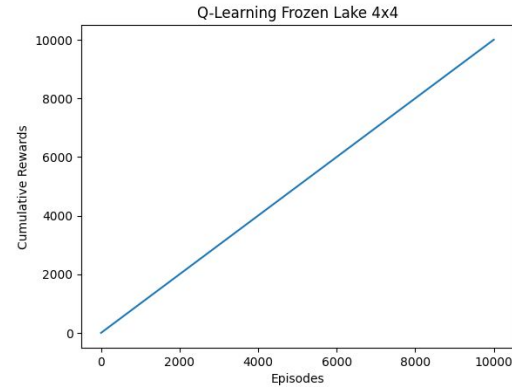
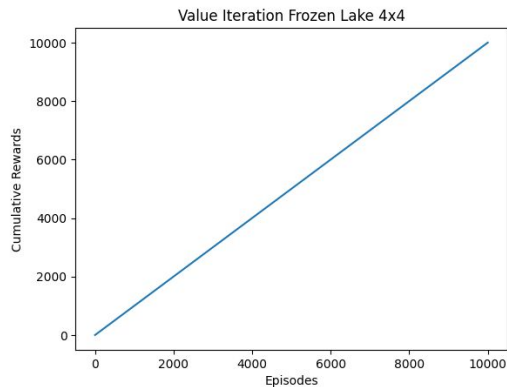
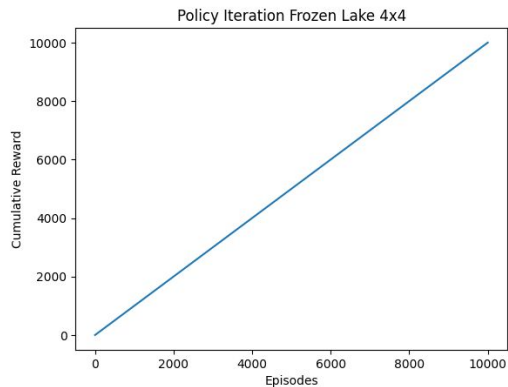
Value Iteration

→ 1	→ 1	→ 1	→ 1	→ 1	→ 1	→ 1	↓ 1
1	1	1	1	1	1	1	↓ 1
1	0.98	0.93	0	0.86	0.95	0.98	↓ 1
1	0.93	0.8	0.47	0.62	0	0.94	↓ 1
1	0.83	0.54	0	0.54	0.61	0.85	↓ 1
1	0	0	0.17	0.38	0.44	0	↓ 1
1	0	0.19	0.12	0	0.33	0	↓ 1
1	0.73	0.46	0	0.28	0.55	0.78	0

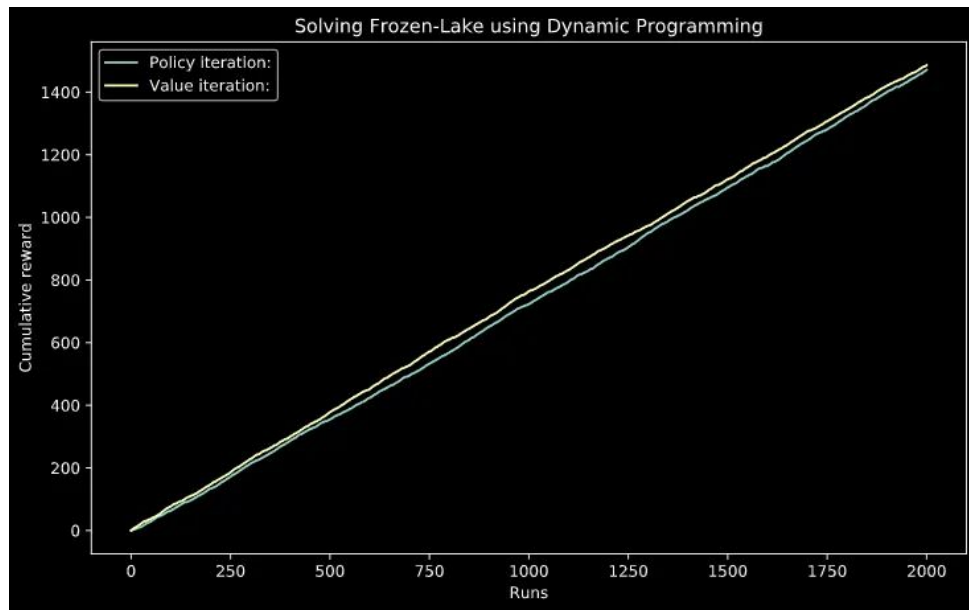
Q-Learning

→ 0.0022	→ 0.003	→ 0.004	→ 0.0057	→ 0.0083	→ 0.0076	→ 0.0086	↓ 0.0075
0.0017	0.0016	0.0029	0.0047	0.0088	0.0078	0.012	↓ 0.012
0.0029	0.0056	0.0012	0	0.0066	0.013	0.016	↓ 0.028
8.1e-05	0.00041	1.6e-05	4.8e-05	0.0029	0	0.026	↓ 0.067
4.3e-05	4.1e-05	9e-06	0	0.022	0.018	0.046	↓ 0.24
3.7e-05	0	0	1.6e-07	0.0015	0.0091	0	↓ 0.6
3.2e-05	0	2.5e-06	1.5e-05	0	0.072	0	↓ 0.91
4.2e-05	3.3e-05	3.1e-05	0	8.3e-06	0.87	0.8	0

Comparing Cumulative Rewards

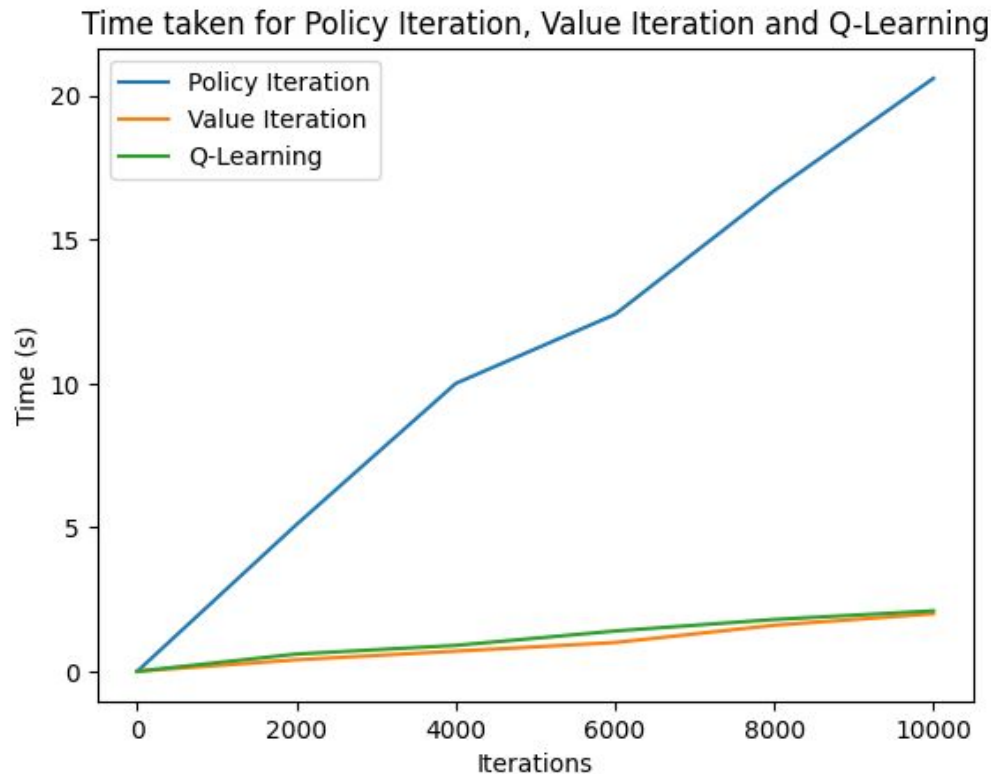


Comparing Cumulative Rewards

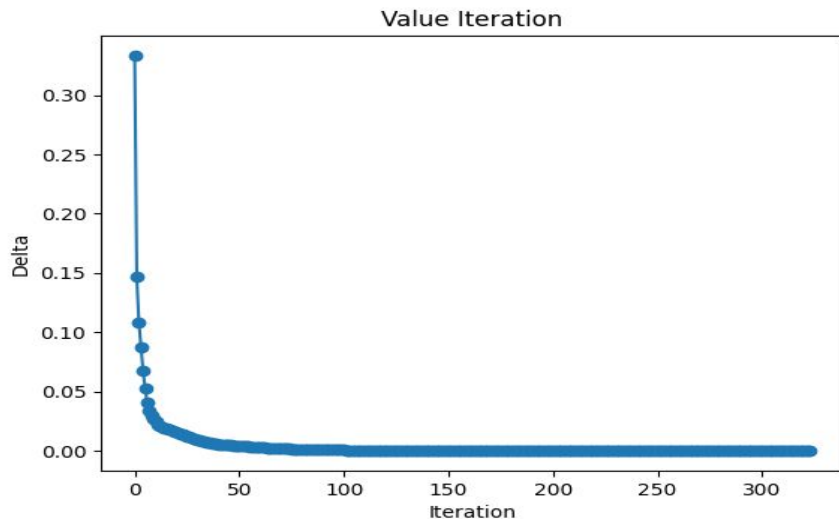
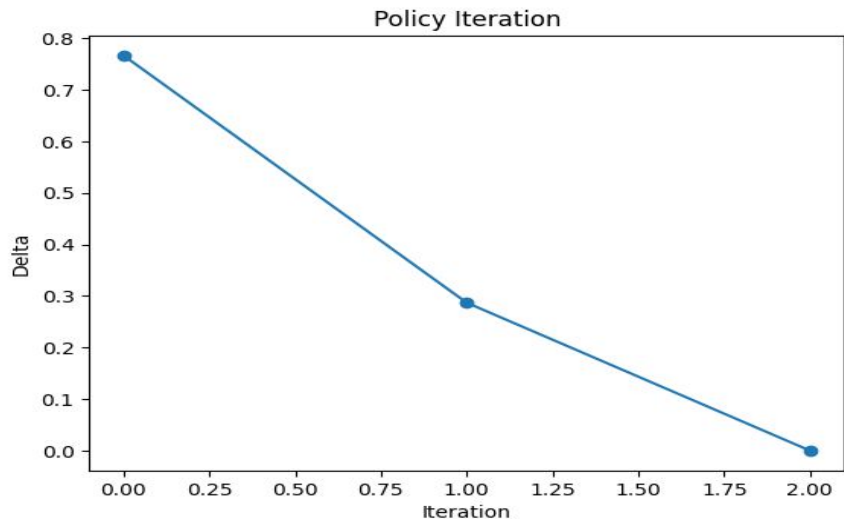


Taken from [5]

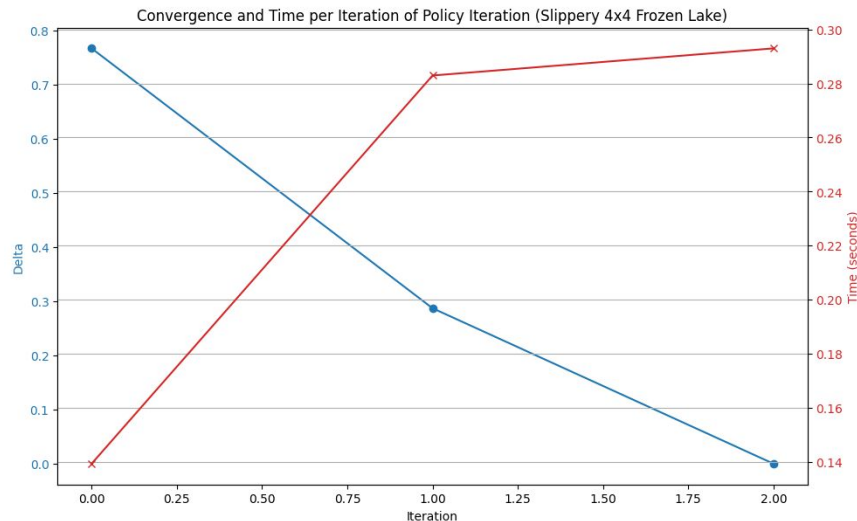
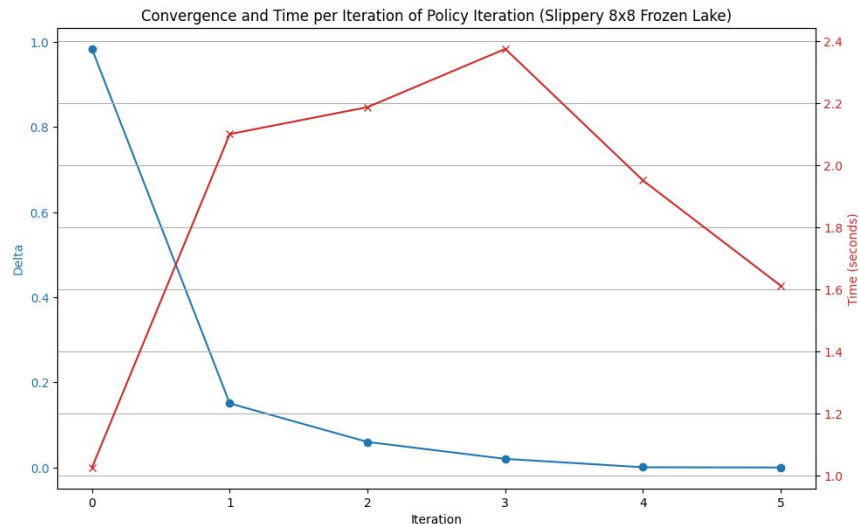
Comparing Executing Time



Convergence with respect to Delta

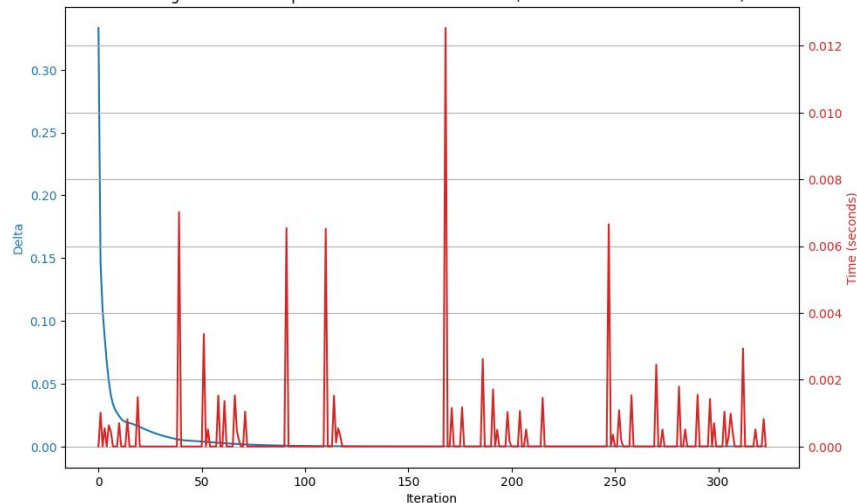


Policy Iteration

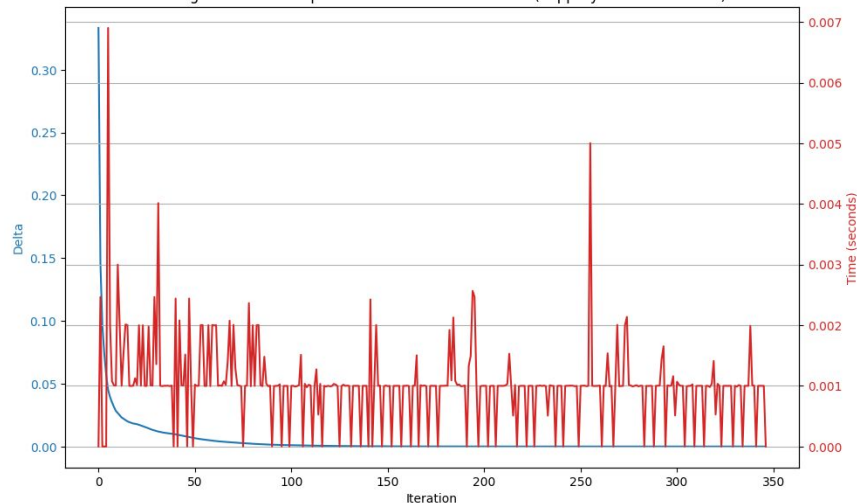


Value Iteration

Convergence and Time per Iteration of Value Iteration (Deterministic 8x8 Frozen Lake)

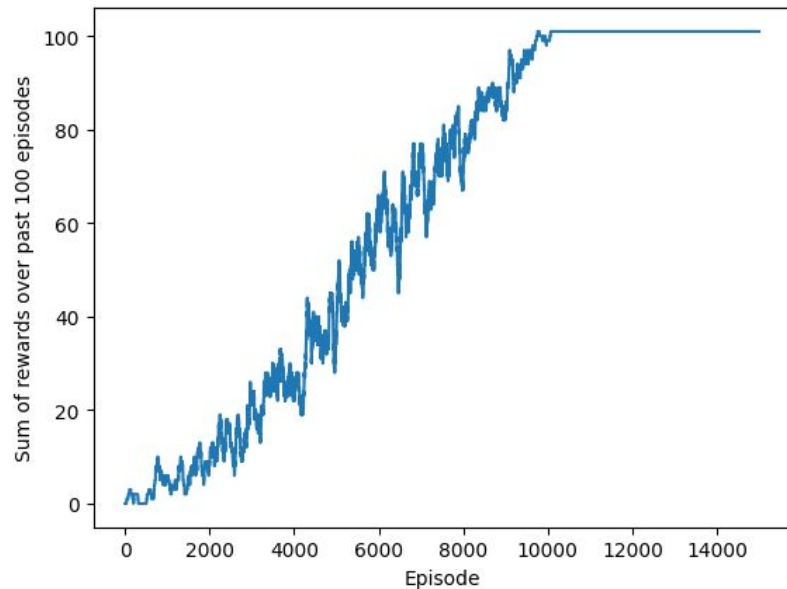


Convergence and Time per Iteration of Value Iteration (Slippery 4x4 Frozen Lake)



Q-Learning Deterministic

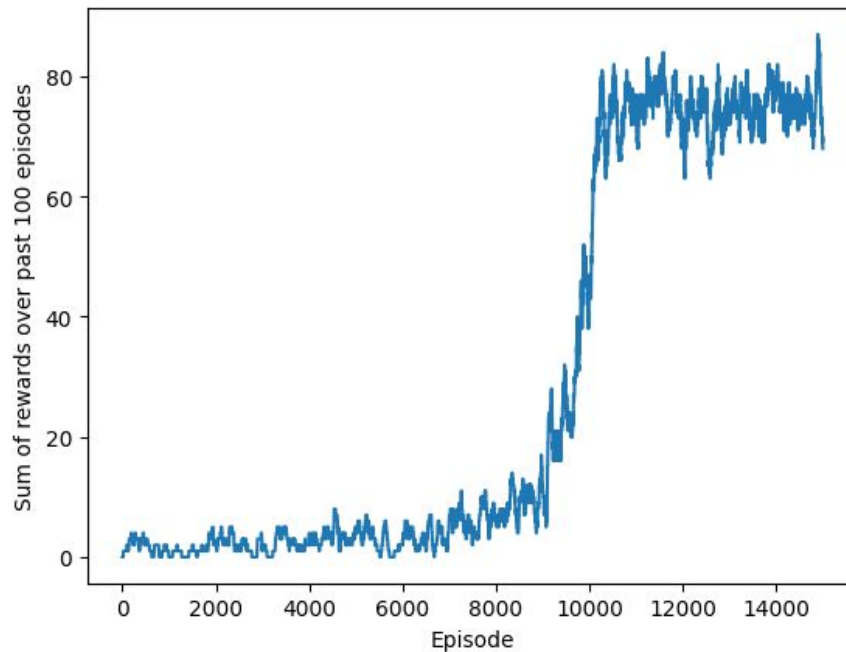
0	0.59	0.66	0.73	0.66
1	0.66	0	0.81	0
2	0.73	0.81	0.9	0
3	0	0.9	1	0
	0	1	2	3



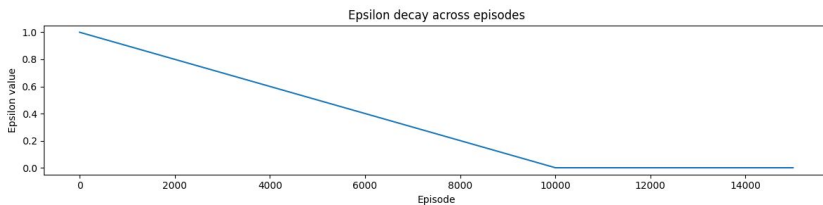
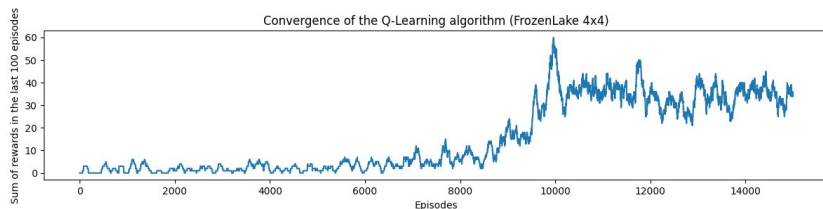
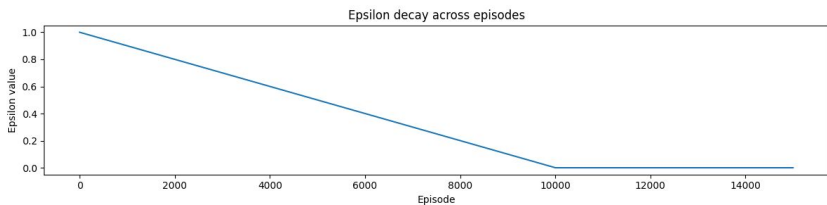
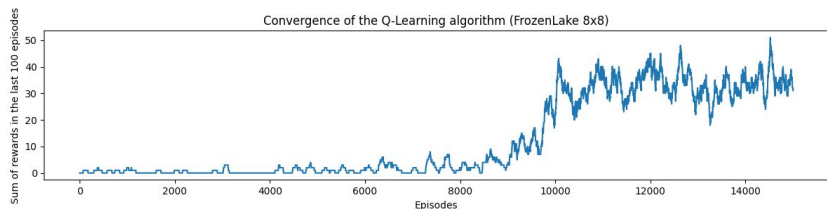
Q-Learning - Rewards Stochastic

0	0.035	0.003	0.0012	0.001
1	0.065	0	0.014	0
2	0.14	0.31	0.23	0
3	0	0.68	0.77	0
	0	1	2	3

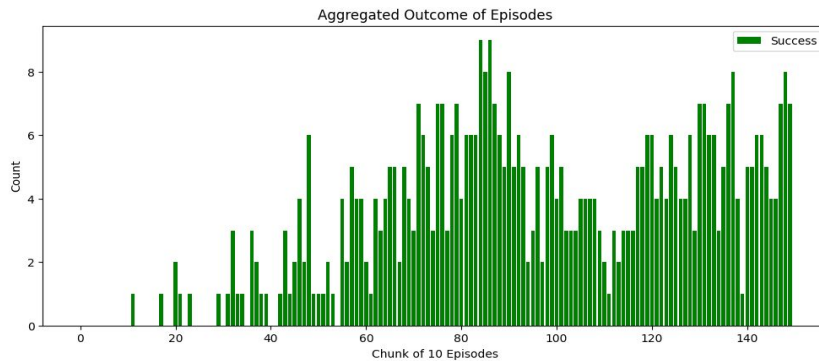
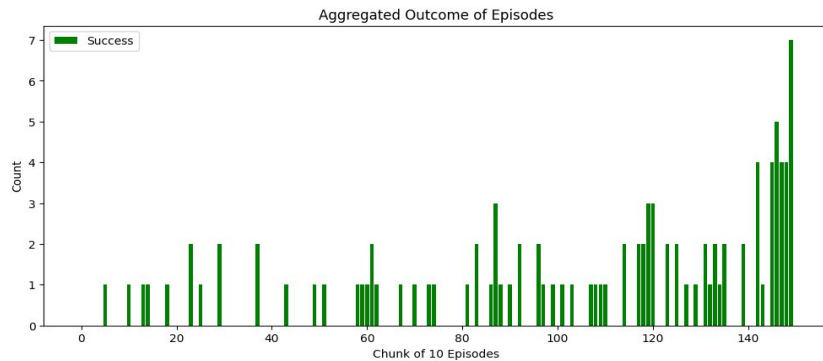
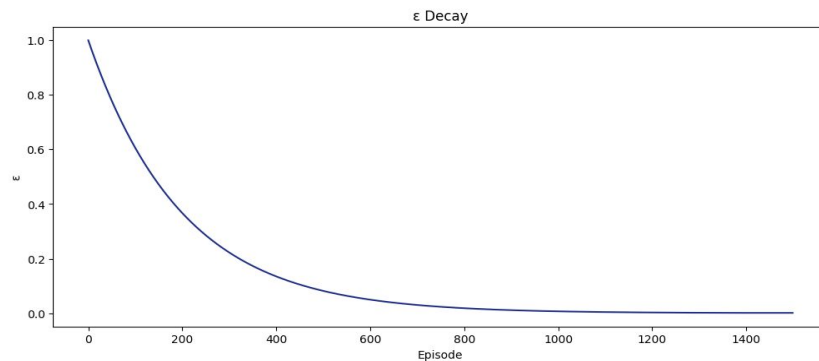
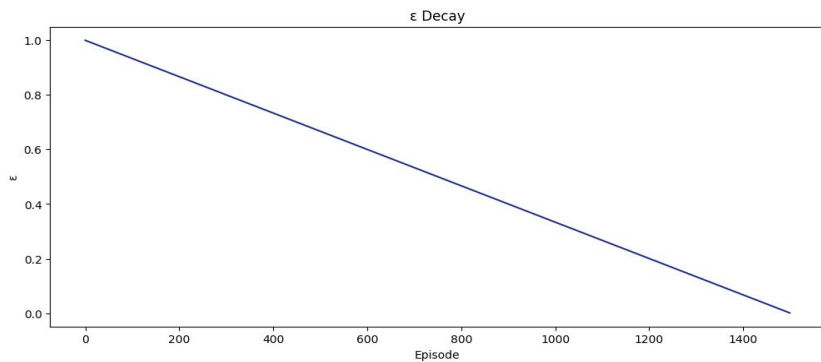
0	0.054	0.01	0.0048	0.00084
1	0.068	0	0.038	0
2	0.1	0.18	0.37	0
3	0	0.14	0.38	0
	0	1	2	3



Q-Learning



Dilemma



Conclusions

- 1) The Frozen Lake problem can be modeled by a model-based (policy iteration and value iteration) or by a model-free (Q-learning) algorithm.
- 2) There is not a big difference between 4x4 and 8x8 grid results.
- 3) Executing time is very similar for the three algorithms. However, we should rank from fastest to slowest: value iteration, Q-learning, and policy iteration.
- 4) In the exploration-exploitation dilemma, exponential decay performs better than linear decay.

What more can I do?

- Generalized Policy Iteration.
- Asynchronous Dynamic Programming.
- Explore more shapes of epsilon.
- Fix my mistakes, i.e., improve the code.

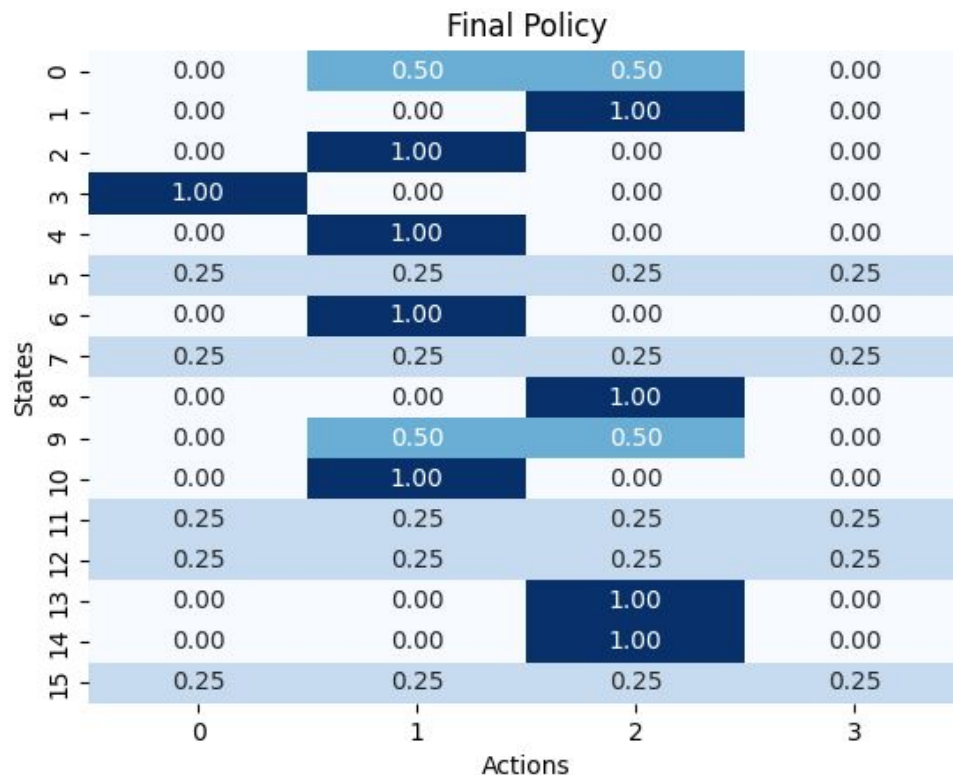
**THANK YOU SO
MUCH!**

References

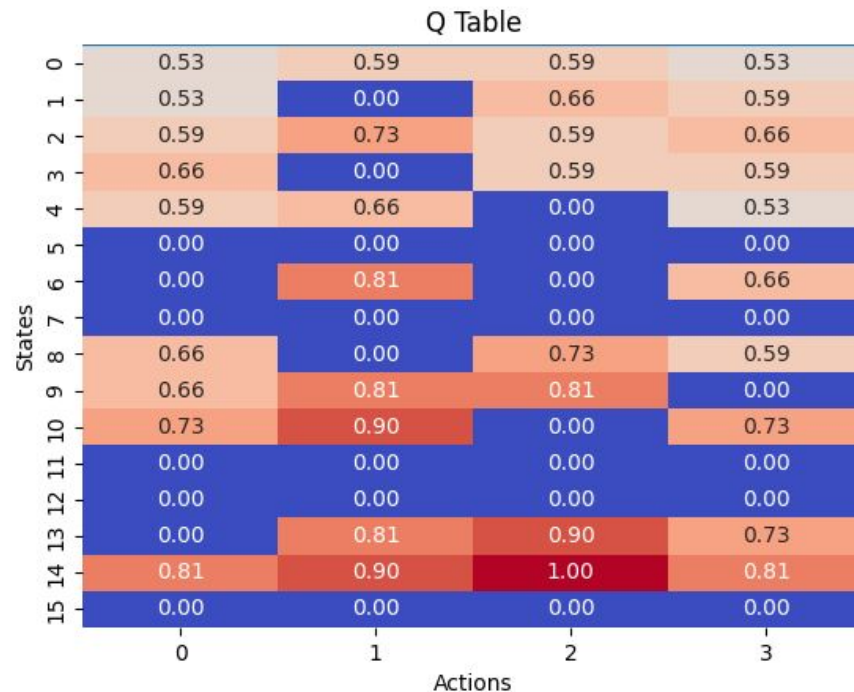
- 1) Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction, second edition*. The MIT Press, Cambridge, Massachusetts, London, England, 2018, 2020.
- 2) <https://karan-jakhar.medium.com/100-days-of-code-day-1-35afe174000e>
- 3) https://gymnasium.farama.org/environments/toy_text/frozen_lake/#frozen-lake
- 4) <https://karan-jakhar.medium.com/100-days-of-code-day-1-35afe174000e>
- 5) <https://medium.com/swlh/frozen-lake-as-a-markov-decision-process-1692815ecfd1>

NOTEBOOK TOUR!

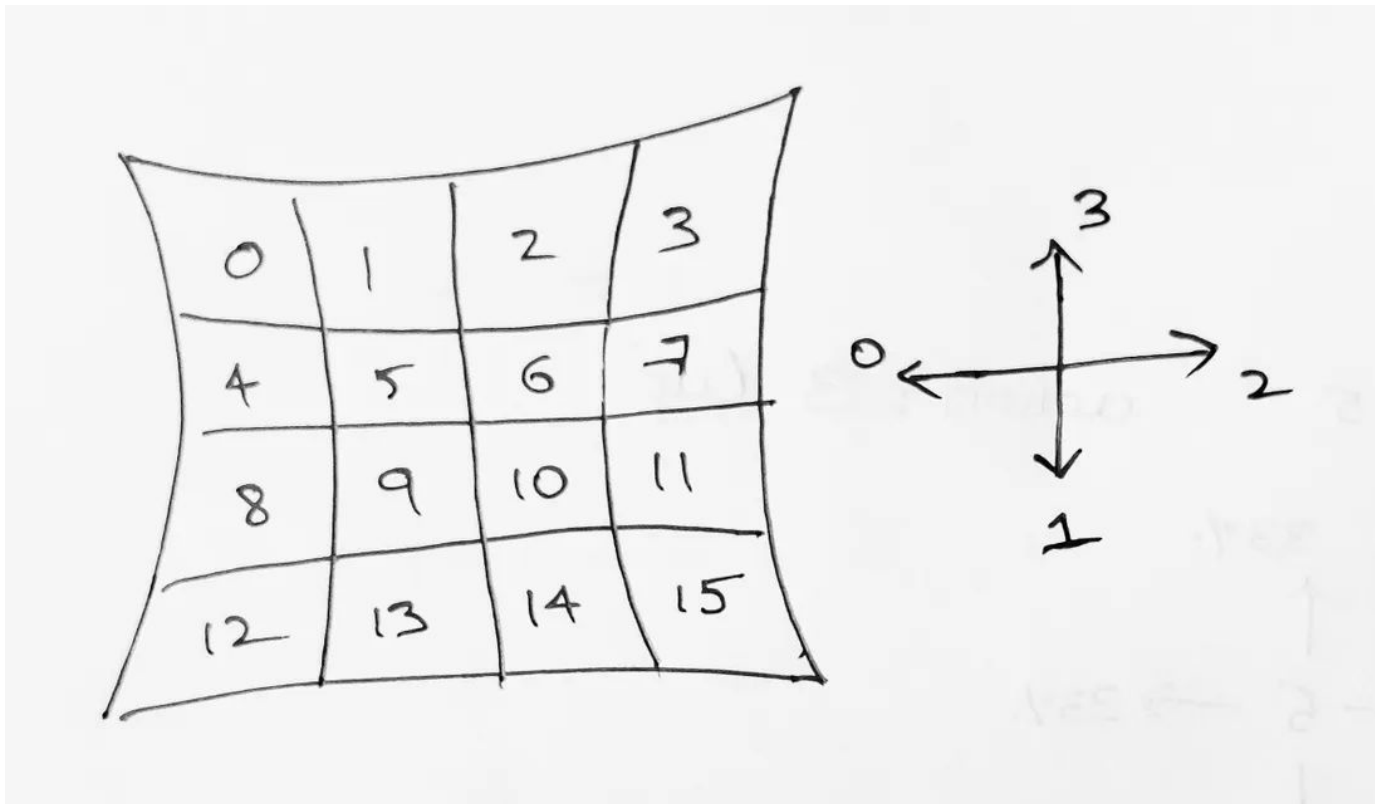
Policy Iteration



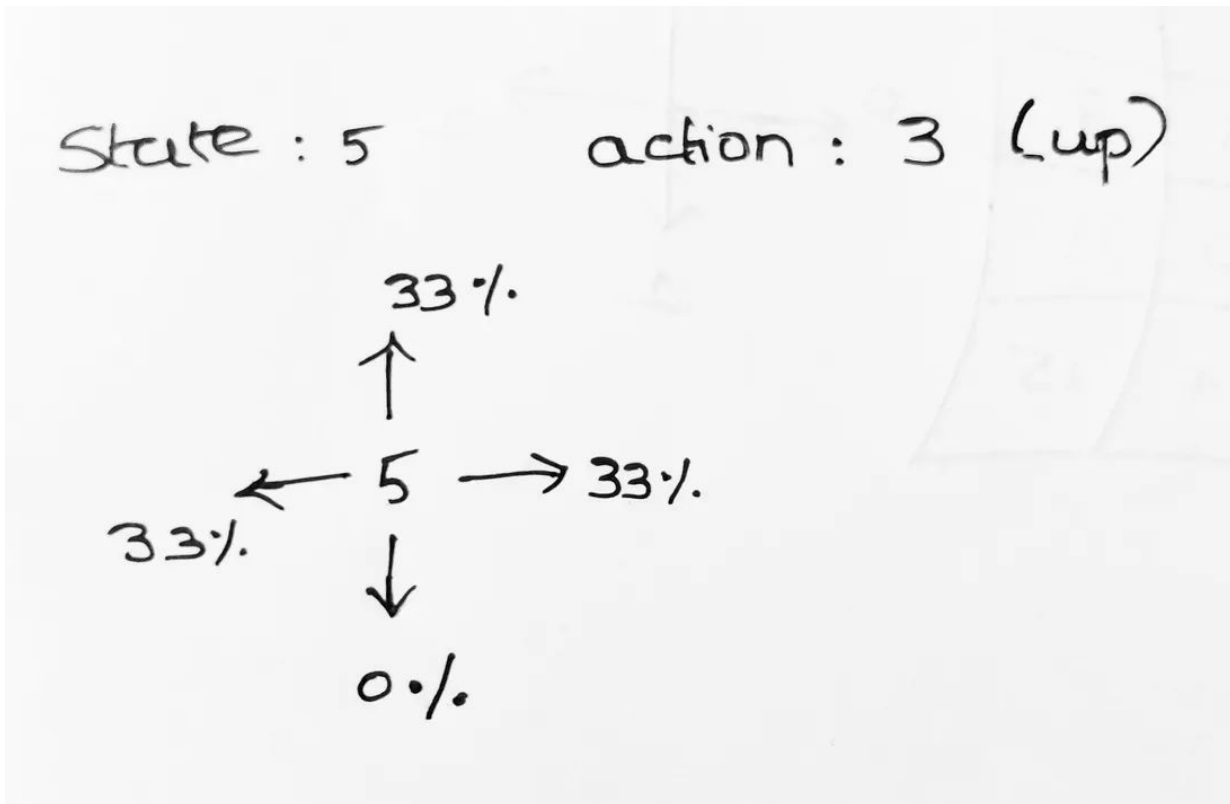
Q-Learning



Frozen Lake as an MDP



Frozen Lake as an MDP



Frozen Lake as an MDP

SFFF

FHFH

FFFH

HFFG

Number of states 16

Number of actions 4

Transitions for state 0 and action LEFT are

[(0.33, 0, 0.0, False), (0.33, 0, 0.0, False), (0.33, 4, 0.0, False)]

Transitions for state 0 and action UP are

[(0.33, 1, 0.0, False), (0.33, 0, 0.0, False), (0.33, 0, 0.0, False)]

Transitions for state 11 and action LEFT are

[(1.0, 11, 0, True)]

Transitions for state 11 and action UP are

[(1.0, 11, 0, True)]

Transitions for state 15 and action LEFT are

[(1.0, 15, 0, True)]

Transitions for state 15 and action UP are

[(1.0, 15, 0, True)]