ICTP The Abdus Salam International Centre for Theoretical Physics

QUANTITATIVE LIFE SCIENCES

The Abdus Salam International Centre for Theoretical Physics

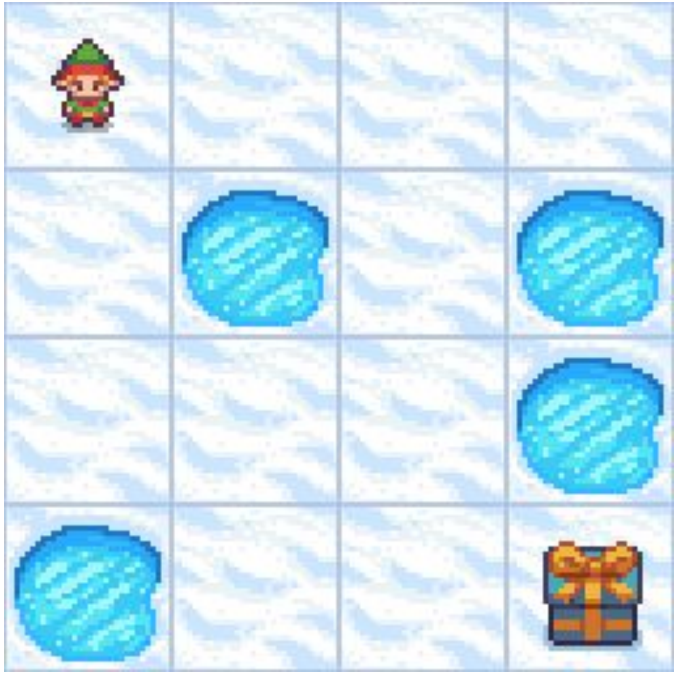
QUANTITATIVE LIFE SCIENCES

Frozen Lake

SA Torres Orozco

Description of the problem

Description of the problem



Motivation for the problem

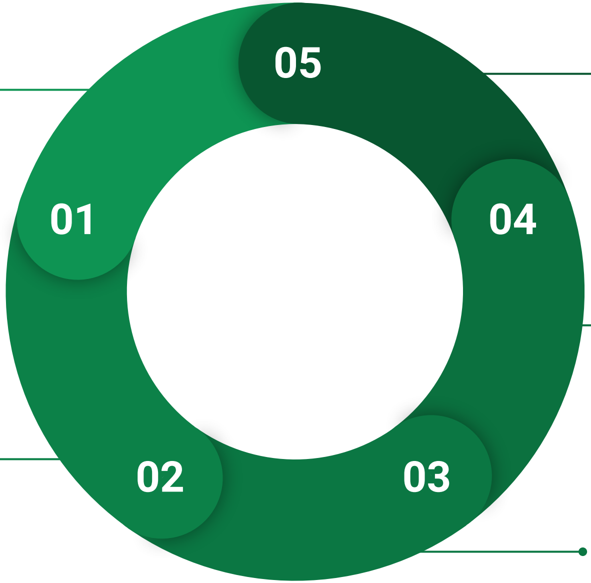
- Simplicity and Clarity
- Introduction to Key Concepts
- Educational Value

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



Approach and Methods
Explanation of the methods

Results
Analysis of the outcomes and their interpretation

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

OBJECTIVES

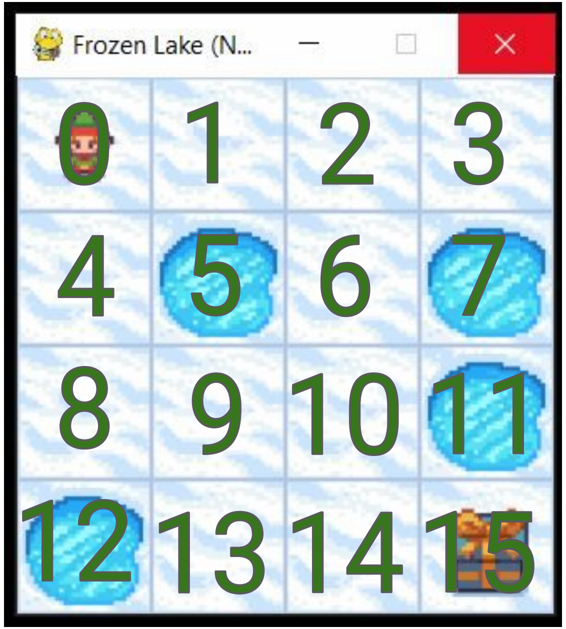
- Determine **Optimal Policies** Using **Different Methods**:
  Policy Iteration.
  Value Iteration.
  Q-Learning.

- Compare and **Analyze Policies.**

- **Evaluate the performance** of each method in terms of convergence speed, policy quality, and robustness in deterministic and stochastic environments.

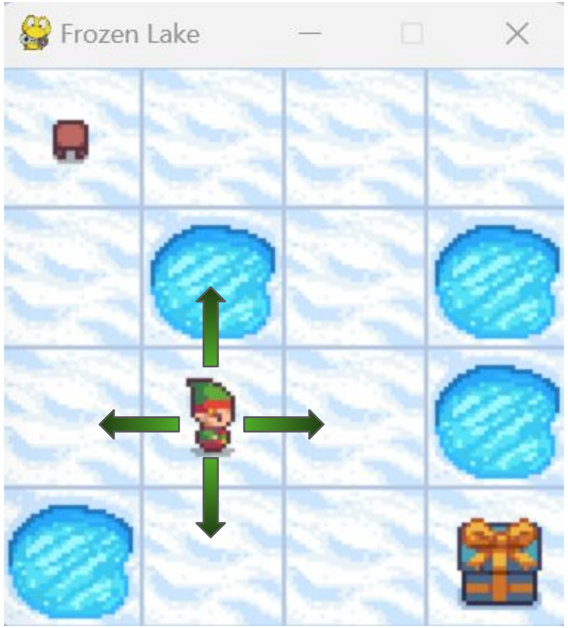
Reinforcement learning framework

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



Reinforcement learning framework

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



Reinforcement learning framework

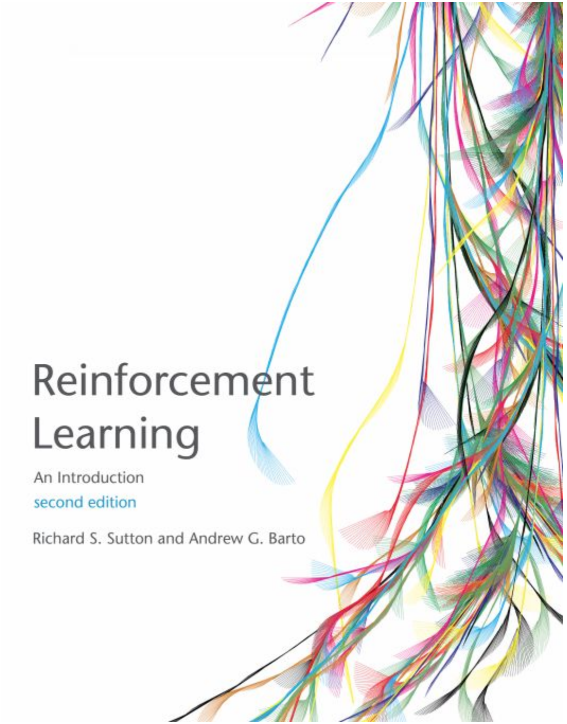
- **States;** coordinates in the grid {(0,0), ..., (3,3)}
  S = {0, 1, 2, ..., 15}
- **Actions;** A = {left, down, right, up}
  A = {0, 1, 2, 3}
- **Transitions/Dynamics**
  Deterministic → *100 % probability*
  Stochastic → *33 % probability*

Reinforcement learning framework

- **States;** coordinates in the grid {(0,0), ..., (3,3)}
  S = {0, 1, 2, ..., 15}
- **Actions;** A = {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 $s \in \mathcal{S}$:
     - $v \leftarrow V(s)$
     - $V(s) \leftarrow \sum_{s',r} p(s',r|s,\pi(s))\big[r + \gamma V(s')\big]$
     - $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

   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* $\leftarrow \pi(s)$
   - $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
   - 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 $V(terminal) = 0$

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

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$

How I tried to solve it?

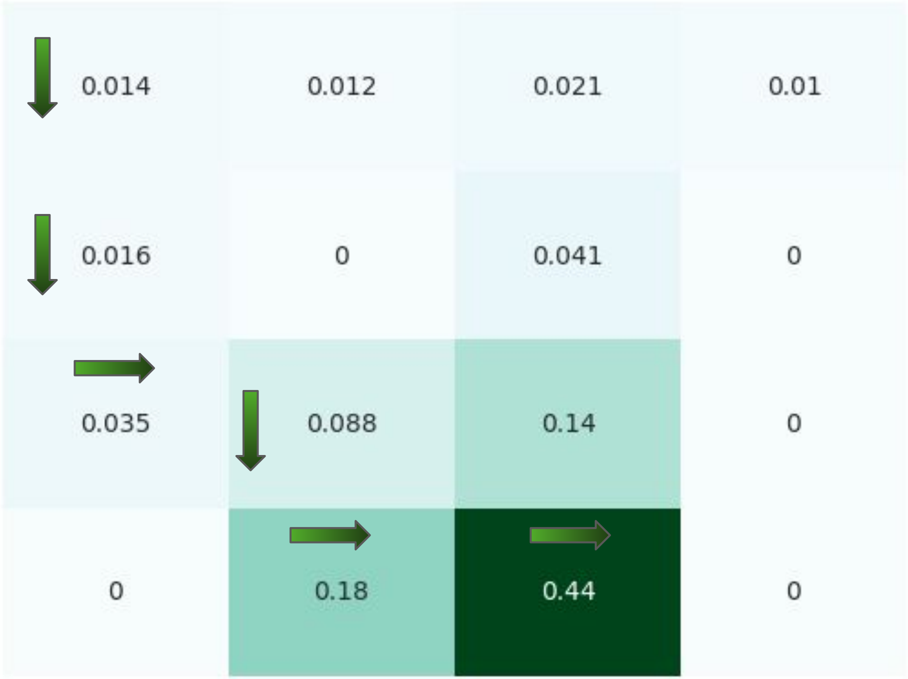
Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

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 $Q(terminal, \cdot) = 0$

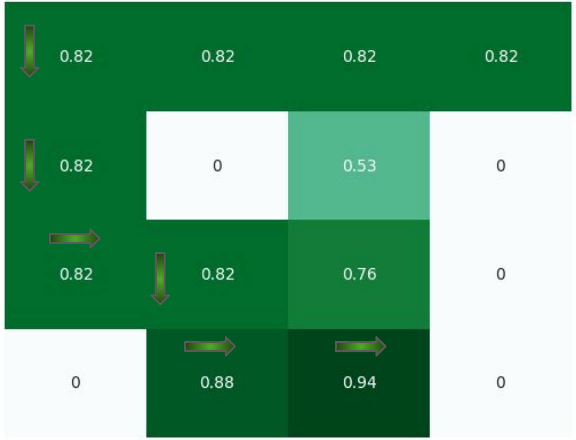
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

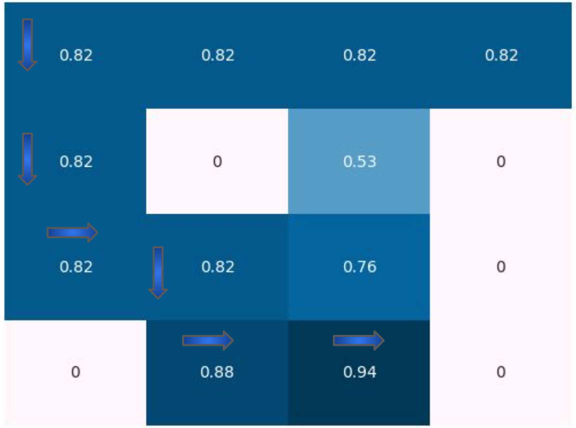
Results
Slippery

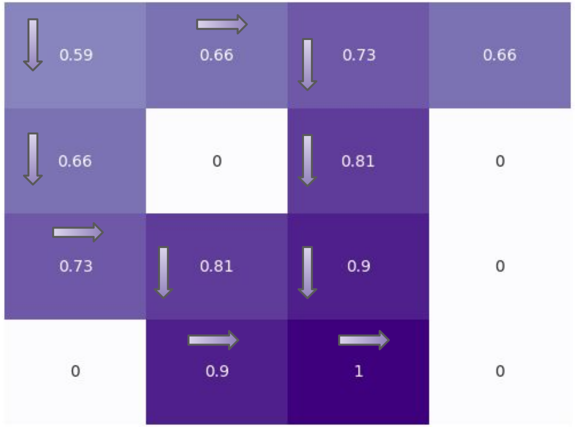
Policy Evaluation



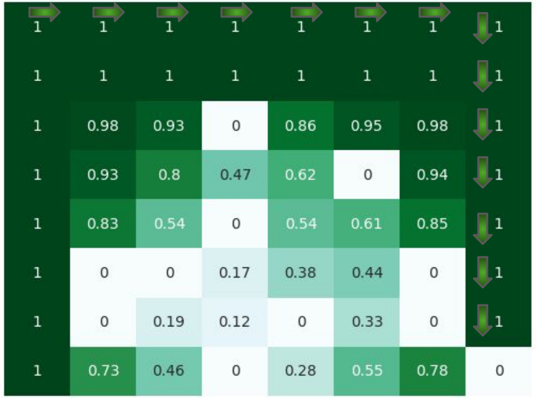
Optimal policies for 3 (4x4)

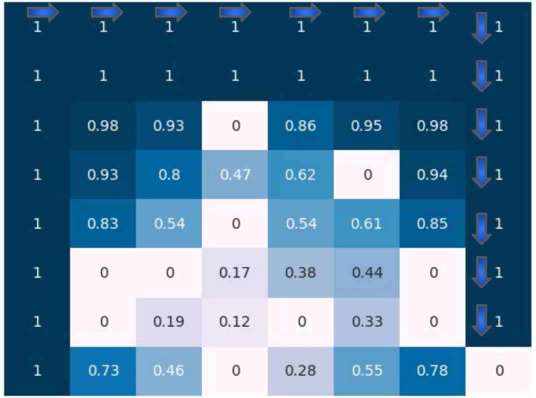


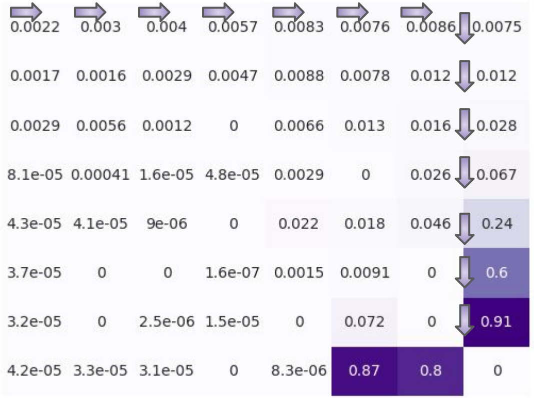




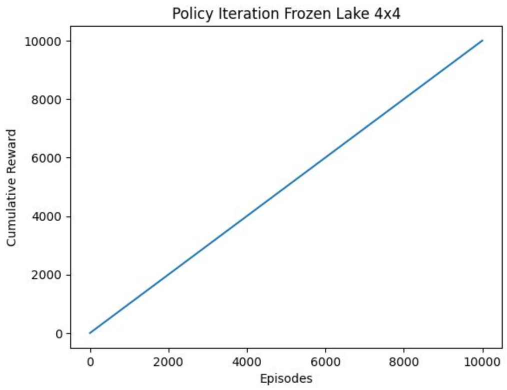
Optimal policies for 3 (8x8)

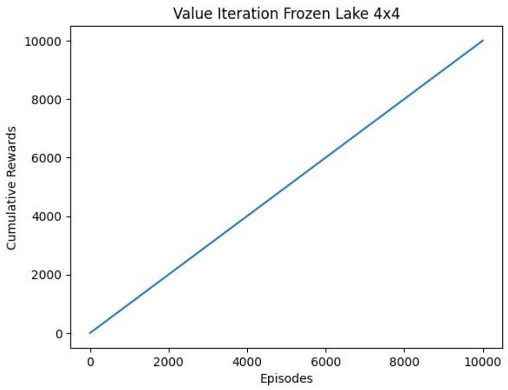


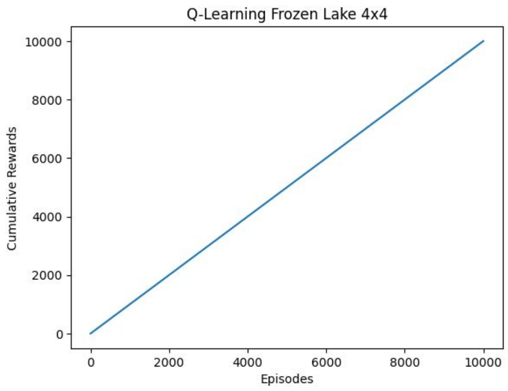




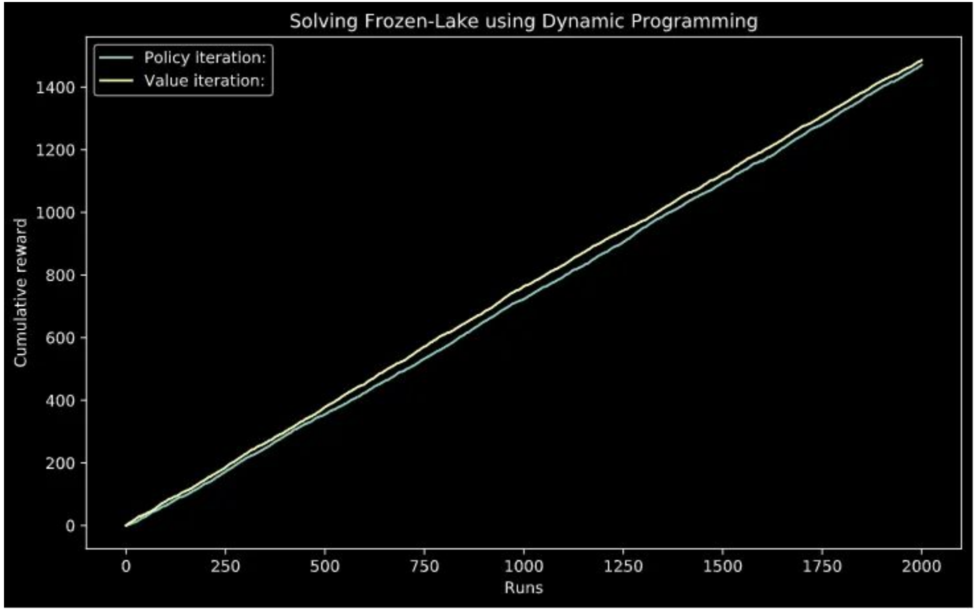
Comparing Cumulative Rewards





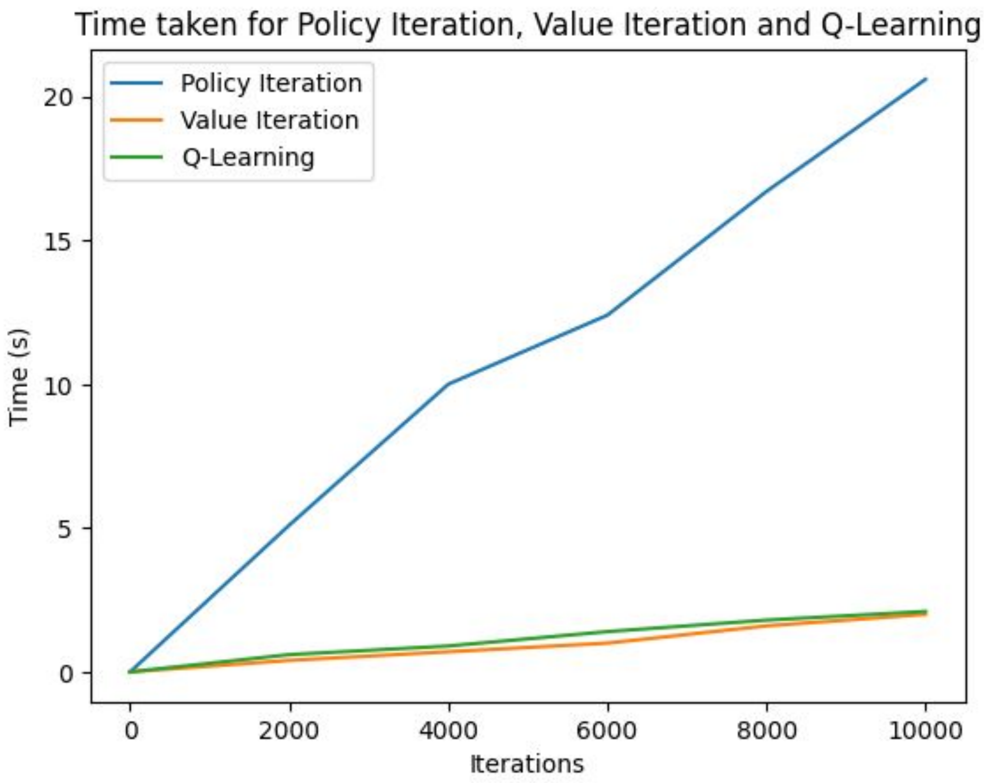


Comparing Cumulative Rewards

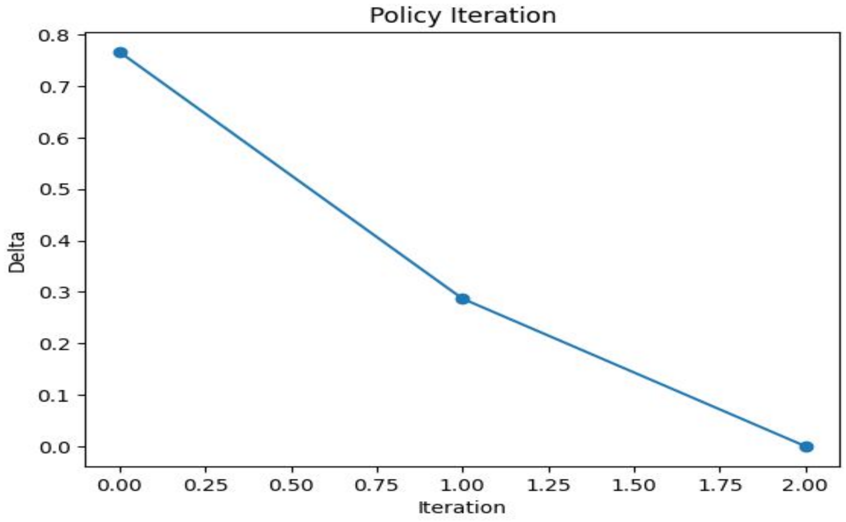


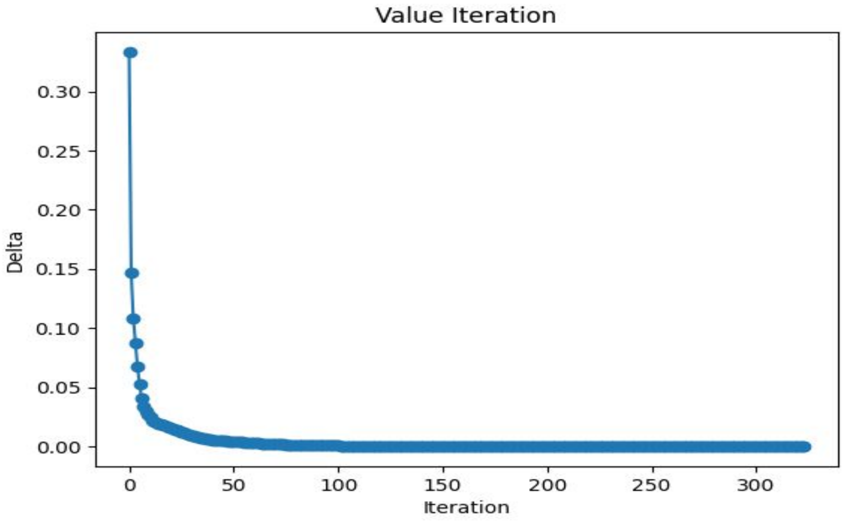
Taken from [5]

Comparing Executing Time

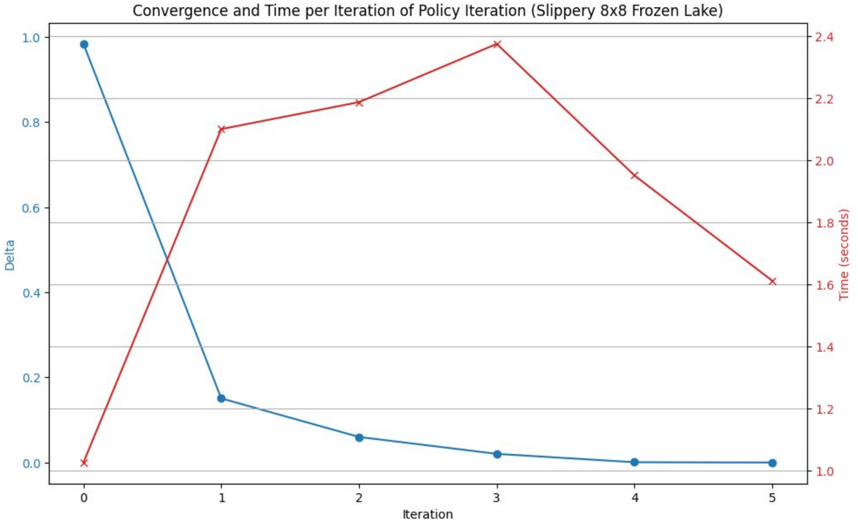


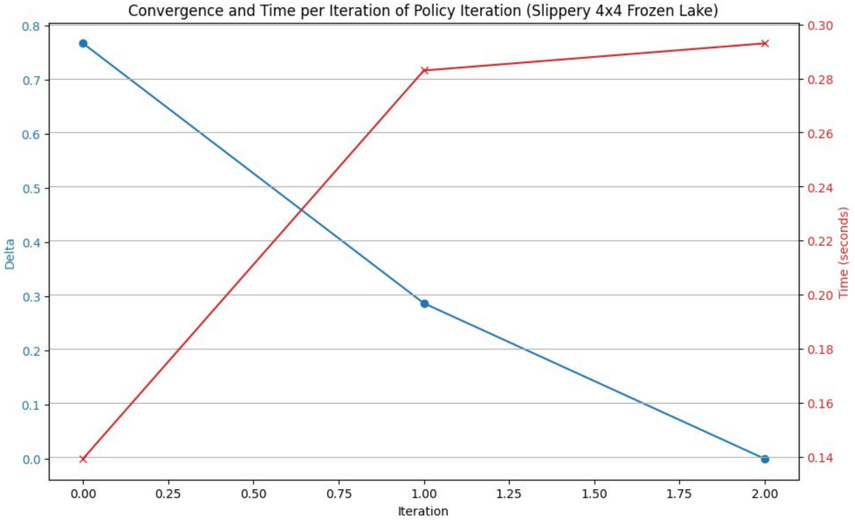
Convergence with respect to Delta



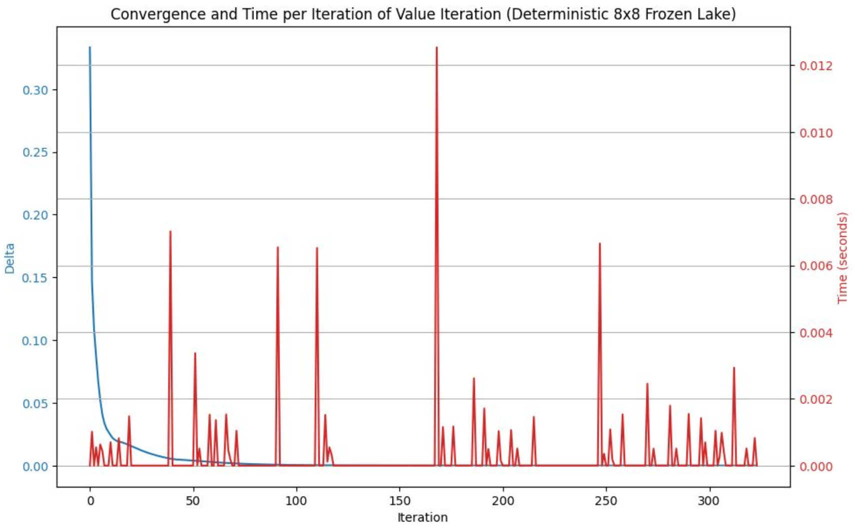


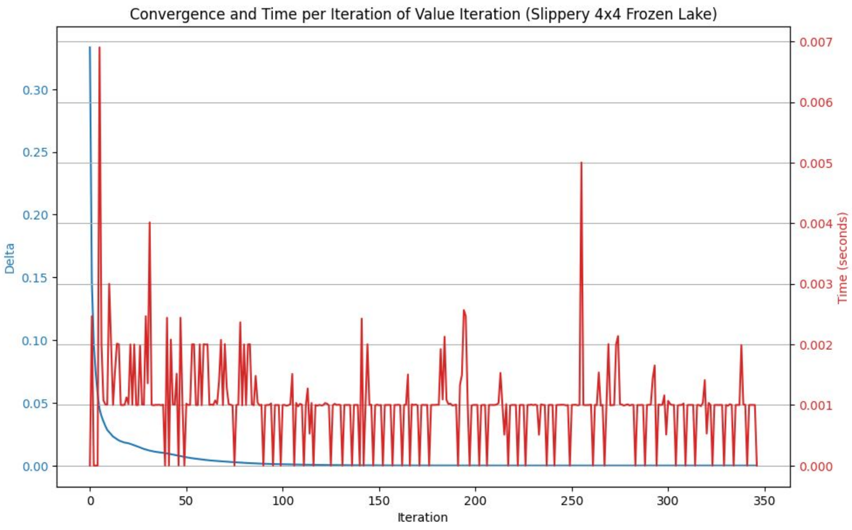
Policy Iteration



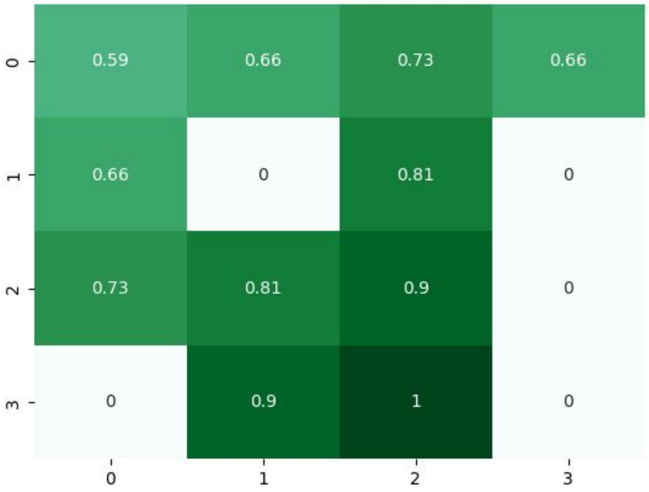


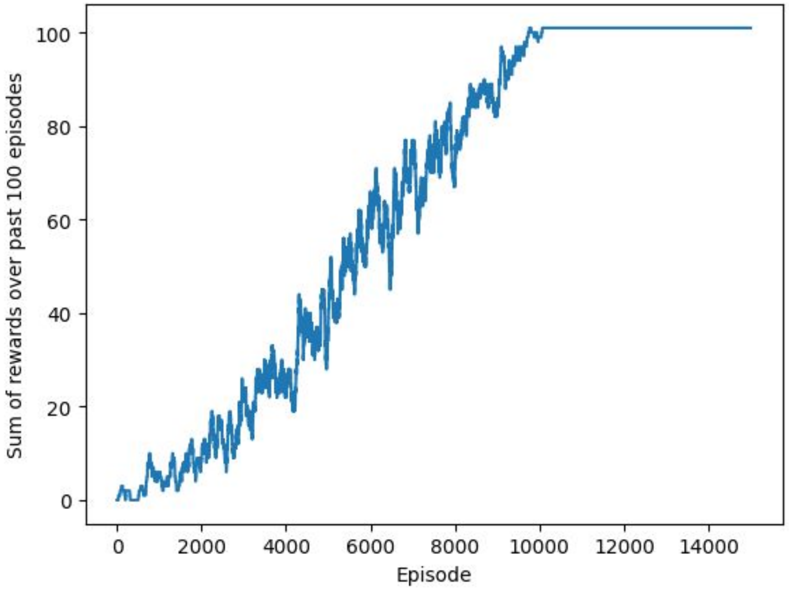
Value Iteration



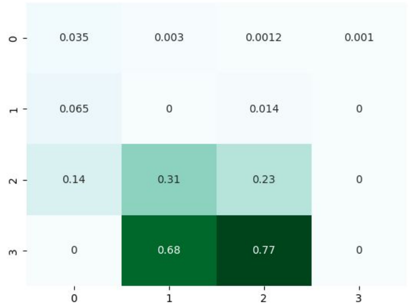


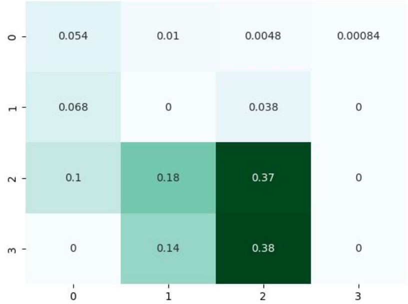
Q-Learning Deterministic

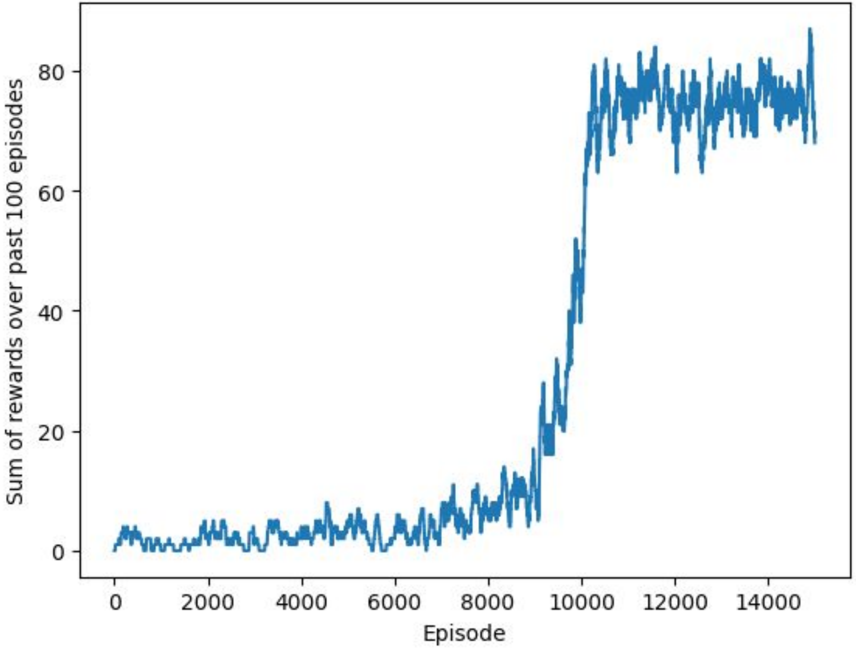




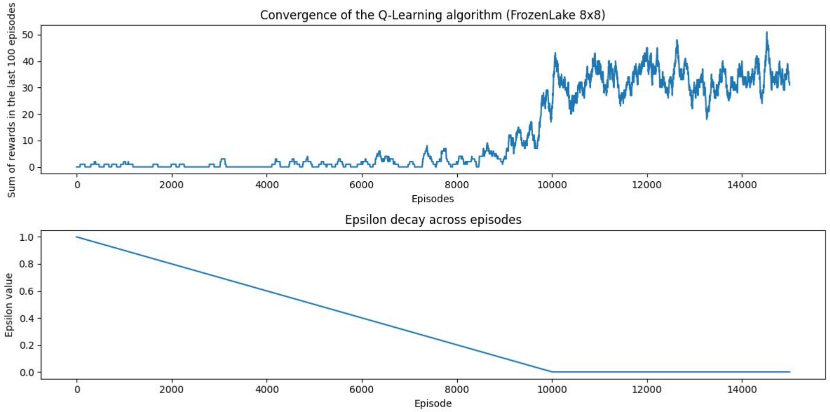
Q-Learning - Rewards Stochastic

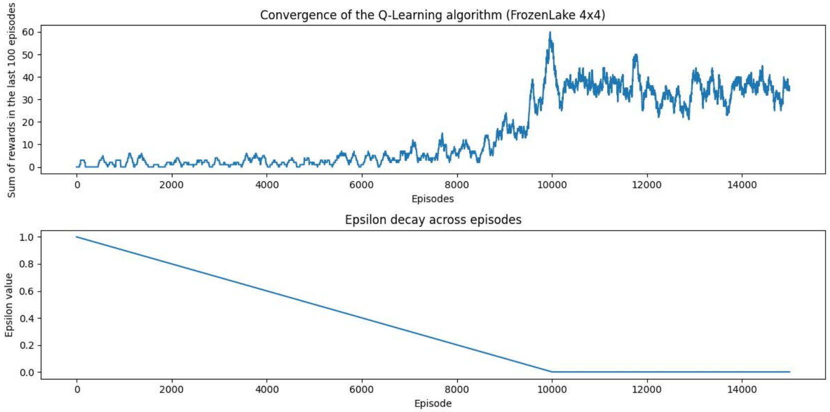




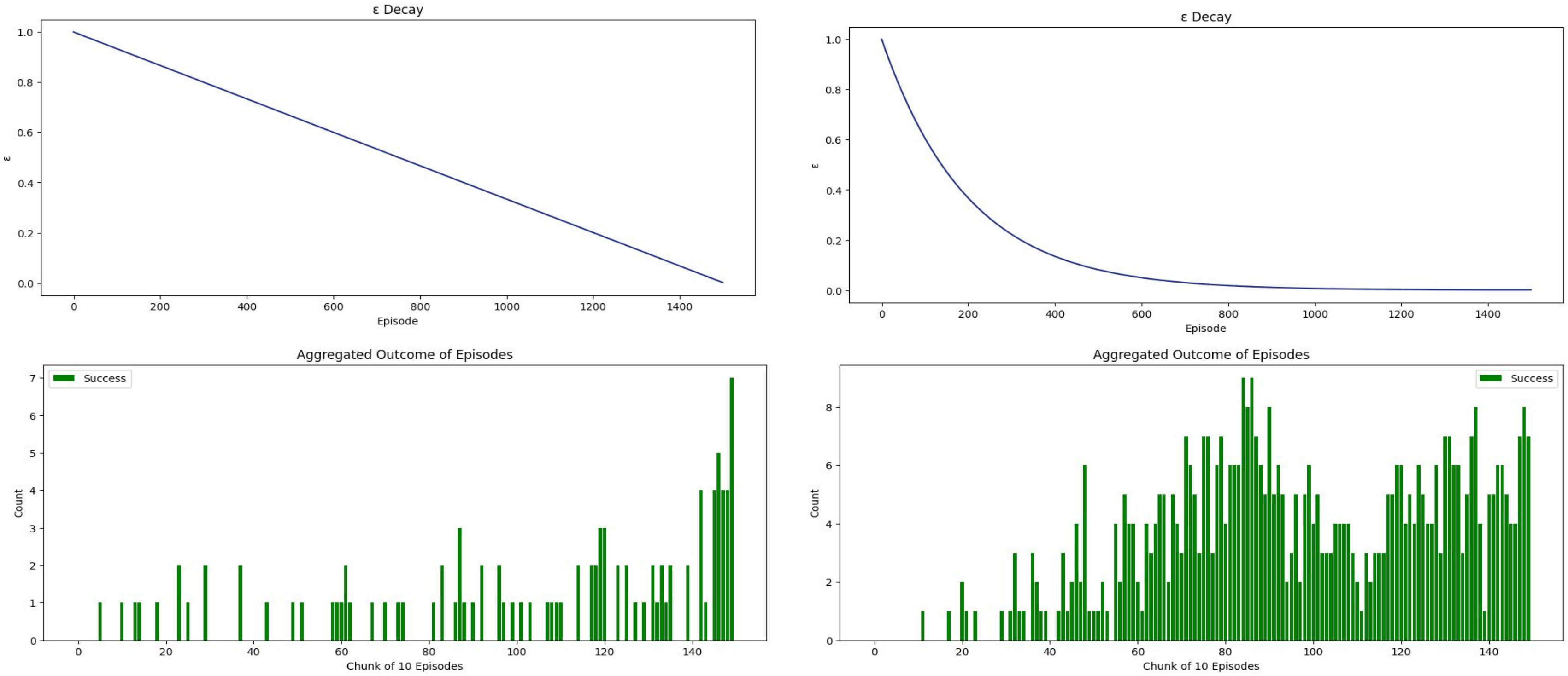


Q-Learning


Convergence of the Q-Learning algorithm (FrozenLake 8x8)
Sum of rewards in the last 100 episodes
Episodes
Epsilon decay across episodes
Epsilon value
Episode


Convergence of the Q-Learning algorithm (FrozenLake 4x4)
Sum of rewards in the last 100 episodes
Episodes
Epsilon decay across episodes
Epsilon value
Episode

Dilemma


ε Decay
ε
Episode
Aggregated Outcome of Episodes
Success
Count
Chunk of 10 Episodes
ε Decay
ε
Episode
Aggregated Outcome of Episodes
Success
Count
Chunk of 10 Episodes

Conclusions

- Frozen Lake problem can be modelling by a model-free (policy iteration and value iteration) or by a model-based (Q-learning) algorithms.
- There is not big difference between 4x4 and 8x8 grid results
- Executing time is very similar for three algorithms and this is because the problem is simple. However, if we should to rank from fastest to slowest: value iteration, Q-learning, policy iteration.
- In 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

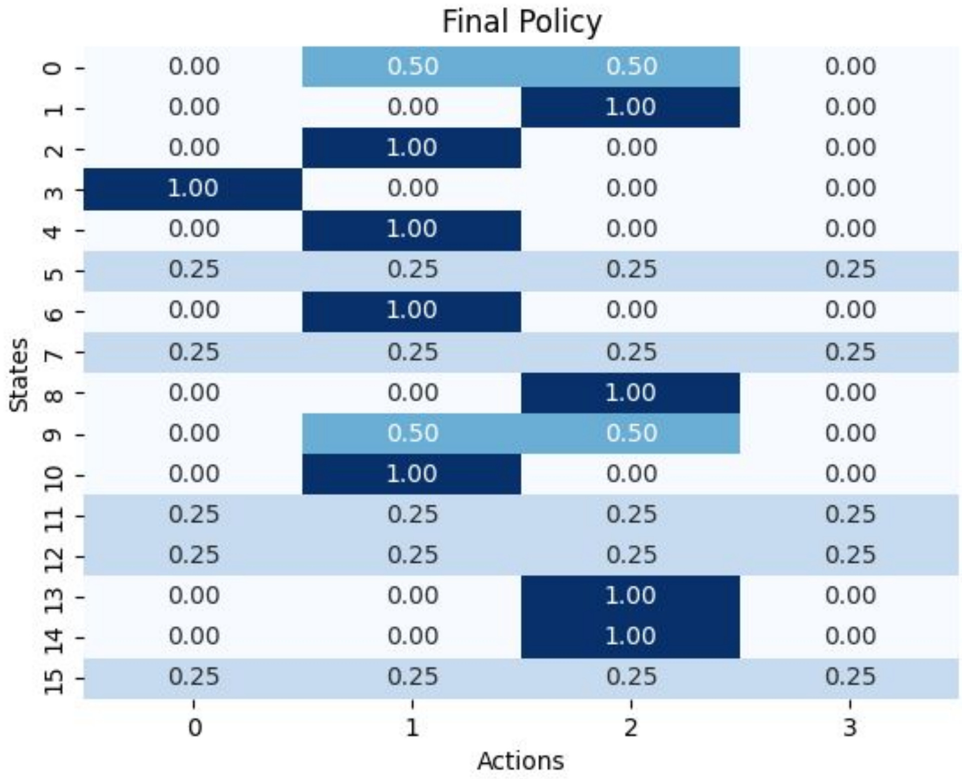
THANK YOU SO MUCH!

References

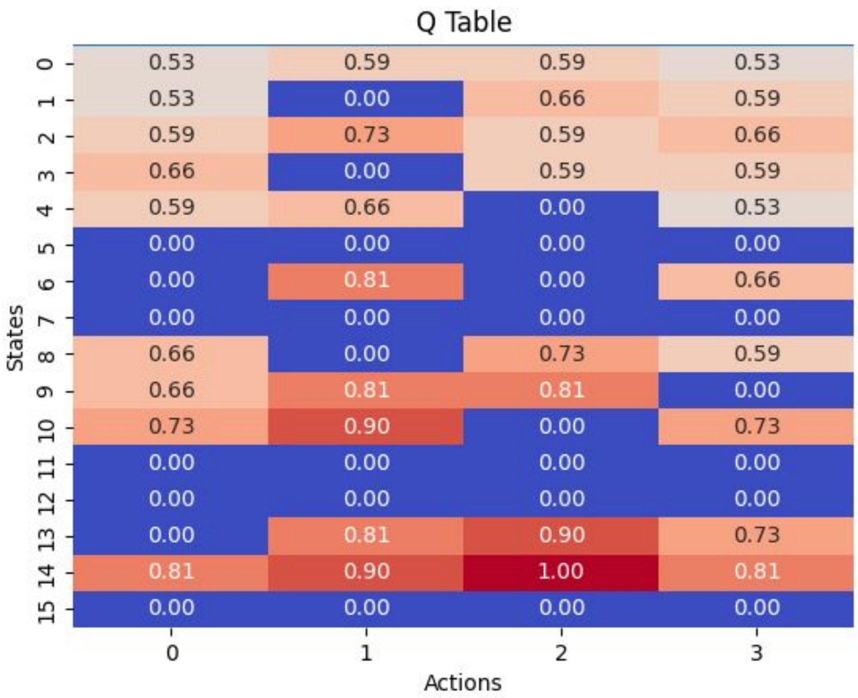
1) Rechard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction, second edition.* The MIT Press, Cambridge, Massachusetts, London, England, 2018, 2020.
2) https://karan-jakhar.medium.com/100-days-of-code-day-1-35afe174000e
3) https://gymnasium.farama.org/environments/toy_text/frozen_lake/#frozen-lake
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!

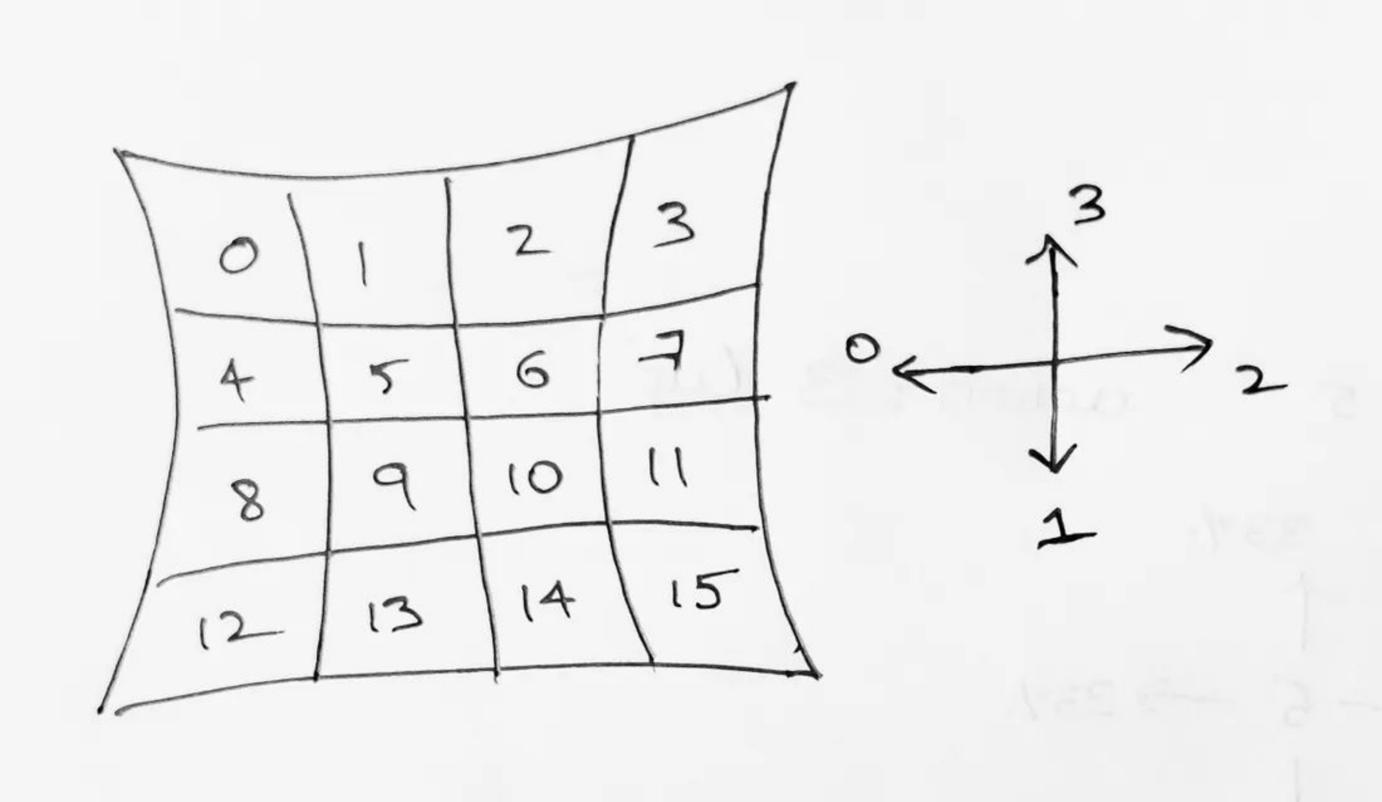
Policy Iteration



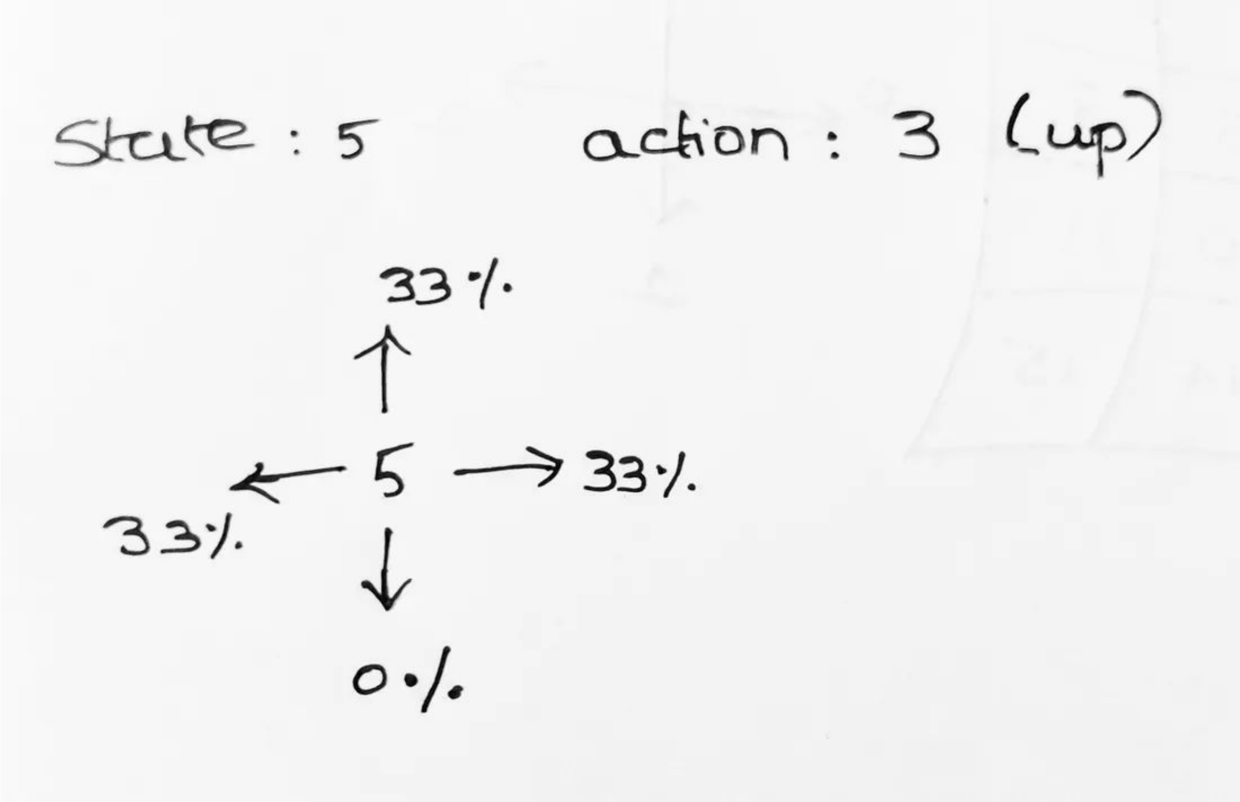
Q-Learning


Q Table
0.53 0.59 0.59 0.53
0.53 0.00 0.66 0.59
0.59 0.73 0.59 0.66
0.66 0.00 0.59 0.59
0.59 0.66 0.00 0.53
0.00 0.00 0.00 0.00
0.00 0.81 0.00 0.66
0.00 0.00 0.00 0.00
0.66 0.00 0.73 0.59
0.66 0.81 0.81 0.00
0.73 0.90 0.00 0.73
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.81 0.90 0.73
0.81 0.90 1.00 0.81
0.00 0.00 0.00 0.00
States
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Actions
0
1
2
3

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)]
```