Tejas Rane – 901015176
Sumeet Shanbhag - 641020714

Reinforcement Learning Homework 3

1. Policy Iteration:

Policy Iteration consists of two main steps: Policy Evaluation and Policy Improvement. The algorithm starts with a random policy and iteratively evaluates and improves it until the policy is stable.

- **Policy Evaluation:** It calculates the value function for each state under the current policy. The value function represents the expected return (sum of rewards) when starting in a given state and following the current policy.
- **Policy Improvement**: It updates the policy based on the calculated value function by choosing the action that maximizes the expected value in each state.
- **Termination:** The process repeats until the policy is stable, meaning the policy no longer changes.

2. Value Iteration:

Value Iteration directly calculates the optimal value function without maintaining an explicit policy. It iteratively updates the value of each state based on the Bellman optimality equation until convergence.

- **Value Update:** For each state, it calculates the expected return for each possible action and updates the value of the state to the maximum expected return.
- **Policy Extraction:** Once the value function has converged, the optimal policy is extracted by selecting the action that maximizes the expected return in each state.

3. Generalized Policy Iteration:

- **Generalized Policy Iteration (GPI)** is a framework that encompasses both Policy Iteration and Value Iteration. It involves the simultaneous, interleaved operation of policy evaluation and policy improvement.
- **Policy Evaluation and Improvement:** GPI performs policy evaluation and improvement in an interleaved manner, allowing the value function and policy to influence each other and gradually move towards optimality.

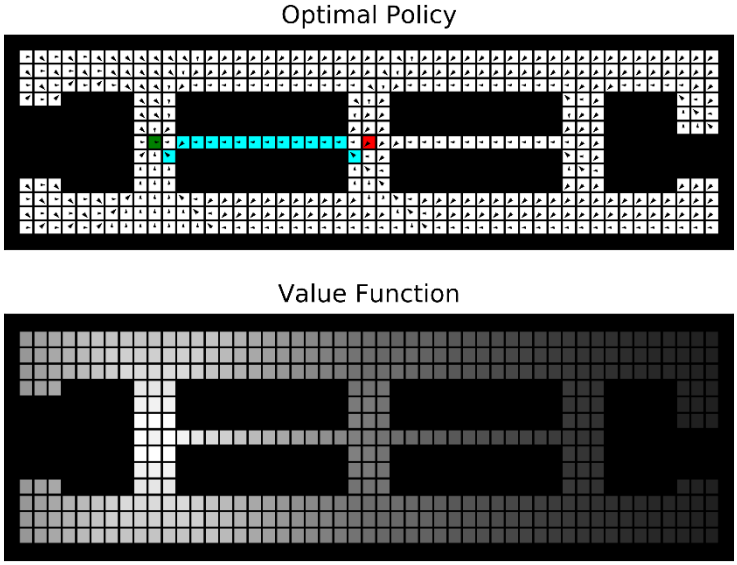
For all plots:

Top: Optimal Policy. Each block in the graph represents a state, and shows the action to be taken in that state. Red block is the start position, green block is the goal position, and the cyan blocks show the path taken by the agent.

Bottom: Value Function. Each block in the graph represents a state, and the color intensity represents the value of the state normalized over all states.

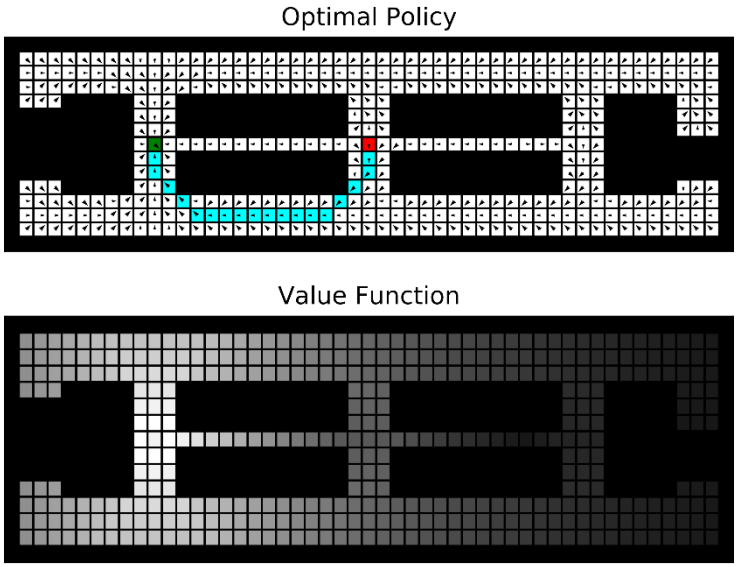
1 (a). Policy Iteration – Deterministic Agent:

Convergence time: 1.6311707496643066 sec



1 (b). Policy Iteration – Stochastic Agent:

Convergence Time: 2.918398380279541 sec



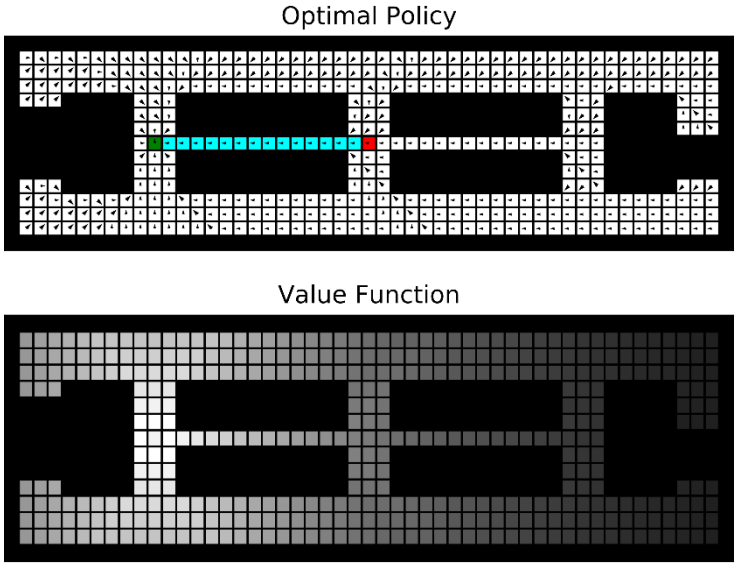
For all plots:

Top: Optimal Policy. Each block in the graph represents a state, and shows the action to be taken in that state. Red block is the start position, green block is the goal position, and the cyan blocks show the path taken by the agent.

Bottom: Value Function. Each block in the graph represents a state, and the color intensity represents the value of the state normalized over all states.

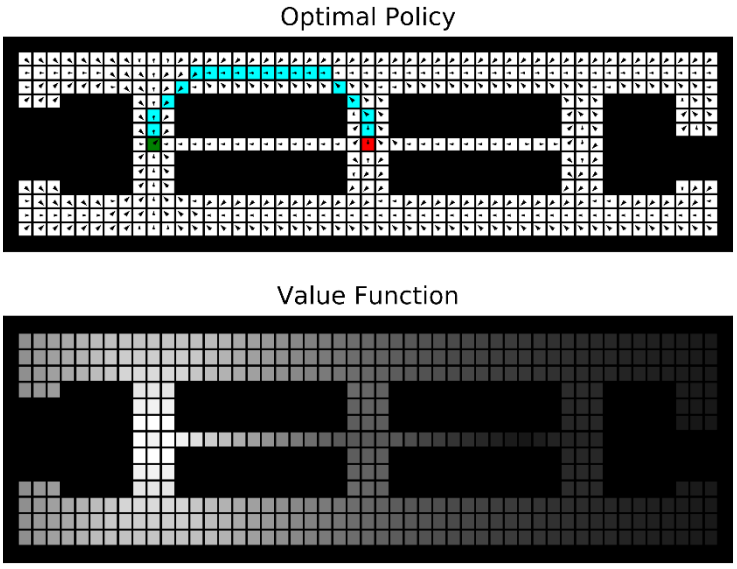
2 (a). Value Iteration – Deterministic Agent:

Convergence time: 1.9410829544067383 sec



2 (b). Value Iteration – Stochastic Agent:

Convergence time: 3.1693921089172363 sec



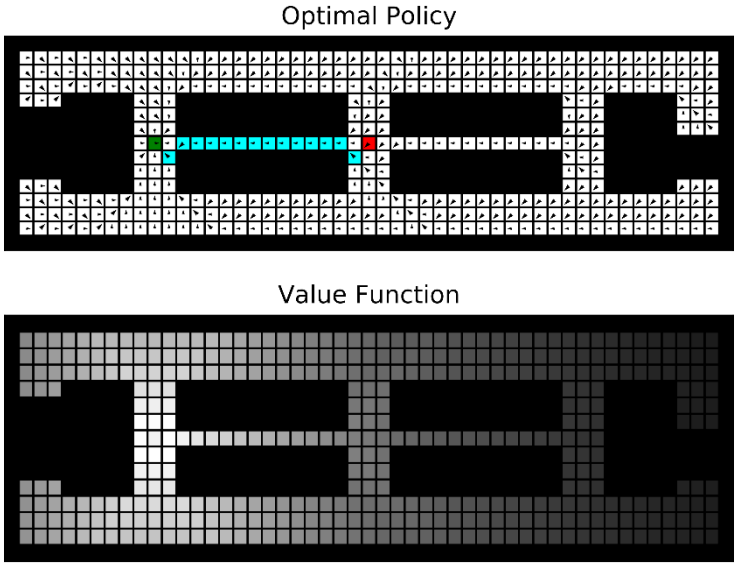
For all plots:

Top: Optimal Policy. Each block in the graph represents a state, and shows the action to be taken in that state. Red block is the start position, green block is the goal position, and the cyan blocks show the path taken by the agent.

Bottom: Value Function. Each block in the graph represents a state, and the color intensity represents the value of the state normalized over all states.

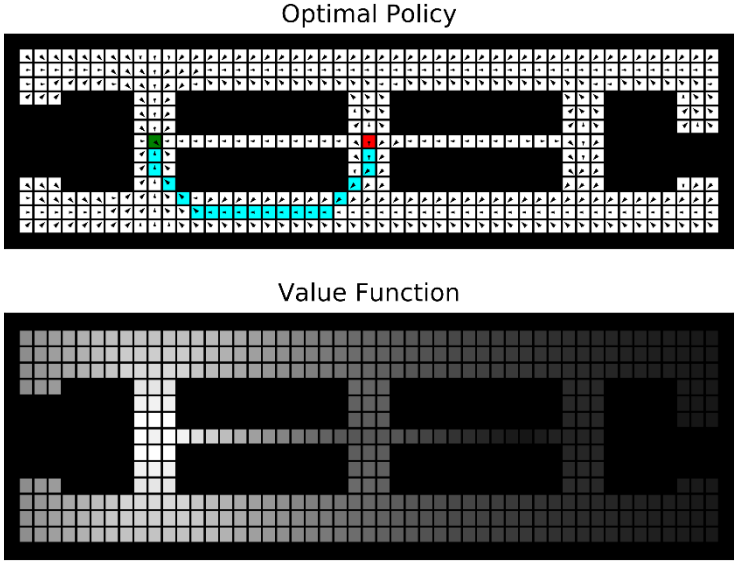
3 (a). Generalized Policy Iteration – Deterministic Agent:

Convergence time: 0.4213895797729492 sec



3 (b). Generalized Policy Iteration – Stochastic Agent:

Convergence time: 1.1156082153320312 sec



As seen from all of the plots above, the Generalised Policy Iteration has the fastest convergence time amongst all the algorithms with a convergence time of 0.42 seconds and 1.11 seconds for Deterministic and Stochastic Agents respectively.

Steps to run the code

For Deterministic Agent
python3 HW3_programming.py -t pi -d # Policy Iteration
python3 HW3_programming.py -t vi -d # Value Iteration
python3 HW3_programming.py -t gpi -d # Generalized Policy Iteration

For Non-Deterministic (Stochastic) Agent
python3 HW3_programming.py -t pi # Policy Iteration
python3 HW3_programming.py -t vi # Value Iteration
python3 HW3_programming.py -t gpi # Generalized Policy Iteration