Unit-1

Lab experiments

A robot navigates a warehouse to pick and place items.

Aim:

A robot navigates a warehouse to pick and place items. Define states (locations in the warehouse), actions (move in four directions), and rewards (picking an item: +2, reaching the goal: +5, hitting an obstacle: -2). Implement a policy evaluation algorithm to determine the value function for a given policy in Python.

Algorithm:

1. **Initialize** value function V(s)=0$V(s)=0$ for all states.
2. **Repeat** until convergence:
    - Set Δ=0Δ=0
    - For each state ss:
    - Store old value v=V(s)$v=V(s)$
    - Update

V(s)←∑aπ(a|s) [R+γV(s′)]$V(s)\leftarrow a\sum\pi(a|s)[R+\gamma V(s')]$

    - Update Δ=max(Δ,|v−V(s)|)Δ=max(Δ,|$v-V(s)$|)
3. **Stop** when Δ<θΔ<θ
4. **Output** the value function V(s)$V(s)$

Code Github Link:

https://github.com/syekumar/MLA0316-Reinforcement-learning-

Output:

```
Value Function after Policy Evaluation:

[[ 1.65  2.51  4.5   6.16]
 [ 1.52  0.    4.6   6.11]
 [ 2.07  3.21  5.23  8.06]
 [ 2.39  3.77  7.38 10.86]]
```

Result:

The policy evaluation algorithm computes the value function for all states in the warehouse. Higher values indicate states closer to the goal and item, while lower values occur near obstacles. The goal state has the highest value, showing maximum expected reward under the given policy.