

**Project 1: Deadline 2/12/2025**

The environment is modeled by a grid of 5x5 cell size as shown in the figure below, but you are welcome to try a larger grid size (e.g., 10 x 10). The cells of the grid correspond to the states of the environment. Assume that the robot has four actions (up, down, right, left) to select at each time/iteration. You would need to define the reward for the robot to learn to find an optimal way to get to the goal. Optimal here means less number of actions taken by the robot

Suggested reward (you are encouraged to define your own reward):










- Action that makes the robot tend to go out of the grid will get a reward of -1 (when the robot is in the border cells)
- Action that makes the robot reach the goal will get a reward of 100
- All other actions will get a reward of 0

<b>Starting</b> 				
				<b>Goal</b>

Using Q learning (Off-Policy Control) to train the robot for this task. This Q-learning technique is in Chapter 6 of Sutton's book.

**Requirement: Write a report to cover the following requirements**

1. (20 points) Plot the action selection of the initial learning episode and the last learning episode.  
Something is similar to this table:

Starting 				
				
				
				
				Goal 

2. (20 points) Show the Q table of the last learning episode
3. (20 points) Plot the reward of all learning episodes
4. (20 points) Plot number of steps/actions taken from the starting location to the goal.
5. (20 points) Submit your homework with source code and instructions to run your code (readme file) to Canvas
6. (Bonus 10 points) Implement Sarsa( $\lambda$ ) in Slide 15, Lecture 4, or Implement Q ( $\lambda$ ) in Slide 17, Lecture 4.