# TUGAS 3: Q - LEARNING


Name = Puruso Muhammad Hanunggul

Class = IF-39-INT

NIM = 1301153680


## Problem Description (Case Study)

In this case, we use Reinforcement Learning method, Q Learning. The problem is we have to build a Q-Learning program to find optimum policy that *Agent* can reach the goal state (10, 10) from start state (1, 1) based on map given:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | -1 | -3 | -5 | -1 | -3 | -3 | -5 | -5 | -1 | 100 |
| 9 | -2 | -1 | -1 | -4 | -2 | -5 | -3 | -5 | -5 | -5 |
| 8 | -3 | -4 | -4 | -1 | -3 | -5 | -5 | -4 | -3 | -5 |
| 7 | -3 | -5 | -2 | -5 | -1 | -4 | -5 | -1 | -3 | -4 |
| 6 | -4 | -3 | -3 | -2 | -1 | -1 | -1 | -4 | -3 | -4 |
| 5 | -4 | -2 | -5 | -2 | -4 | -5 | -1 | -2 | -2 | -4 |
| 4 | -4 | -3 | -2 | -3 | -1 | -3 | -4 | -3 | -1 | -3 |
| 3 | -4 | -2 | -5 | -4 | -1 | -4 | -5 | -5 | -2 | -4 |
| 2 | -2 | -1 | -1 | -4 | -1 | -3 | -5 | -1 | -4 | -1 |
| 1 | -5 | -3 | -1 | -2 | -4 | -3 | -5 | -2 | -2 | -2 |

While the *Agent* is learning, it start from random initial state and stop when he reach the goal

## Method and Design

This Problem is solved by using Q-Learning, one of Reinforcement Method in Machine Learning. In this case, I map the state and action as a number for the row and column. For example row 0 in Q matrix represent state in row 10 column 1 grid world. Row 1 in Q matrix represent state in row 10 column 2, etc. For action, is represent 0, 1, 2, and 3 for each Up, Right, Down, and Left

There are few steps of algorithm to solve this problem. First, we have to determine the Episode (Iteration) that has been set 10.000, gamma, the value between 0 until 1 that has been set 0.9, and The R matrix which is the reward matrix that we designed based on the grid world above.

Next step is we initialize the matrix Q as zero matrix. This matrix will become a *Brain* of the *Agent*. The Q matrix will be updated while the agent learn.

For each episode, program select the random initial state. While the current state is not goal state, select one random state among all possible action from the current state, whether go to 'Up', 'Right', 'Down', and 'Left'. Then, program select the direction (index of the value represent the direction) of the maximum value from this new state that has been selected randomly. Then compute the Q value to update the Q matrix by using this formula:

$$Q(state, action) = R(state, action) + Gamma * Max[Q(next\ state, all\ actions)]$$

Then, the matrix will be updated, and the new state will set as current state. The program will did this iteration until 10.000 episode (iteration) as I set before.

## Output

For the implementation, the initial state is set in (1, 1) then the program will find the best path from Q matrix that program get. The program will record the direction that it head for and the value of each path.

The output will be the record of direction and summation of every path that program passed.

## Screenshot

```
Agent is Learning. Please wait....
```

When the program is learning.

After this, the program will generate Q matrix, the result of learning

```
The path that crossed by agent [80, 81, 82, 83, 84, 74, 64, 54, 44, 45, 46, 47, 37, 38, 28, 18, 8, 9]
The value of cost is with sum 3610
Direction ['Up', 'Right', 'Right', 'Right', 'Right', 'Up', 'Up', 'Up', 'Up', 'Right', 'Right', 'Right', 'Up', 'Right', 'Up', 'Up', 'Up', 'Right']
```

One of example of the program output.