I have make use Of Q learning to solve this path planning problem .

**Q-Learning** is a basic form of Reinforcement Learning which uses Q-values (also called action values) to iteratively improve the behavior of the learning agent.

1.**Q-Values or Action-Values:** Q-values are defined for states and actions. Q(S,A) is an estimation of how good is it to take the action A at the state S . This estimation of which will be iteratively computed using the **TD- Update rule .**

2. **Rewards and Episodes:** An agent over the course of its lifetime starts from a start state, makes a number of transitions from its current state to a next state based on its choice of action and also the environment the agent is interacting in. At every step of transition, the agent from a state takes an action, observes a reward from the environment, and then transits to another state. If at any point of time the agent ends up in one of the terminating states that means there are no further transition possible. This is said to be the completion of an episode.

3. **Temporal Difference or TD-Update:**

The Temporal Difference or TD-Update rule can be represented as follows :



This update rule to estimate the value of Q is applied at every time step of the agents interaction with the environment. The terms used are explained below. :

* S: Current State of the agent.
* A: Current Action Picked according to some policy.
* S’: Next State where the agent ends up.
* A’: Next best action to be picked using current Q-value estimation, i.e. pick the action with the maximum Q-value in the next state.
* R: Current Reward observed from the environment in Response of current action.
* Y(>0 and <=1) : Discounting Factor for Future Rewards. Future rewards are less valuable than current rewards so they must be discounted. Since Q-value is an estimation of expected rewards from a state, discounting rule applies here as well.
* **Learning Rate:** lr or learning rate, often referred to as *alpha* orα can simply be defined as how much you accept the new value vs the old value.

4. **Choosing the Action to take using Epsilon-greedy policy:**

Epsilon-greedy policy of is a very simple policy of choosing actions using the current Q-value estimations. It goes as follows :

* With probability 1-epsilon choose the action which has the highest Q-value.
* With probability epsilon choose any action at random.

As per the instruction I have taken the discount factor as 1 and run through 2000 training episodes to train the agent and I have taken epsilon as 0.9 and learning rate as 0.9 .

**Future Improvement :**

1. It is not giving desired output , so to rectify the mistake in future .