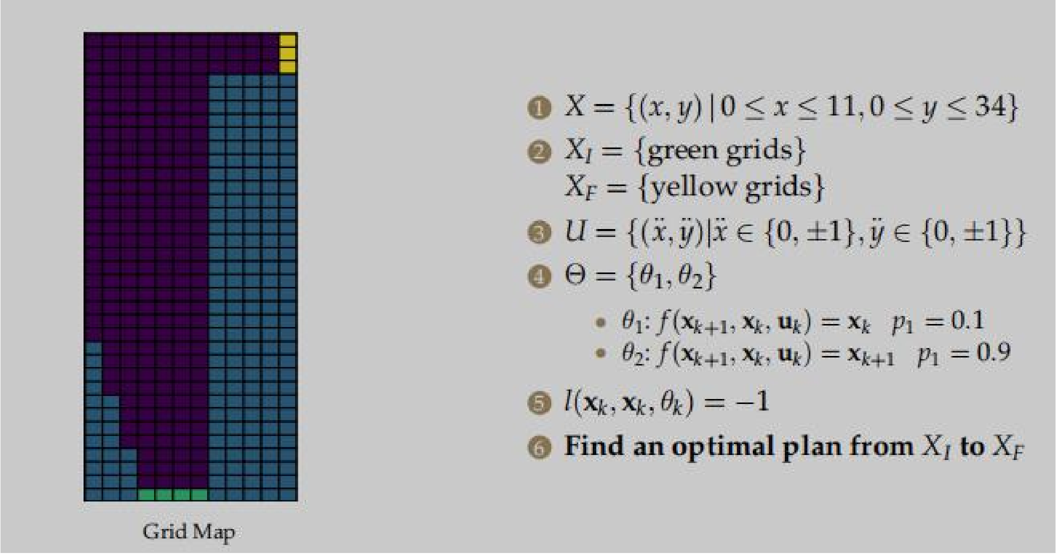
Intro

In this project, I try to implement a MDP-Based planning method (Real Time Dynamic Programming) to solve a car racing problem in Python.

RTDP algorithum

1. Initialize G values of all states to admissible values;
2. Follow greedy policy picking outcomes at random until goal is reached;
3. Backup all states visited on the way;
4. Reset to xs and repeat 2-4 until all states on the current greedy policy have Bellman errors < ∆, where ∆(xk) = kG(xk)) G(xk+1)k.

Problem Description

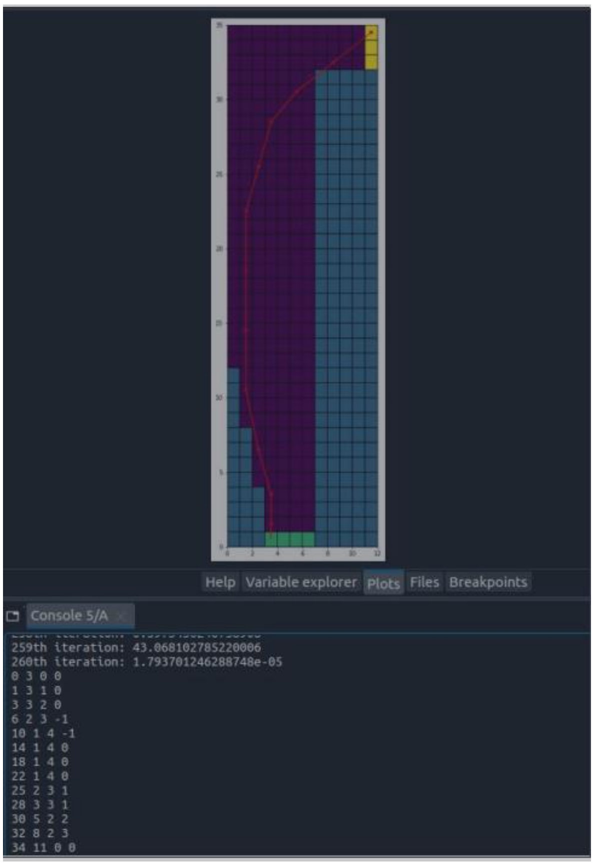


Grid Map

1. $X = \{(x, y) \,|\, 0 \leq x \leq 11, 0 \leq y \leq 34\}$
2. $X_I = \{\text{green grids}\}$
   $X_F = \{\text{yellow grids}\}$
3. $U = \{(\ddot{x}, \ddot{y}) | \ddot{x} \in \{0, \pm 1\}, \ddot{y} \in \{0, \pm 1\}\}$
4. $\Theta = \{\theta_1, \theta_2\}$
   - θ_1: $f(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k \quad p_1 = 0.1$
   - θ_2: $f(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_{k+1} \quad p_1 = 0.9$
5. $l(\mathbf{x}_k, \mathbf{x}_k, \theta_k) = -1$
6. **Find an optimal plan from X_I to X_F**

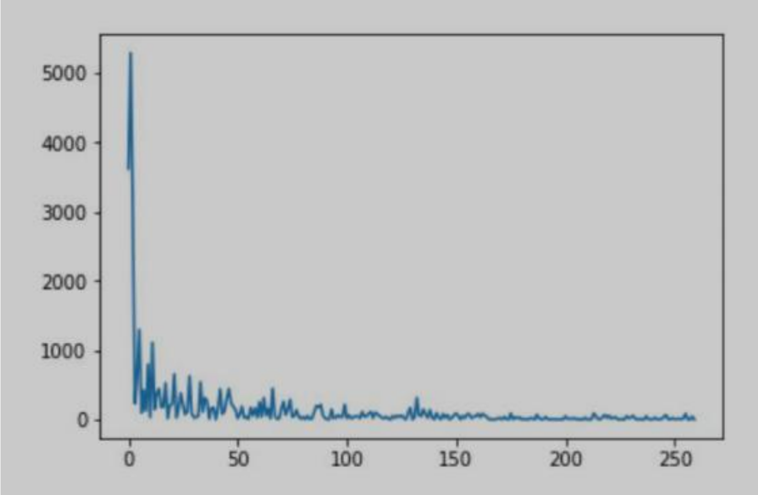
Result(Euler distance as heuristic)

- path:



```
259th iteration: 43.068102785220006
260th iteration: 1.793701246288748e-05
0 3 0 0
1 3 1 0
3 3 2 0
6 2 3 -1
10 1 4 -1
14 1 4 0
18 1 4 0
22 1 4 0
25 2 3 1
28 3 3 1
30 5 2 2
32 8 2 3
34 11 0 0
```

- process of convergence(x-axis: i th of iteration; y-axis: Bellman error after each iteration)


5000
4000
3000
2000
1000
0
0
50
100
150
200
250