# RTDP algorithm

The basic workflow I understand:

1. Initialization of the G value for each state: not only admissible but also close to the optimal G value.
2. Greedily selecting the state from start to end with the optimal G value. Based on the G value, the optimal trajectory (policy) is obtained. One variant is to make some random choices.
3. Update and backup the G value of all visited states (trajectory)
4. Repetition step 2 to step 3 until the sum of Bellman errors of all states between two iterations are less than a threshold.
5. Obtaining the optima trajectory from the G value

# Tricks in the Code

A lot of tricks and efforts have been taken. Therefore, it cannot work for a generalized case.

## Admissible Heuristic

**First step:**

1. Relaxation of constraints: ignoring occupied and outbound situations;
2. Assuming a constant max speed.
3. Considering diagonal movements

An admissible heuristic was proposed but it is better than Euclidean distance:

```
math.ceil((dx + dy - 2*min(dx,dy))/4)  + math.ceil(min(dx,dy)/4)
```

**Second step:**

High penalties were given to those states which will inevitably lead to “collisions”, such as:

```
if abs(vx) == 4 :
    if abs(px - bound_x_0) <= 6 or  abs(px - bound_x_1) <= 6:
        graph[key].g_value = 99
```

## Breaking symmetry

To increase the symmetry, I slightly increased the G-value of those states that are further from the goal and it is still admissible because of the existence of occupied areas.

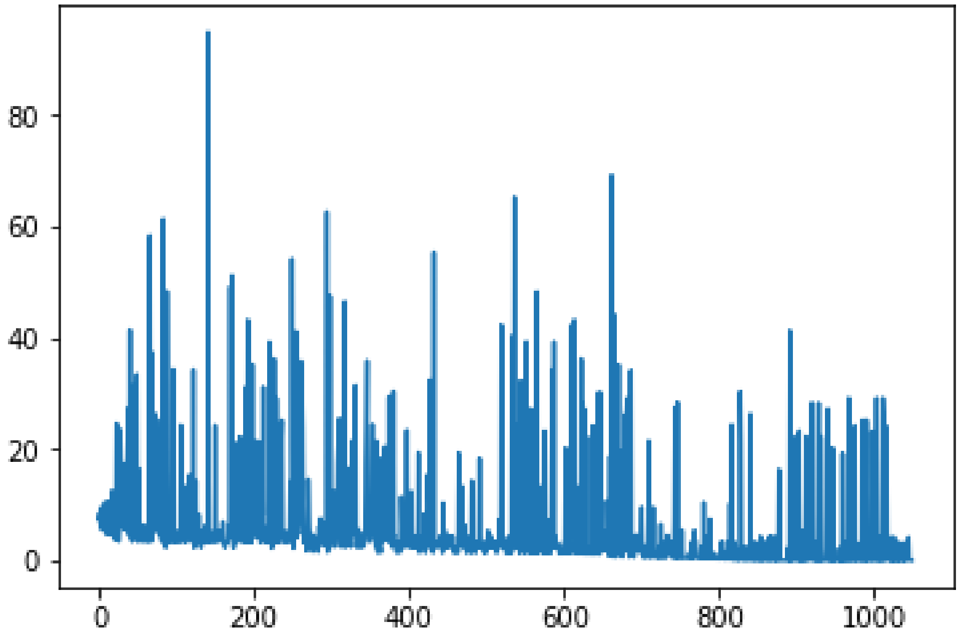
```
Gmap[px,py] = min(g) + 0.01 * abs(px -32)
```

## Forbidden looping during step greedy policy

The explored state (already backup in the trajectory) could not be re-visited in order to avoid an infinite loop.

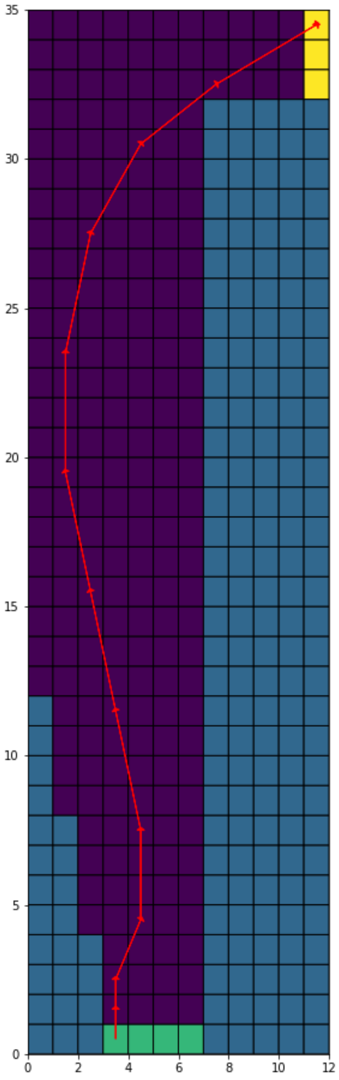
# Simulation Results

After 1049th iteration, Bellman error is 9e-05, less than 0.0001.



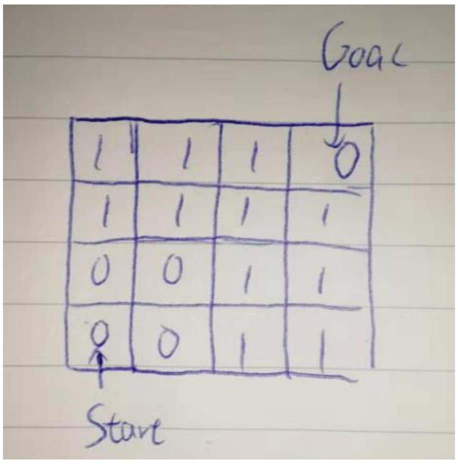
The course of Bellman Error

The optimal trajectory takes 12 steps to reach the final line (see the picture below).



## Open Questions

1. During greedy policy, what is a good way to avoid an infinite loop (repetitions)?
2. I tried to get outcome at random to get a trajectory, but this could only make the convergence more difficult. Why?
3. Admissible heuristic seems to be not sufficient for greedy policy to find the goal, as the following example shows. Heuristic should also be consistent to guarantee that the greedy policy can find a goal instead of an infinite loop [1].



| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |

## Reference

[1] https://en.wikipedia.org/wiki/Consistent_heuristic