强化学习课程《Maze》实验报告

姓名：黄骏齐 学号：2100012956

实验要求

在Maze环境上实现并对比Dyna-Q算法和Dyna-Q+算法的表现，撰写实验报告并提交代码。
使用Sutton RL book P166 Example 8.2 Blocking Maze 和 P167 Example 8.3 Shortcut Maze两个例子来对比上述两个算法。

Maze环境

`maze.hpp`，添加了两个成员函数 `maze_blocking` 和 `maze_shortcut` 来表示迷宫的两种变化

按照书上的例子初始化迷宫并且分别进行blocking和shortcut

实验过程

见 `maze_main.cpp`
分别实现了 `MazePolicyDynaQ` 和 `MazePolicyDynaQplus` 两个类分别表示DynaQ和DynaQ+算法，且设置规划步数n=5。对DynaQ+算法，设置了$k = 0.002$
对于两种情况

1. 在初始状态的1000步后blocking并且总共训练3000步
2. 在初始状态的3000步后shortcut并且总共训练6000步

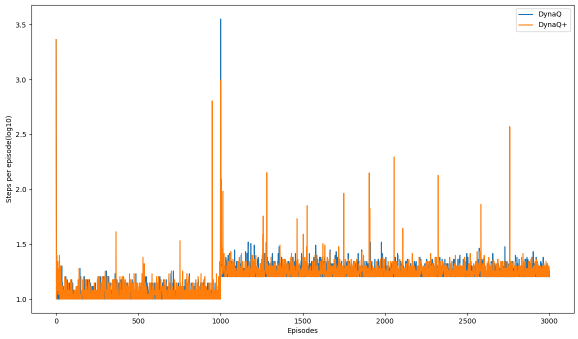
实验结果

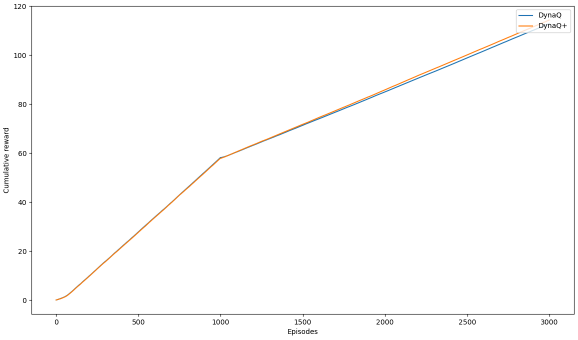
分别见 `DynaQBlocking.out`，`DynaQplusBlocking.out`，`DynaQShortcut.out`，`DynaQplusShortcut.out`

结果可视化

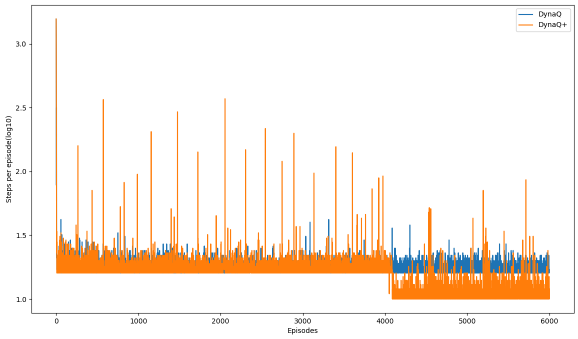
为了显示更加清晰，下图对步数进行了对数($\log_{10}$)处理

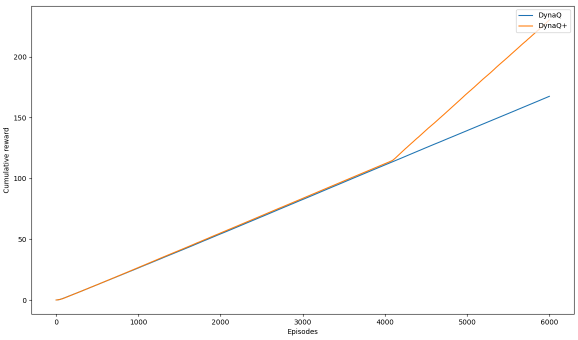
Blocking





Shortcut





结果分析

- 可以看到在Blocking前后， `DynaQ` 和 `DynaQ+` 算法都有一段“寻找道路”的时期，但很快又可以重新找到最短路
- 对于Shortcut后， `DynaQ` 并不能重新找到最短路线，但是由于能够更好地去探索新道路， `DynaQ+` 在shortcut发生的大约1000步后能够找到新的最短路线。

实验代码与结果均可见于https://github.com/huangjunqi1/Reinforcement-Learning-Work/tree/main/%E4%BD%9C%E4%B8%9A1.6%20%E8%BF%B7%E5%AE%AB