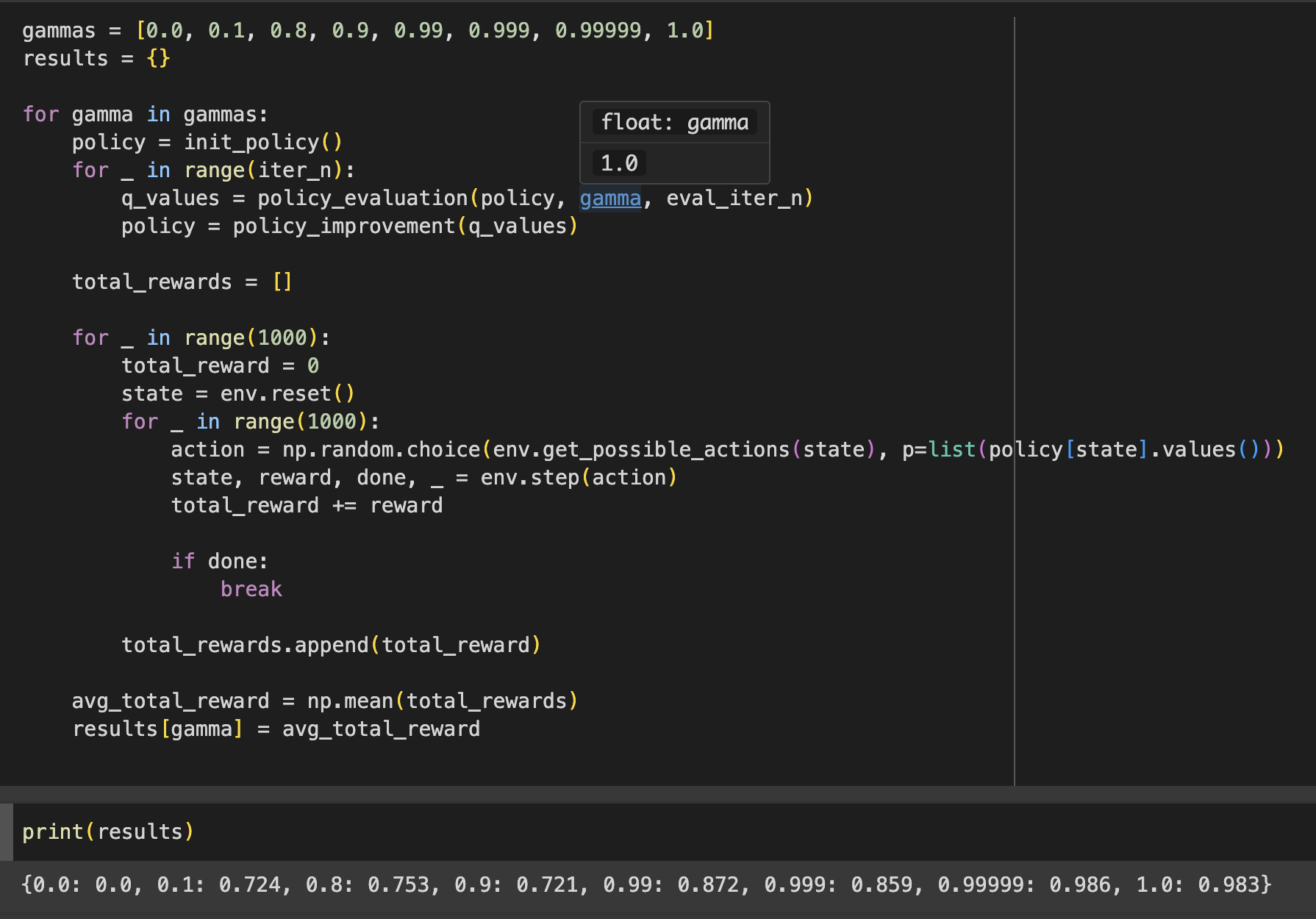
Домашняя работа #3 RL

# Первое задание

## Обучение

### Эксперимент 1

Необходимо было выбрать гамму с наибольшим значением награды.

Чем больше гамма, тем больше значение средней награды. Интересно то, что при значение гамма 1.0 все работает. Видимо из-за того, что среда конечна.

## Вывод

У гаммы 0.99999 наибольшее значение награды.

# Второе задание

## Обучение

### Эксперимент 1

def policy\_evaluation(policy, v\_values, gamma, eval\_iter\_n):

for \_ in range(eval\_iter\_n):

v\_values = policy\_evaluation\_step(v\_values, policy, gamma)

q\_values = get\_q\_values(v\_values, gamma)

return q\_values

Добавил bootstrapping значений. Уменьшил количество итераций, чтобы проверить есть ли преимущество в скорости сходимости, но ответ приблизительно такой же. Это странно. Возможно стоило бы попробовать с другой средой.

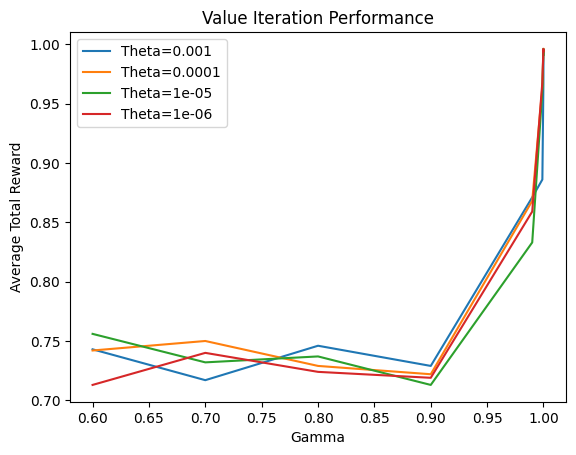
## Вывод

На этой среде не совсем понятно насколько bootstrapping увеличивает точность.

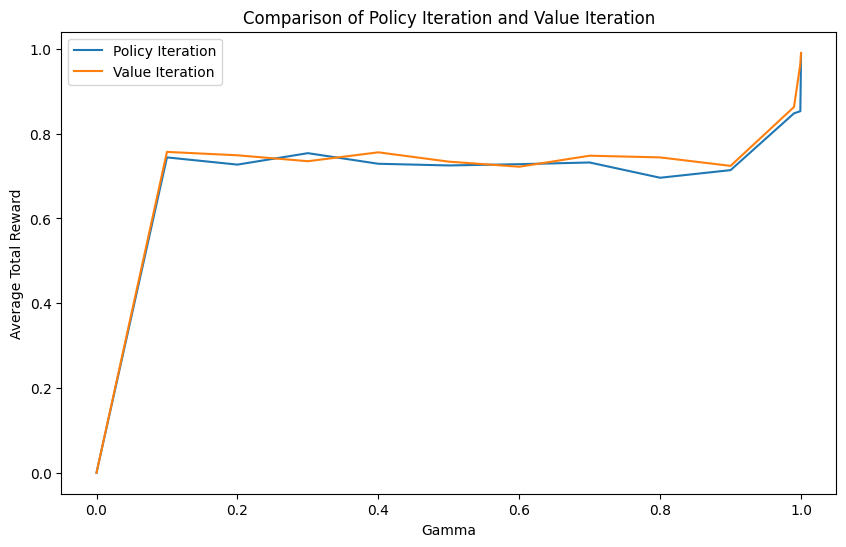
# Третье задание

## Обучение

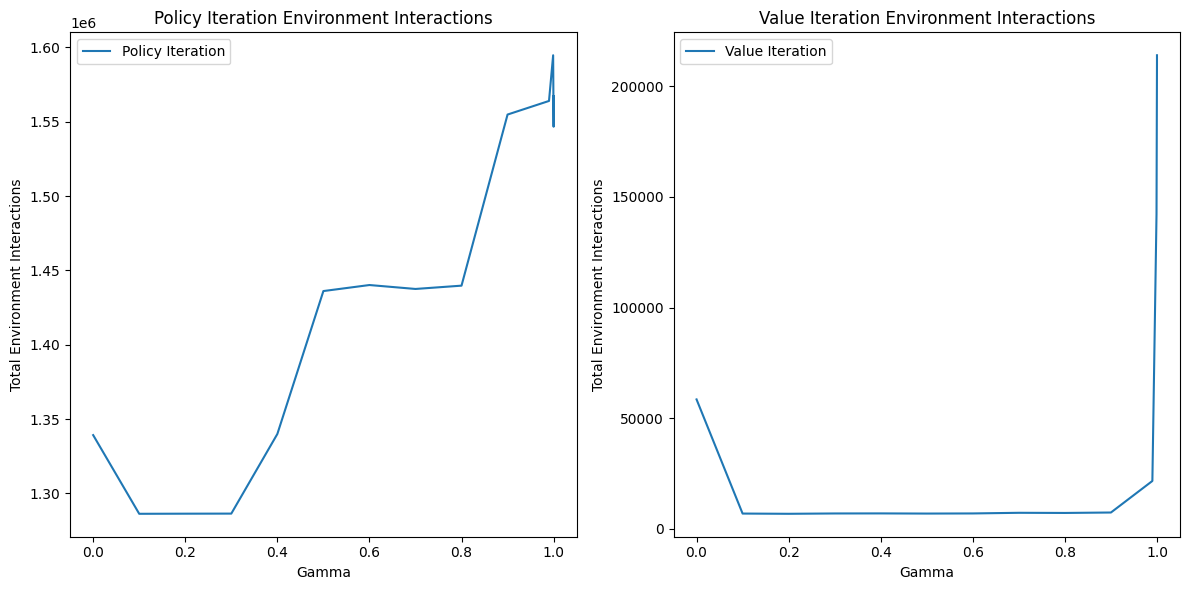
### Эксперимент 1



Написал value iteration. Получились следующие результаты для гиперпараметров. Не совсем было ясно какие еще гиперпараметры поэтому были исследованы тета и гамма. По сравнению с policy iteration total reward немного выше.



Попробуем теперь сравнить по вызовам среды.



Это графики сходимости при разных гамма. У value iteration theta - 1e-5.

## Вывод

Value iteration сходится за значительно меньшее количество вызовов.