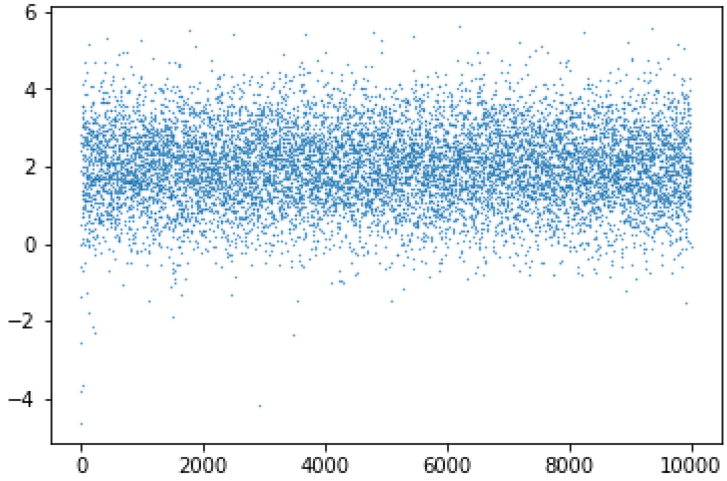
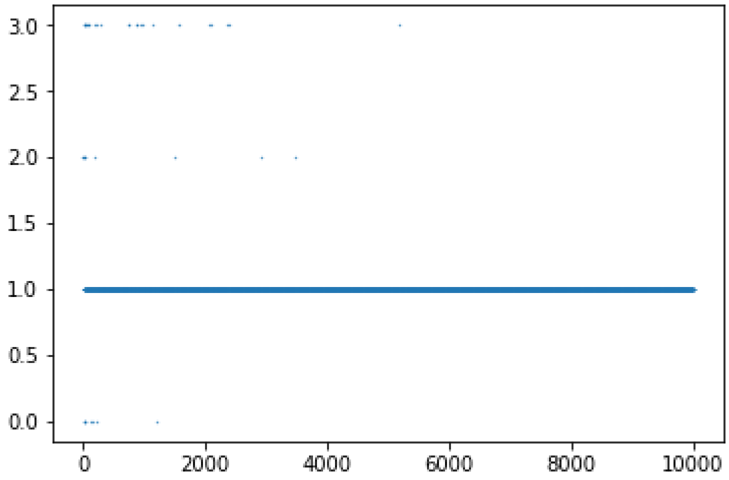
```python
import numpy as np
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
from torch.distributions import Categorical


np.random.seed(652)

K = 4
action_values = np.array([0, 2, -2, 1])



# experiment
T = int(1e4)
alpha = 1e-1
G = []

policy = nn.Sequential(
    nn.Linear(1, K, bias = False),
    nn.Softmax(dim = -1)
)


optim = torch.optim.SGD(params = policy.parameters(), lr = alpha)
actions = []
for t in range(T):
    # draw the action
    dist = Categorical(probs = policy(torch.ones([1])))
    action = dist.sample()

    reward = np.random.normal(action_values[action.item()], 1)
    actions.append(action.item())

    G.append(reward)

    # update the policy
    loss = -dist.log_prob(action) * reward
    optim.zero_grad()
    loss.backward()
    optim.step()

plt.scatter(x = np.arange(T), y = G, s = 0.1)
plt.figure()

plt.scatter(x = np.arange(T), y = actions, s = 0.1)
plt.show()

```


6
4
2
0
−2
−4
0
2000
4000
6000
8000
10000


3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
2000
4000
6000
8000
10000