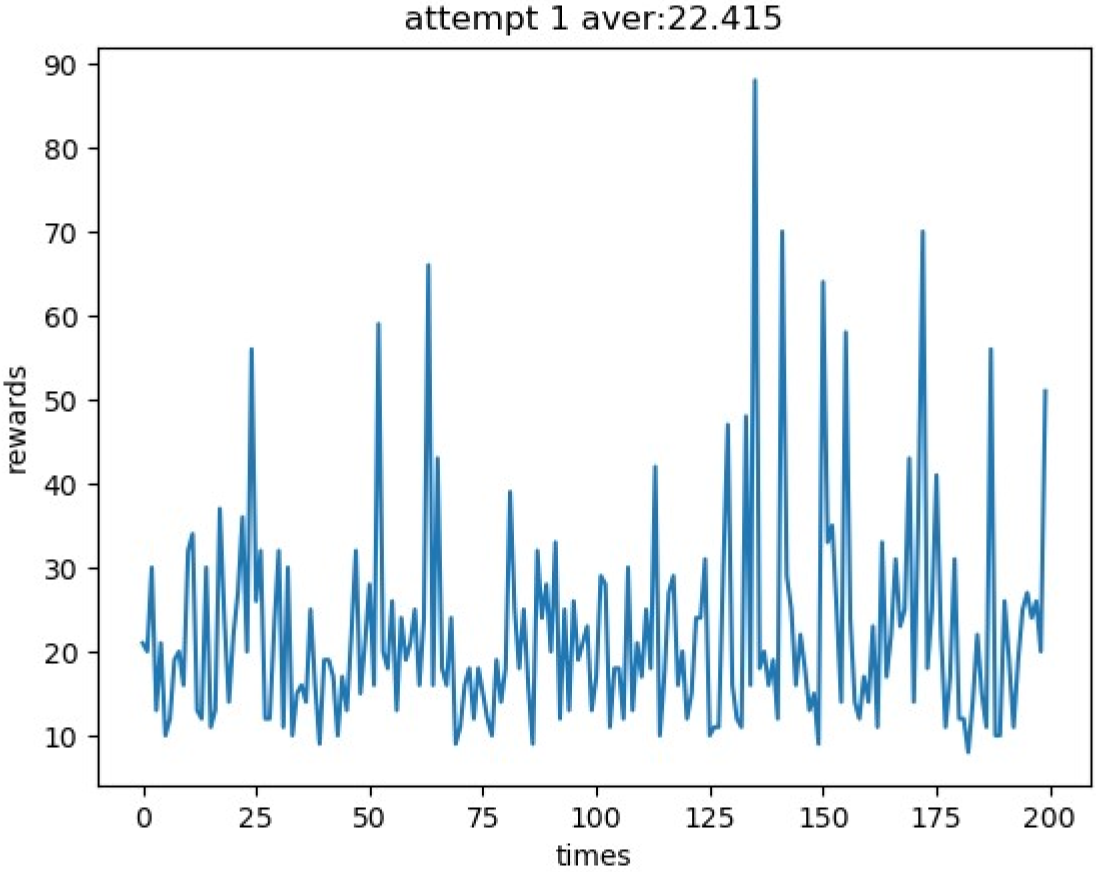
ai reinforcement homework(?

21/5/14 ニオ(neodoggy)

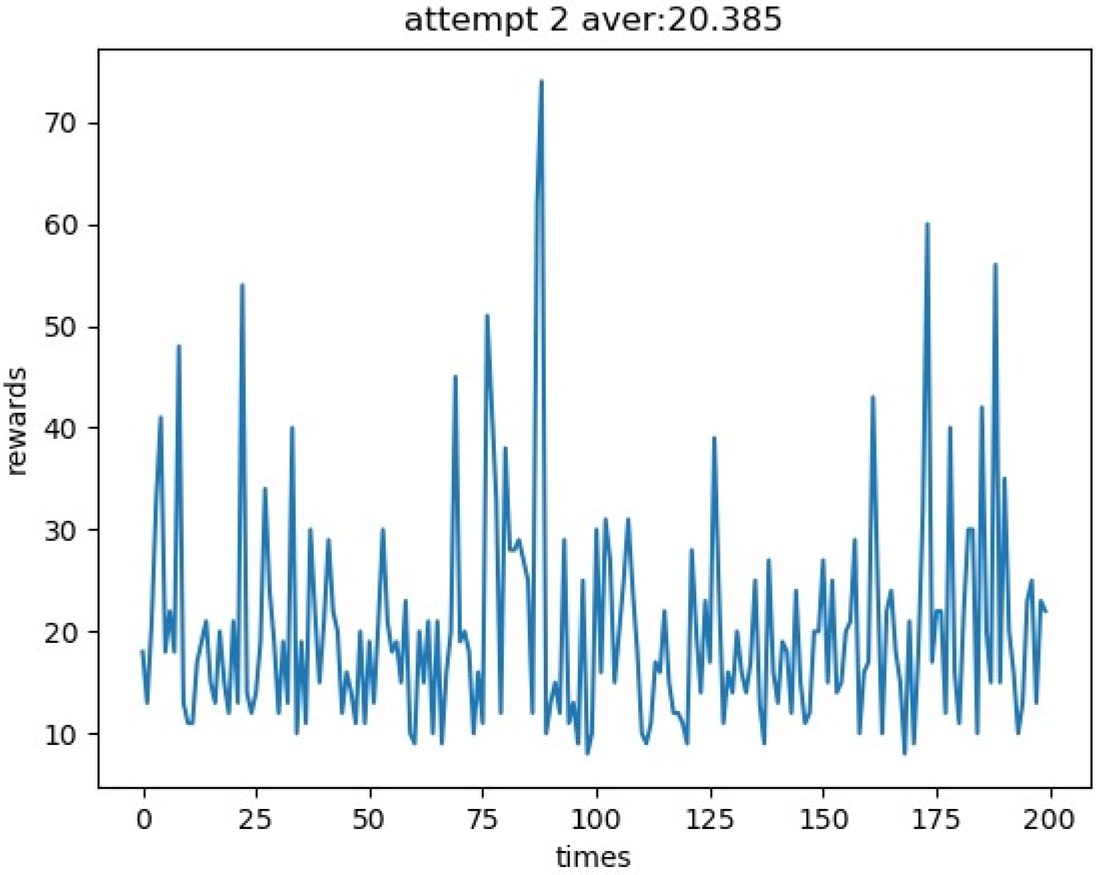
artificial idiot code：

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np
import pydot
import gym
env = gym.make('CartPole-v0')
observation = env.reset()
Kx=[]
Ky=[]
for i in range(200):
    env.reset()
    rewards=0
    for t in range(100):
        #env.render()
        action = env.action_space.sample()
        observation, reward, done, info = env.step(action)
        rewards += reward
        print(observation)
        if done:
            print("Rewards: ", rewards)
            Kx.append(rewards)
            Ky.append(i)
            rewards=0
            break
    env.close()
plt.clf()
plt.plot(Kx,Ky)
plt.title("yeah")
plt.xlabel("rewards")
plt.ylabel("times")
plt.show()
```

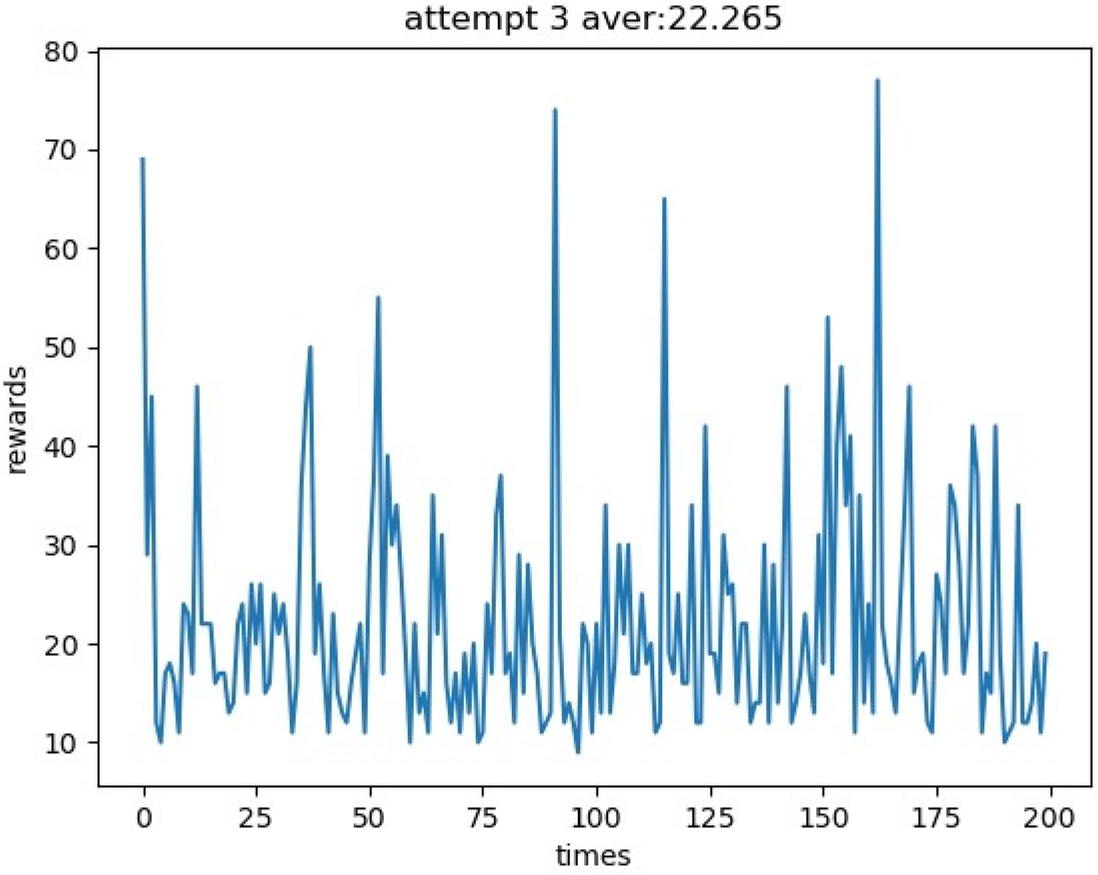
artificial idiot pic：

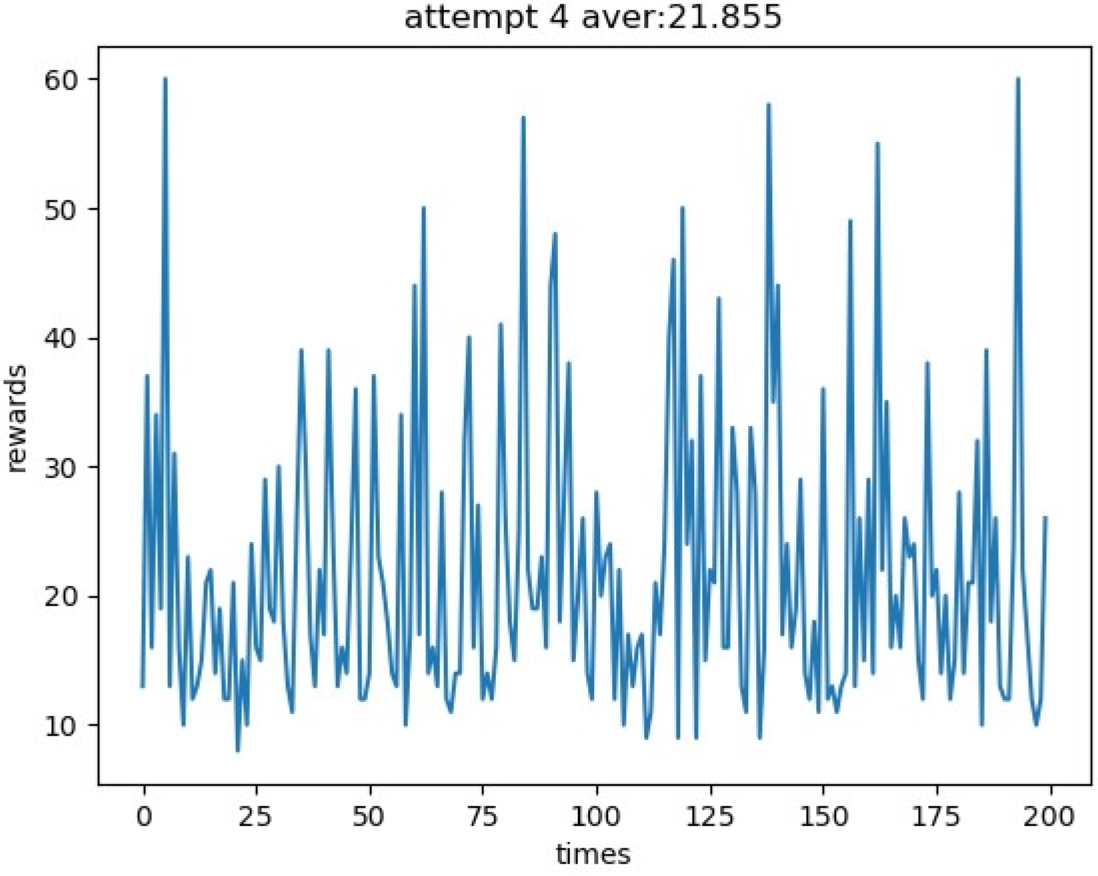


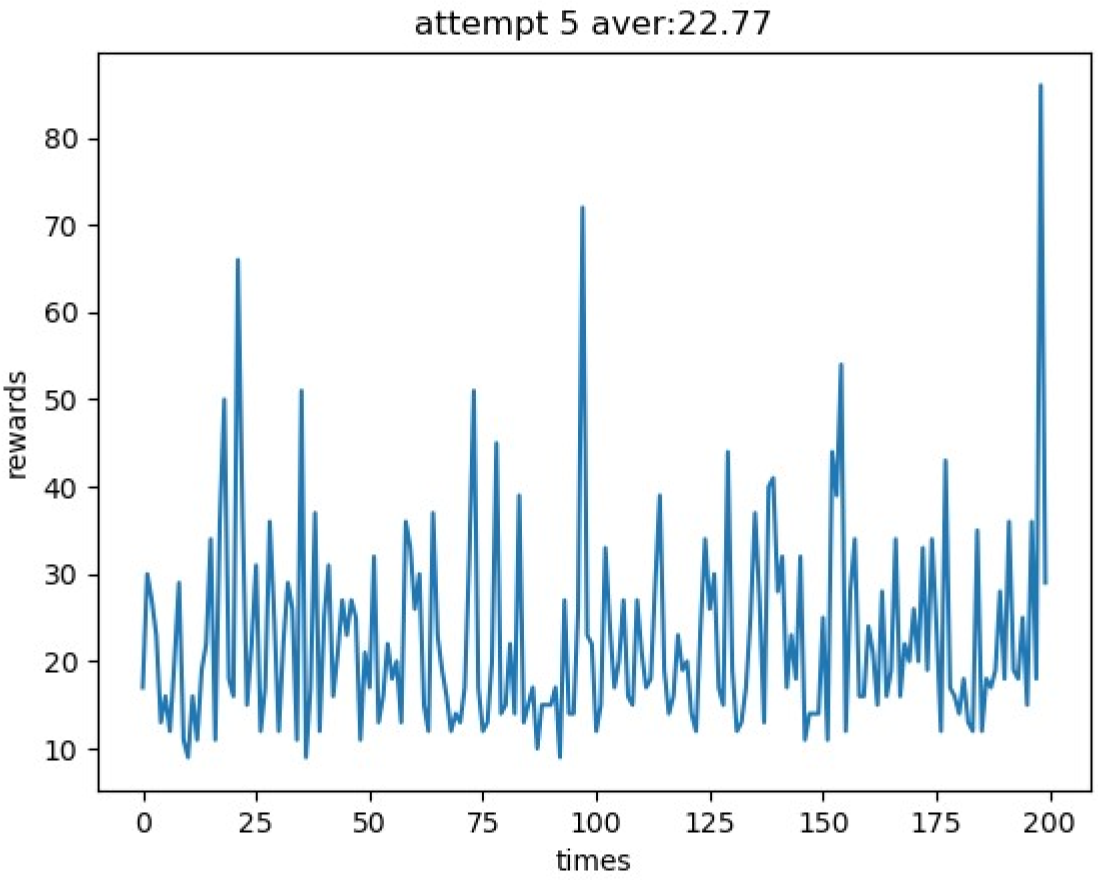
attempt 1 aver:22.415



attempt 2 aver:20.385


attempt 3 aver:22.265

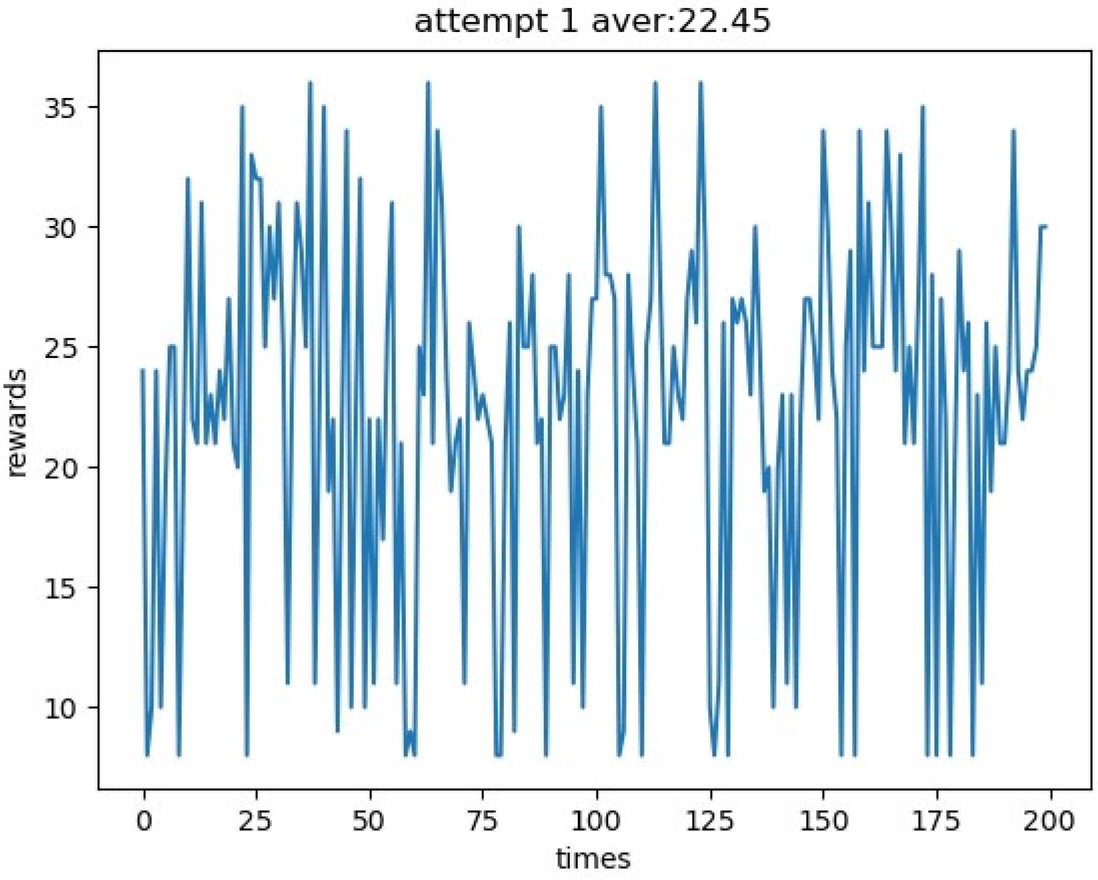

attempt 4 aver:21.855

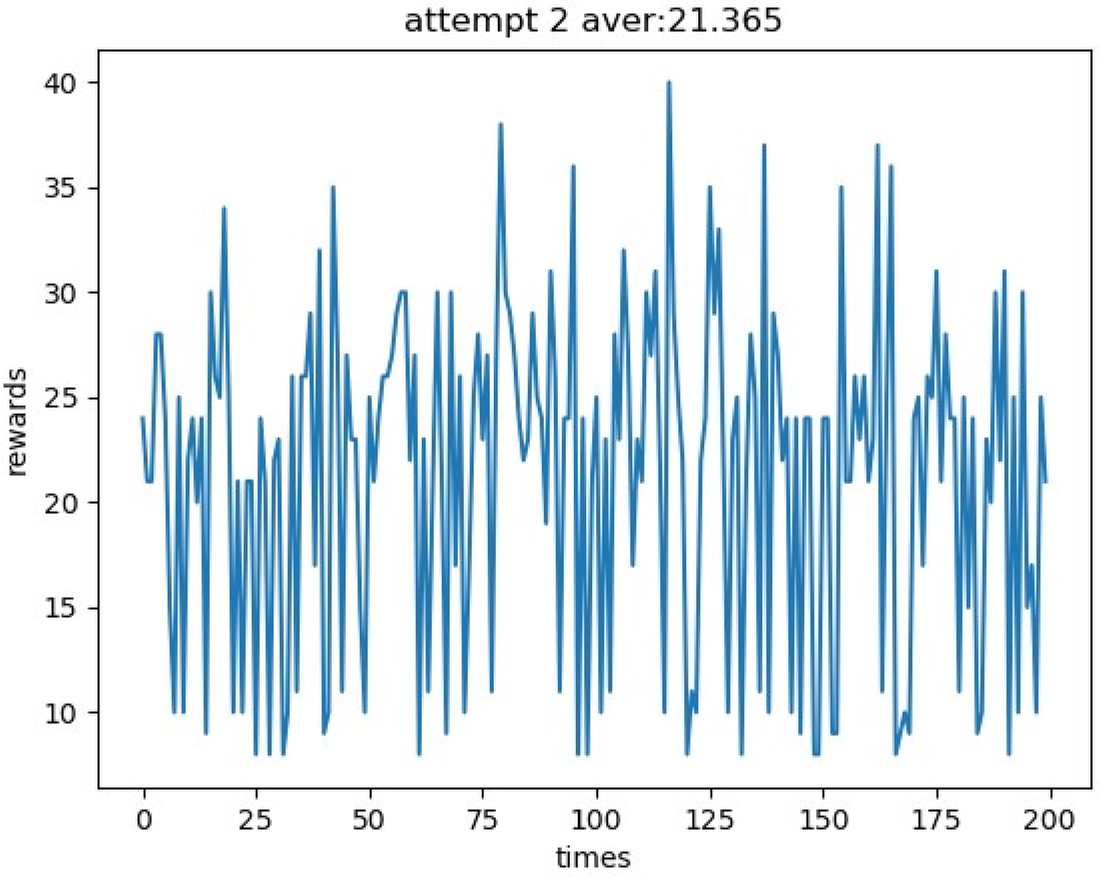

attempt 5 aver:22.77
rewards
80
70
60
50
40
30
20
10
0
25
50
75
100
125
150
175
200
times

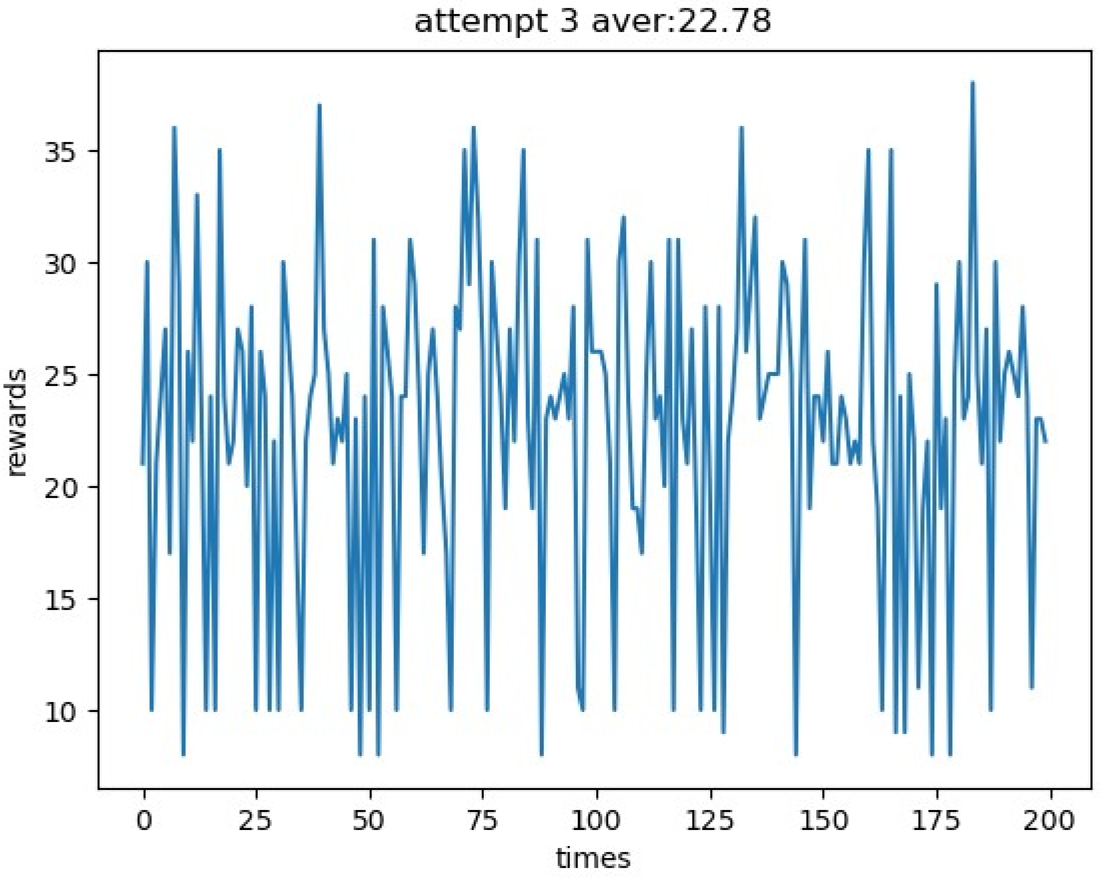
only theta

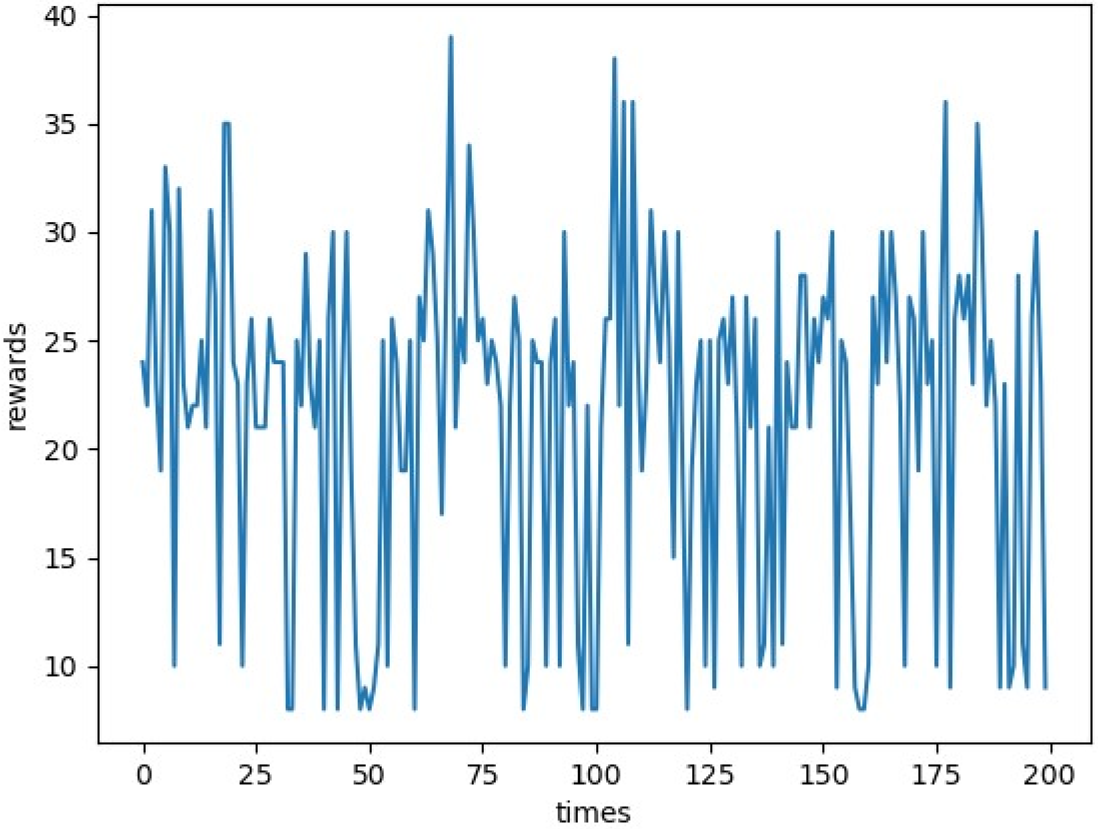
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np
import pydot
import gym

def solve(obs):
    th,w=obs[2:4]
    if abs(th) < 0.03:
        return 0 if w<0 else 1
    else:
        return 0 if th<0 else 1

def omega(obs):
    w=obs[3]
    return 0 if w<0 else 1

def theta(obs):
    th=obs[2]
    return 0 if abs(th)<0.03 else 1

for a in range(1,2):
    env = gym.make('CartPole-v0')
    Kx=[]
    Ky=[]
    av=0
    for i in range(200):
        env.reset()
        rewards=0
        action = env.action_space.sample()
        for t in range(300):
            #env.render()
            observation, reward, done, info = env.step(action)
            rewards += reward
            print(observation)
            if done:
                print("Rewards: ", rewards)
                Ky.append(rewards)
                av+=rewards
                Kx.append(i)
                rewards=0
                break
            action=theta(observation)
        env.close()
    plt.clf()
    plt.plot(Kx,Ky)
    plt.title("attempt "+str(a)+" aver:"+str(av/200))
```

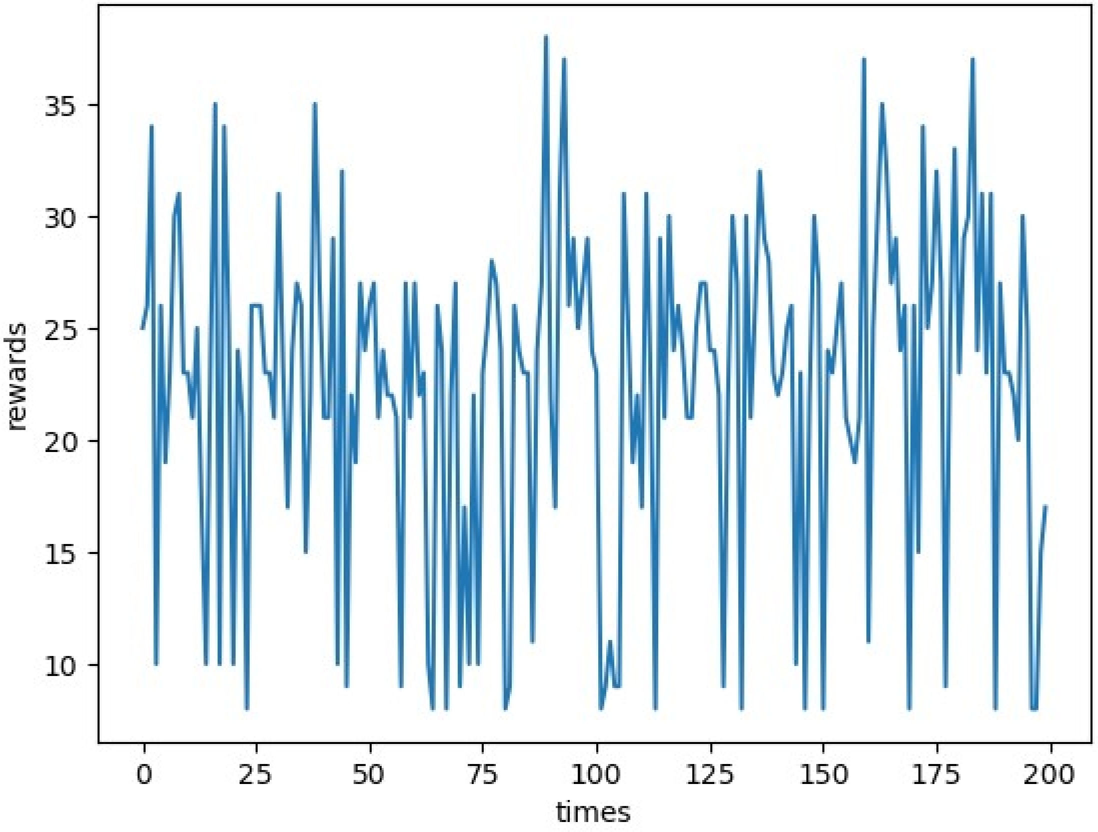
pics:


attempt 1 aver:22.45
rewards
times
35
30
25
20
15
10
0
25
50
75
100
125
150
175
200


attempt 2 aver:21.365


attempt 3 aver:22.78


attempt 4 aver:21.62

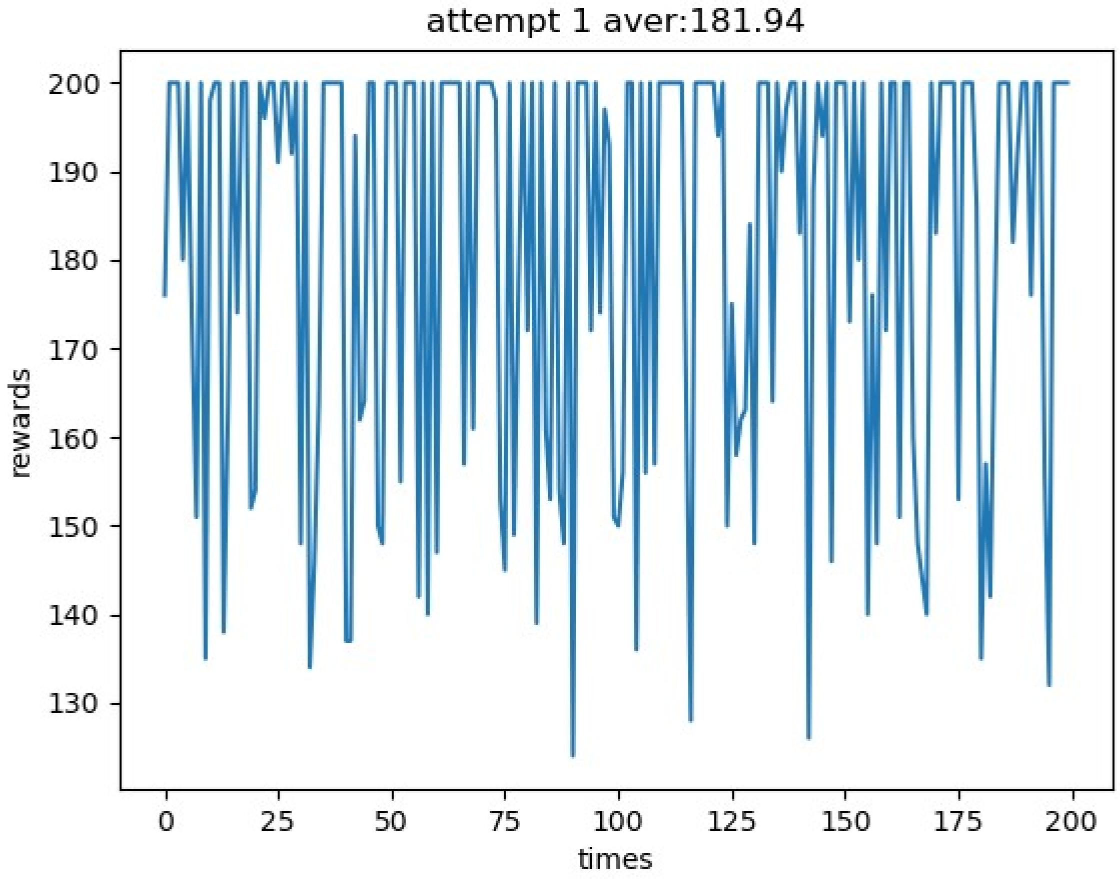

attempt 5 aver:22.505

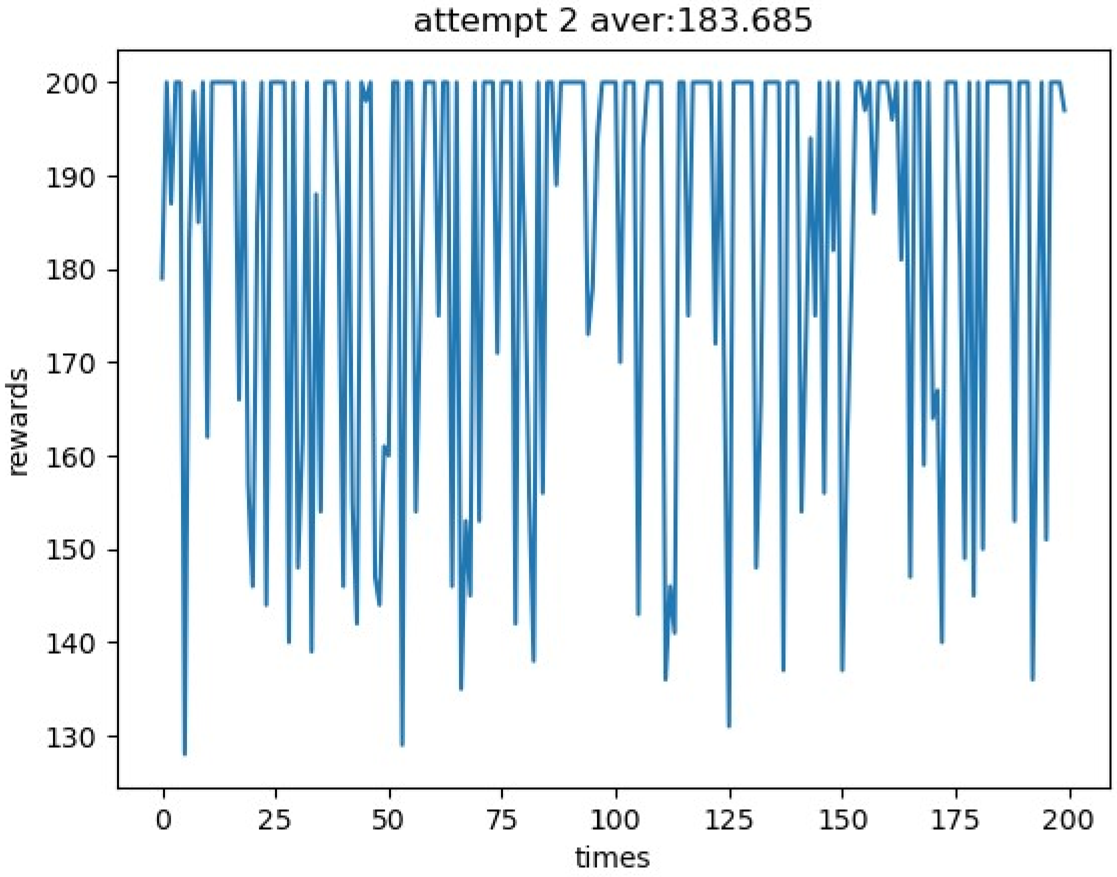
only omega

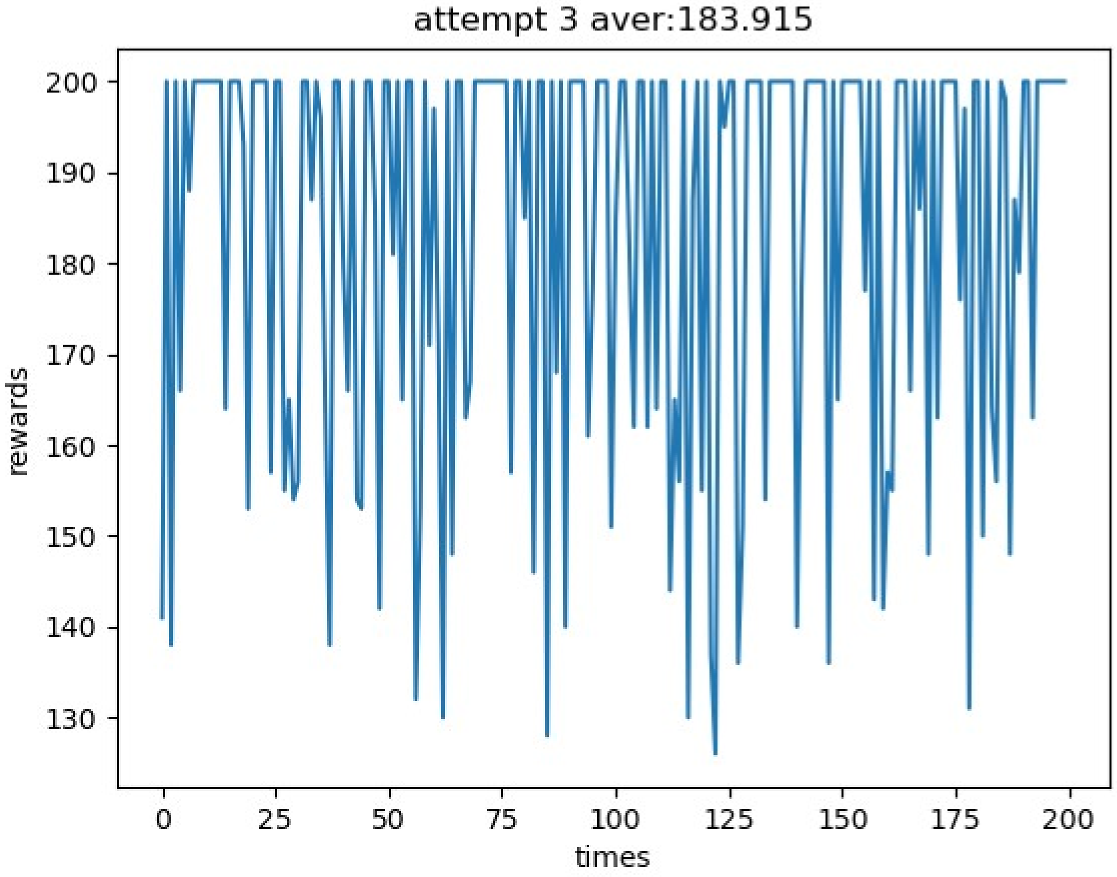
change line 44 to

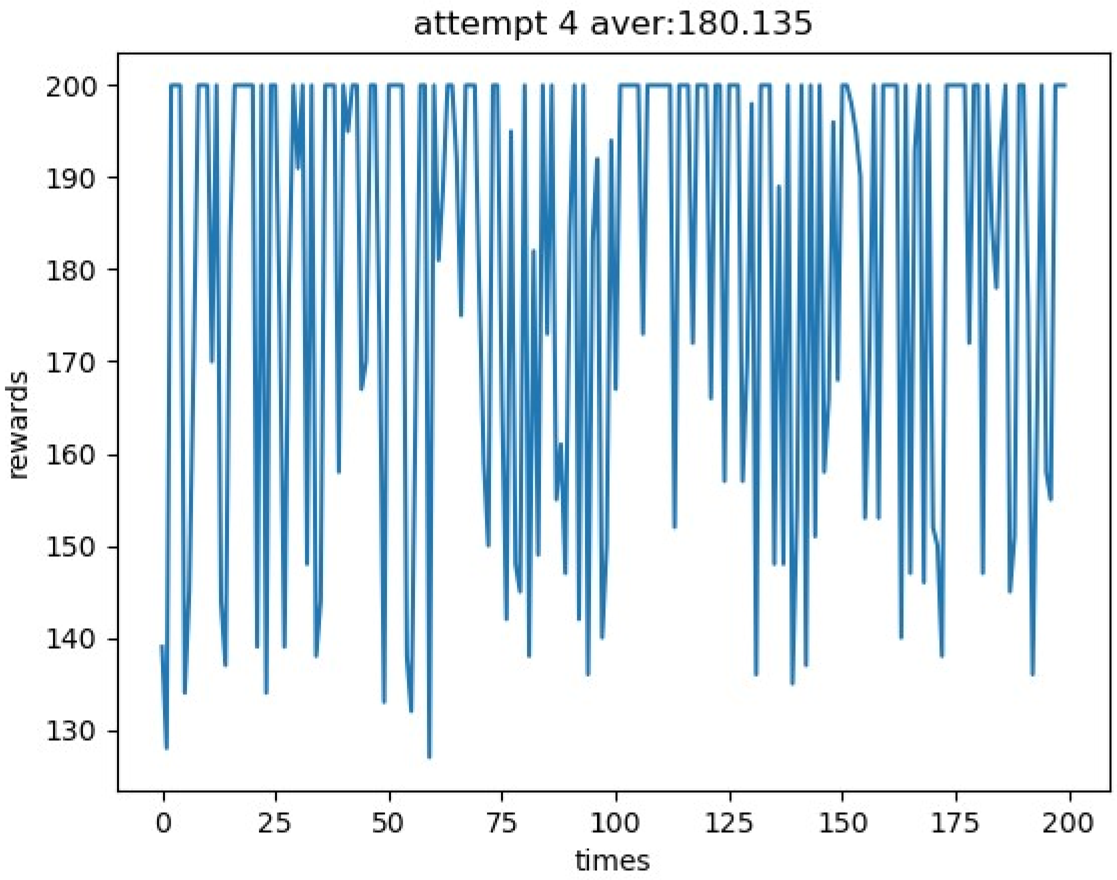
```
44  action=omega(observation)
```

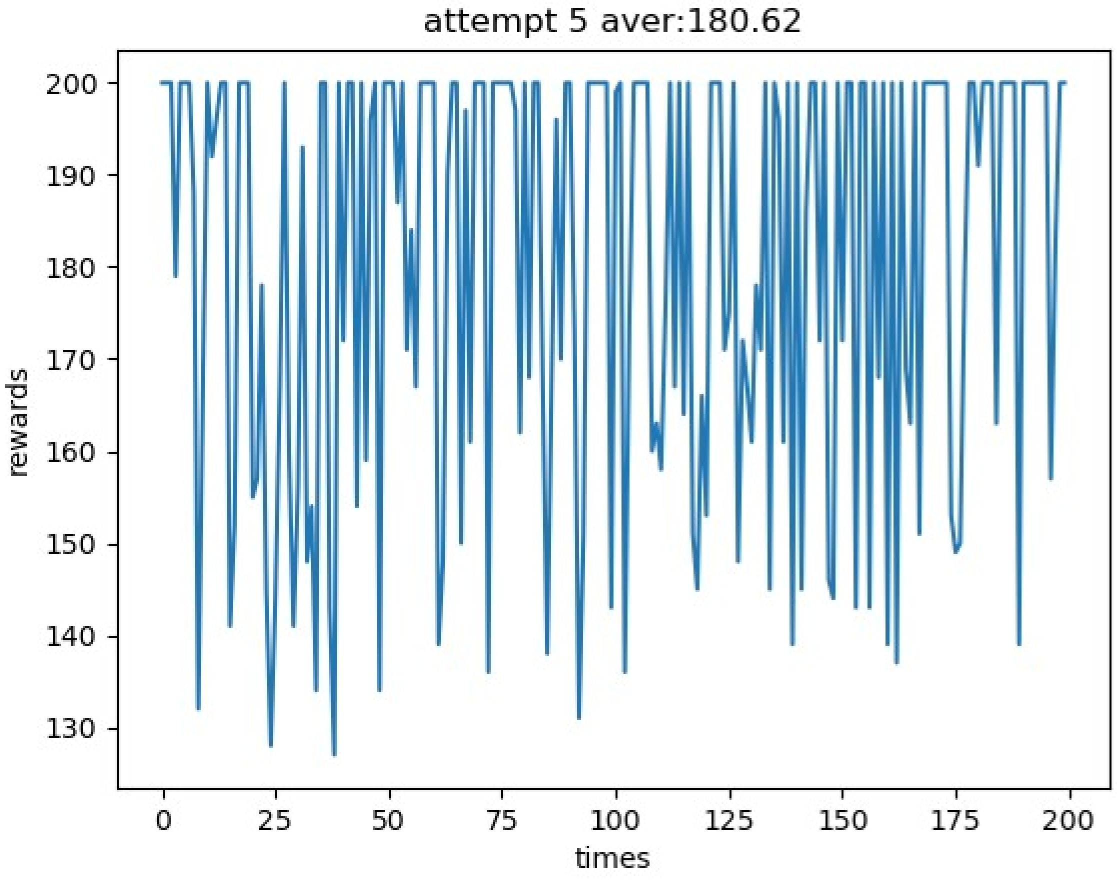
pic:




attempt 2 aver:183.685


attempt 3 aver:183.915


attempt 4 aver:180.135
rewards
200
190
180
170
160
150
140
130
times
0
25
50
75
100
125
150
175
200

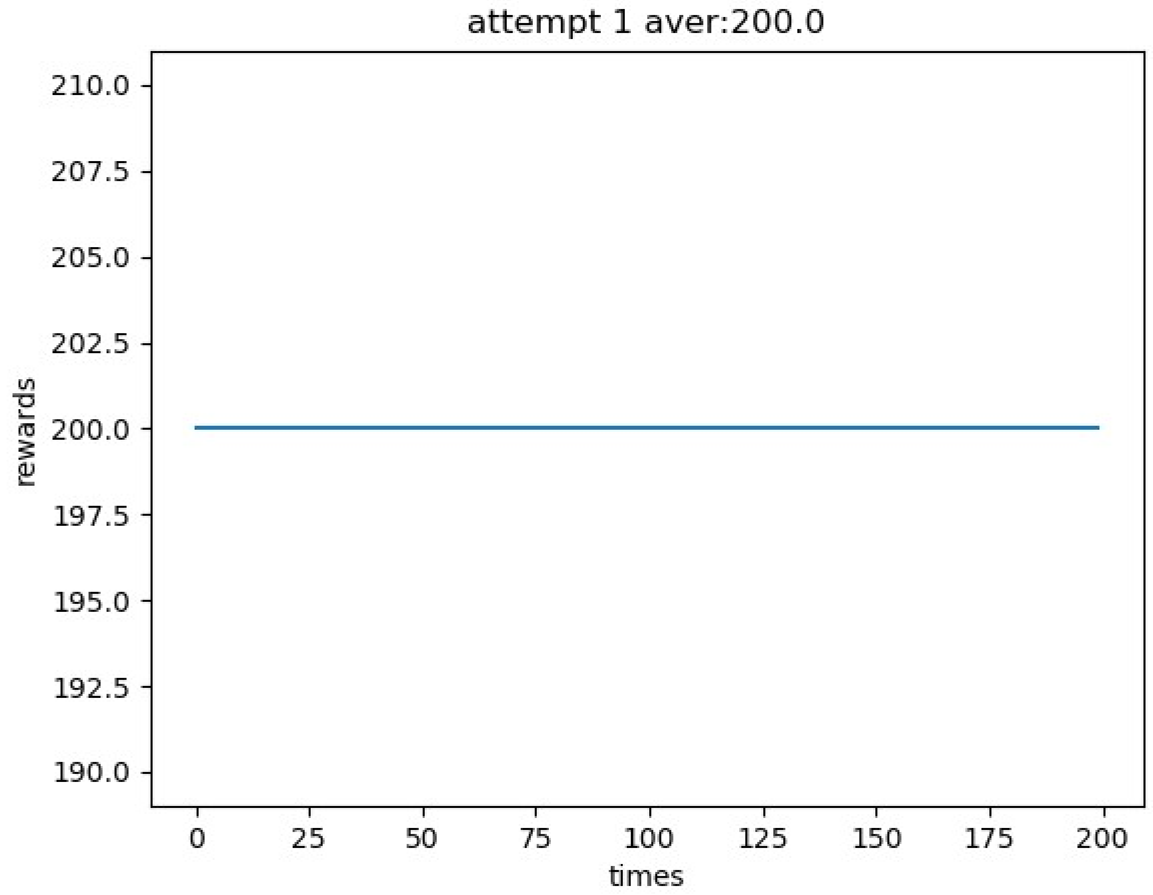

attempt 5 aver:180.62
rewards
200
190
180
170
160
150
140
130
times
0
25
50
75
100
125
150
175
200

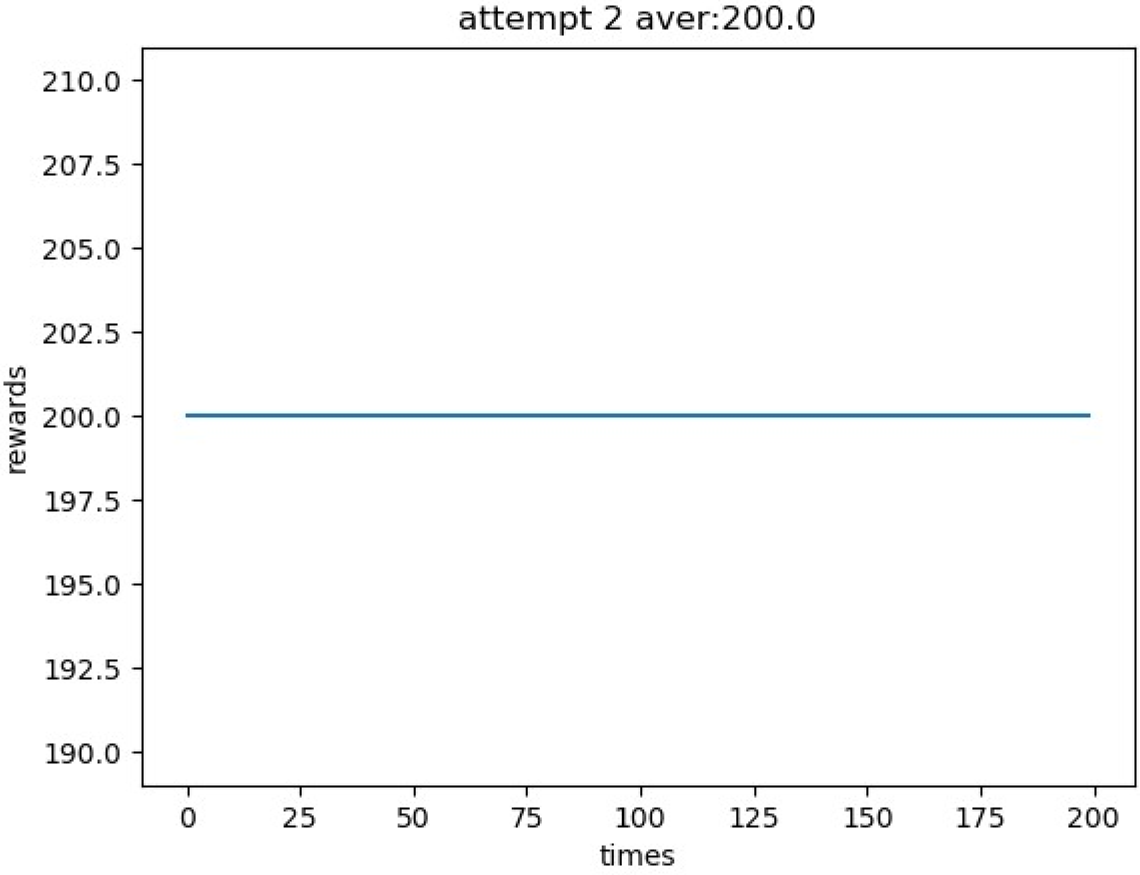
best solution

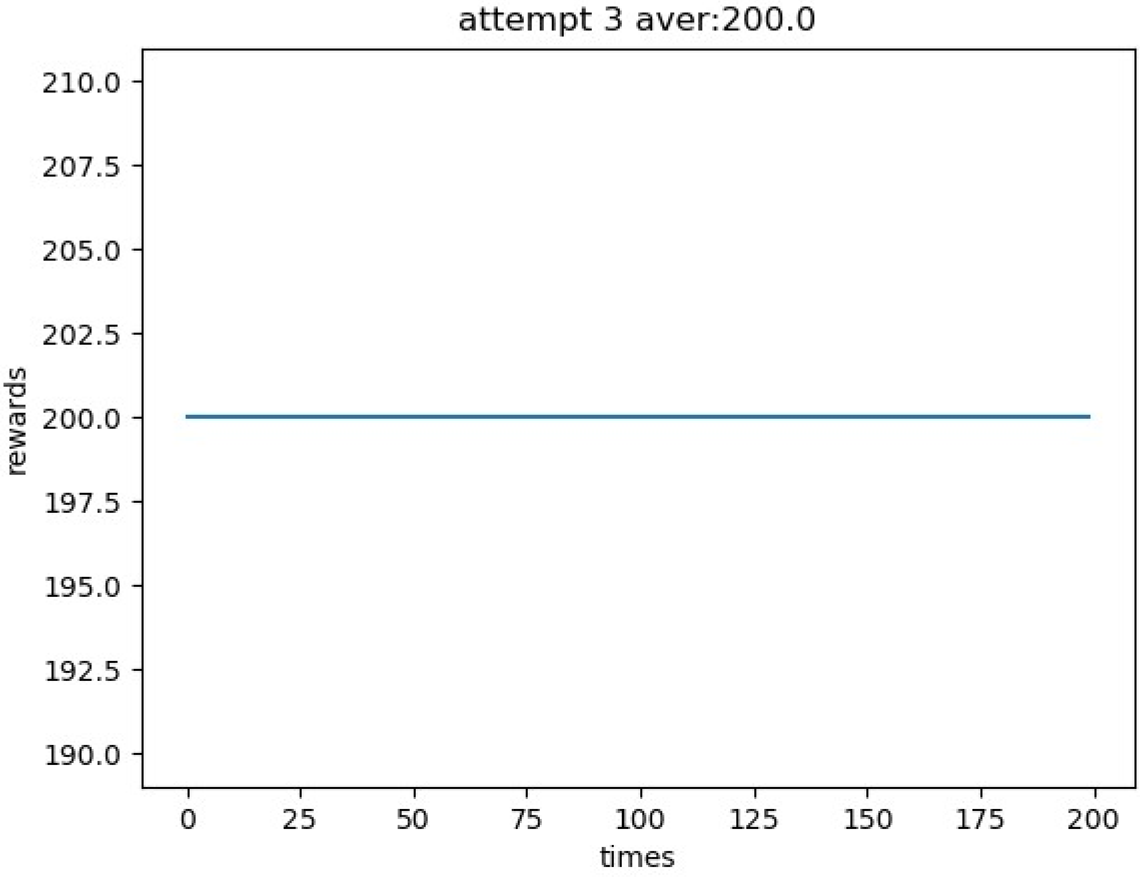
by monitoring omega and theta value
we found out that the best value for theta is 0.03
change line 44 to

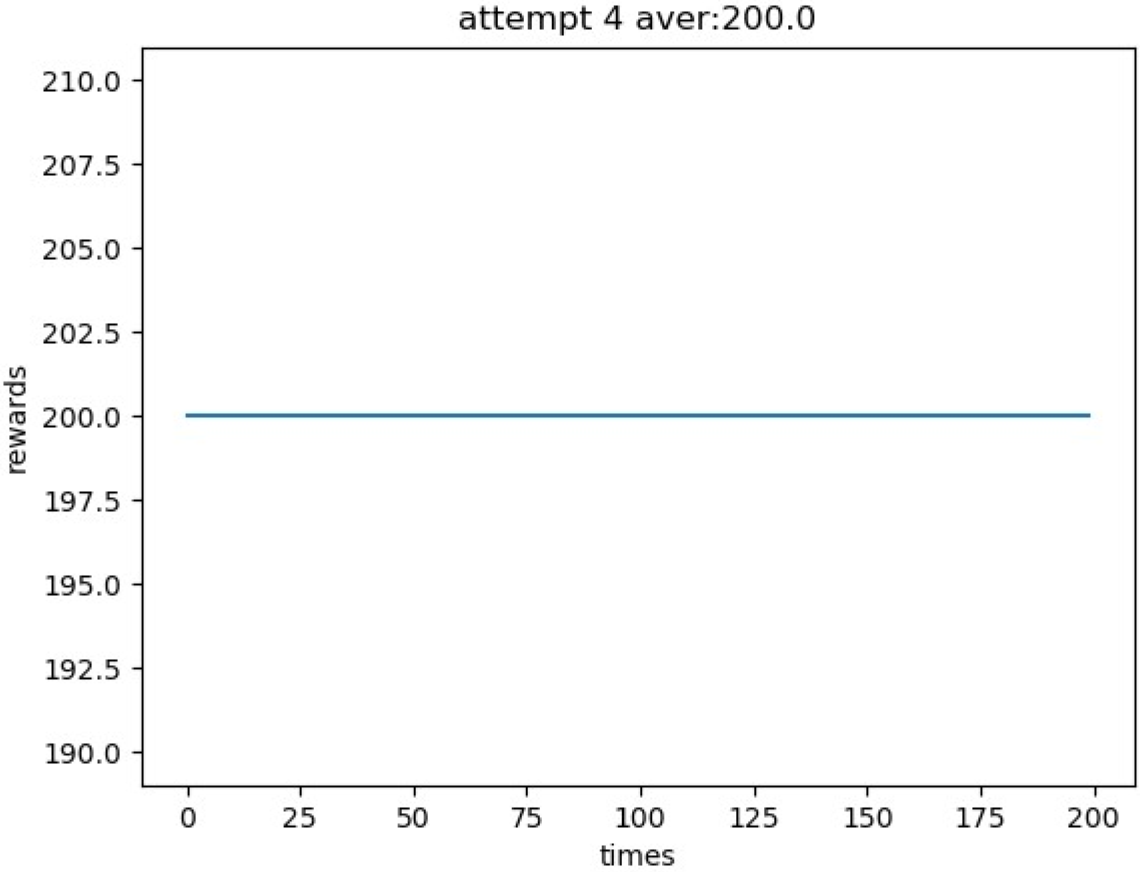
```
44  action=solve(observation)
```

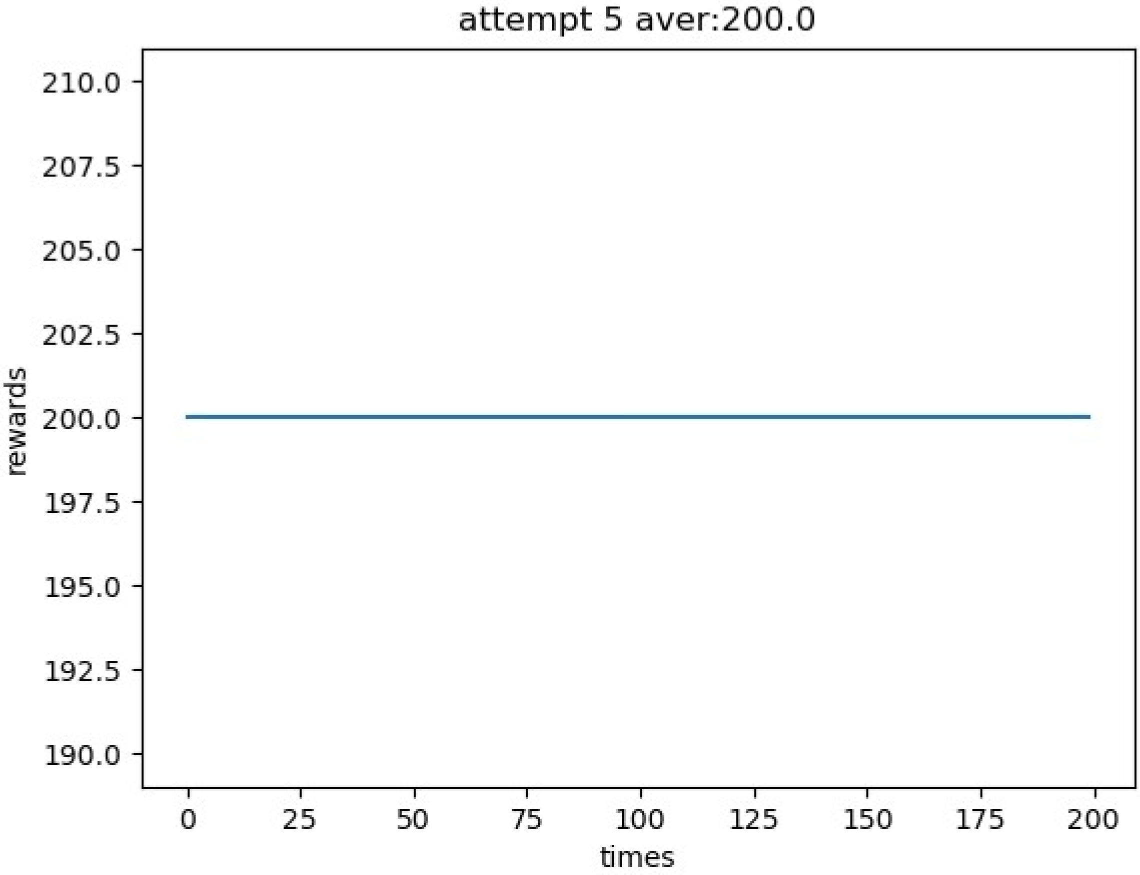
pic:




attempt 2 aver:200.0
rewards
210.0
207.5
205.0
202.5
200.0
197.5
195.0
192.5
190.0
0
25
50
75
100
125
150
175
200
times


attempt 3 aver:200.0
rewards
210.0
207.5
205.0
202.5
200.0
197.5
195.0
192.5
190.0
0
25
50
75
100
125
150
175
200
times


attempt 4 aver:200.0
rewards
210.0
207.5
205.0
202.5
200.0
197.5
195.0
192.5
190.0
0
25
50
75
100
125
150
175
200
times


attempt 5 aver:200.0
rewards
210.0
207.5
205.0
202.5
200.0
197.5
195.0
192.5
190.0
0
25
50
75
100
125
150
175
200
times

功德圓滿www

tags: ai