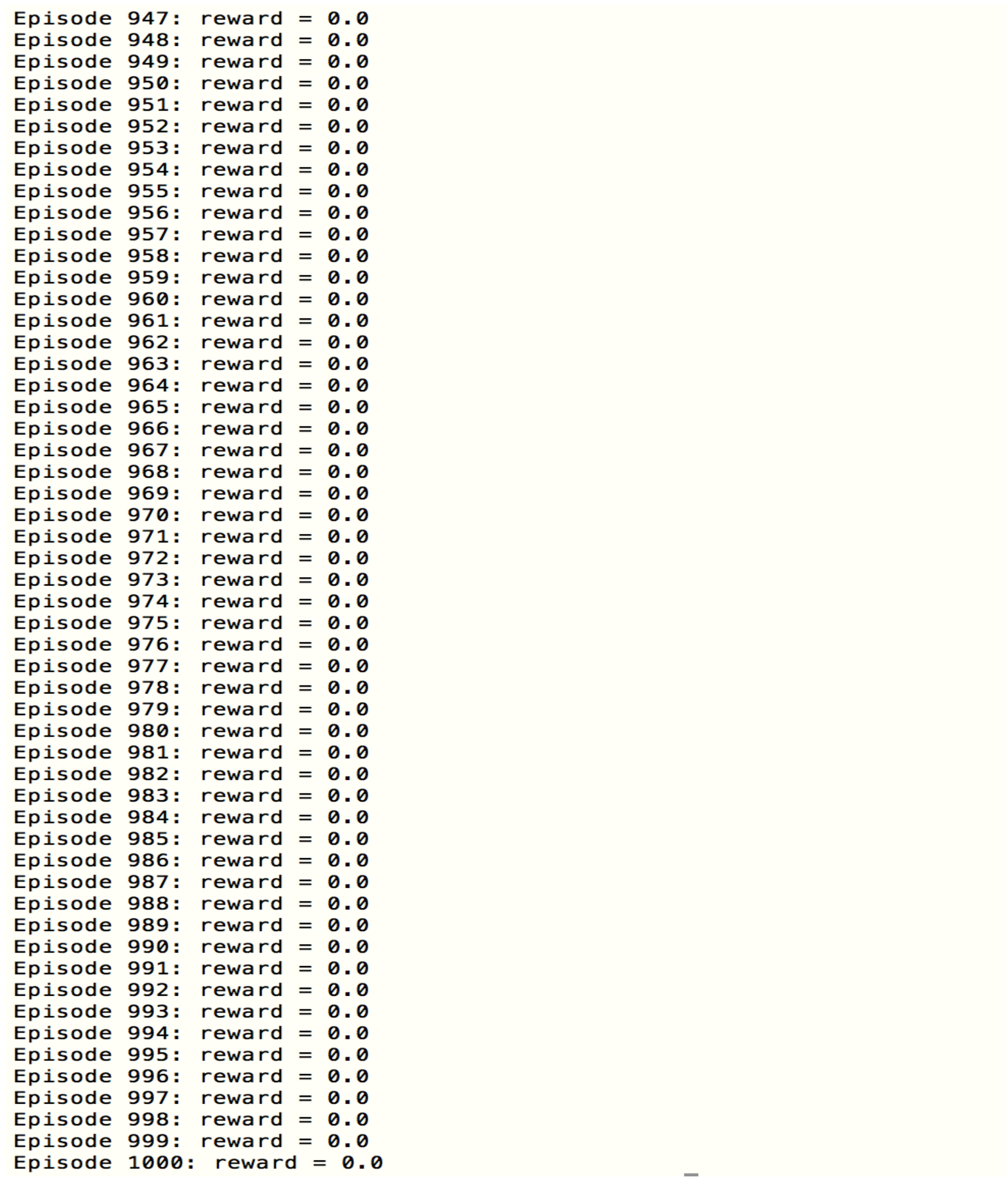
The Vinh Ha
CS 440
MP1

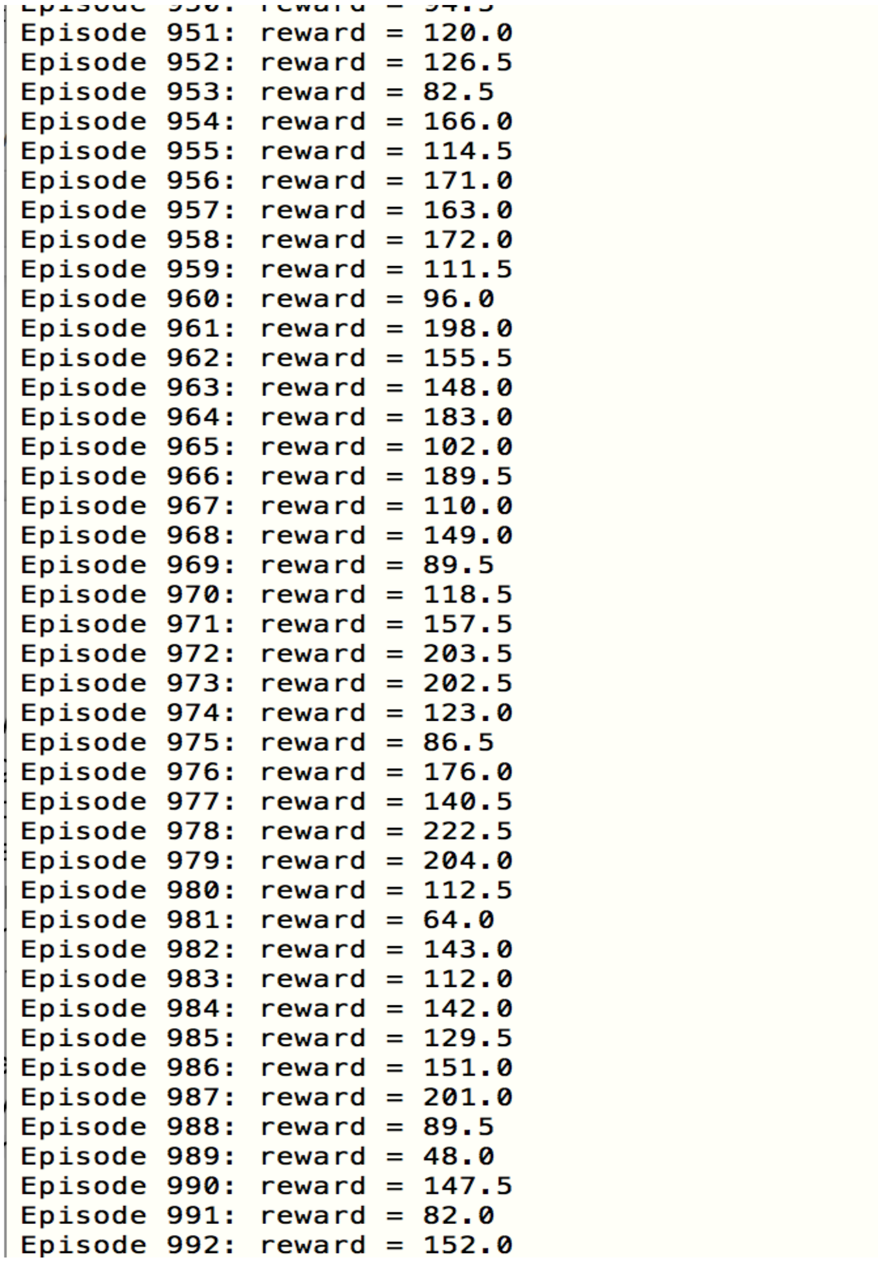
1) a) The reward after each episode is 0.0. Because the choose action method has the epsilon of 0.0, so the Q agent does not have any random exploration to explore any better alternatives. Moreover, the map itself, contains numerous tiles containing slipperiness. So as, the robot never take any risk, it is going to learn that it should not going anywhere because entering tiles with slipperiness will bring him the negative rewards for first few times.



```
Episode 947: reward = 0.0
Episode 948: reward = 0.0
Episode 949: reward = 0.0
Episode 950: reward = 0.0
Episode 951: reward = 0.0
Episode 952: reward = 0.0
Episode 953: reward = 0.0
Episode 954: reward = 0.0
Episode 955: reward = 0.0
Episode 956: reward = 0.0
Episode 957: reward = 0.0
Episode 958: reward = 0.0
Episode 959: reward = 0.0
Episode 960: reward = 0.0
Episode 961: reward = 0.0
Episode 962: reward = 0.0
Episode 963: reward = 0.0
Episode 964: reward = 0.0
Episode 965: reward = 0.0
Episode 966: reward = 0.0
Episode 967: reward = 0.0
Episode 968: reward = 0.0
Episode 969: reward = 0.0
Episode 970: reward = 0.0
Episode 971: reward = 0.0
Episode 972: reward = 0.0
Episode 973: reward = 0.0
Episode 974: reward = 0.0
Episode 975: reward = 0.0
Episode 976: reward = 0.0
Episode 977: reward = 0.0
Episode 978: reward = 0.0
Episode 979: reward = 0.0
Episode 980: reward = 0.0
Episode 981: reward = 0.0
Episode 982: reward = 0.0
Episode 983: reward = 0.0
Episode 984: reward = 0.0
Episode 985: reward = 0.0
Episode 986: reward = 0.0
Episode 987: reward = 0.0
Episode 988: reward = 0.0
Episode 989: reward = 0.0
Episode 990: reward = 0.0
Episode 991: reward = 0.0
Episode 992: reward = 0.0
Episode 993: reward = 0.0
Episode 994: reward = 0.0
Episode 995: reward = 0.0
Episode 996: reward = 0.0
Episode 997: reward = 0.0
Episode 998: reward = 0.0
Episode 999: reward = 0.0
Episode 1000: reward = 0.0
```

The Vinh Ha
CS 440
MP1

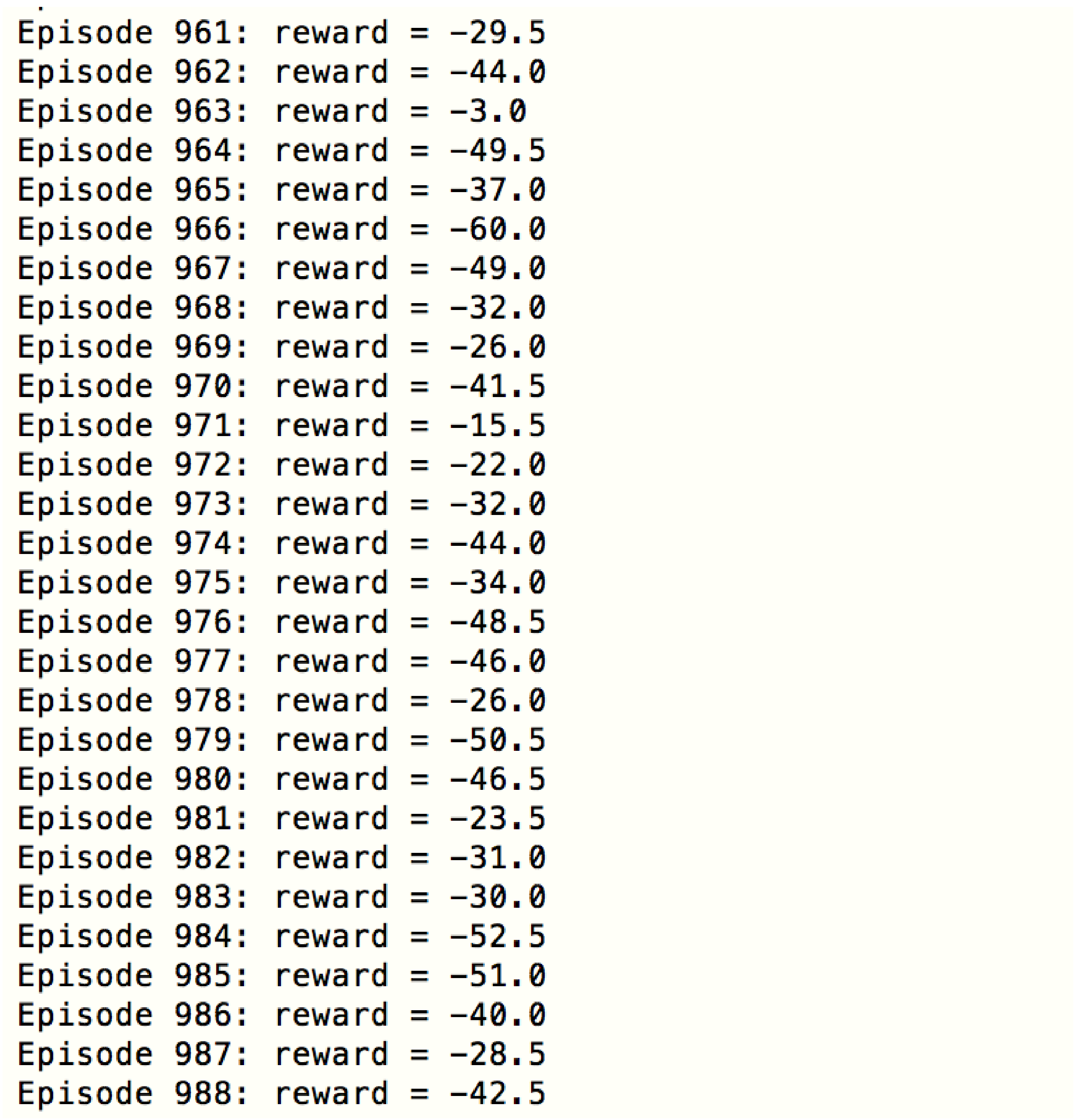
b) The performance is way better this time. As the choose action method has the epsilon of 0.1, Q Learning agent is able to take risk to learn more and update the rewards of the Q value array. So it seems to be an acceptable amount of random exploration. The learning agent now realizes that tiles with slipperiness is not bad after all, so after a few turns of negative rewards, it decides to get over the middle line which contains all the slipperiness to deliver the packages.



```
Episode 951: reward = 120.0
Episode 952: reward = 126.5
Episode 953: reward = 82.5
Episode 954: reward = 166.0
Episode 955: reward = 114.5
Episode 956: reward = 171.0
Episode 957: reward = 163.0
Episode 958: reward = 172.0
Episode 959: reward = 111.5
Episode 960: reward = 96.0
Episode 961: reward = 198.0
Episode 962: reward = 155.5
Episode 963: reward = 148.0
Episode 964: reward = 183.0
Episode 965: reward = 102.0
Episode 966: reward = 189.5
Episode 967: reward = 110.0
Episode 968: reward = 149.0
Episode 969: reward = 89.5
Episode 970: reward = 118.5
Episode 971: reward = 157.5
Episode 972: reward = 203.5
Episode 973: reward = 202.5
Episode 974: reward = 123.0
Episode 975: reward = 86.5
Episode 976: reward = 176.0
Episode 977: reward = 140.5
Episode 978: reward = 222.5
Episode 979: reward = 204.0
Episode 980: reward = 112.5
Episode 981: reward = 64.0
Episode 982: reward = 143.0
Episode 983: reward = 112.0
Episode 984: reward = 142.0
Episode 985: reward = 129.5
Episode 986: reward = 151.0
Episode 987: reward = 201.0
Episode 988: reward = 89.5
Episode 989: reward = 48.0
Episode 990: reward = 147.5
Episode 991: reward = 82.0
Episode 992: reward = 152.0
```

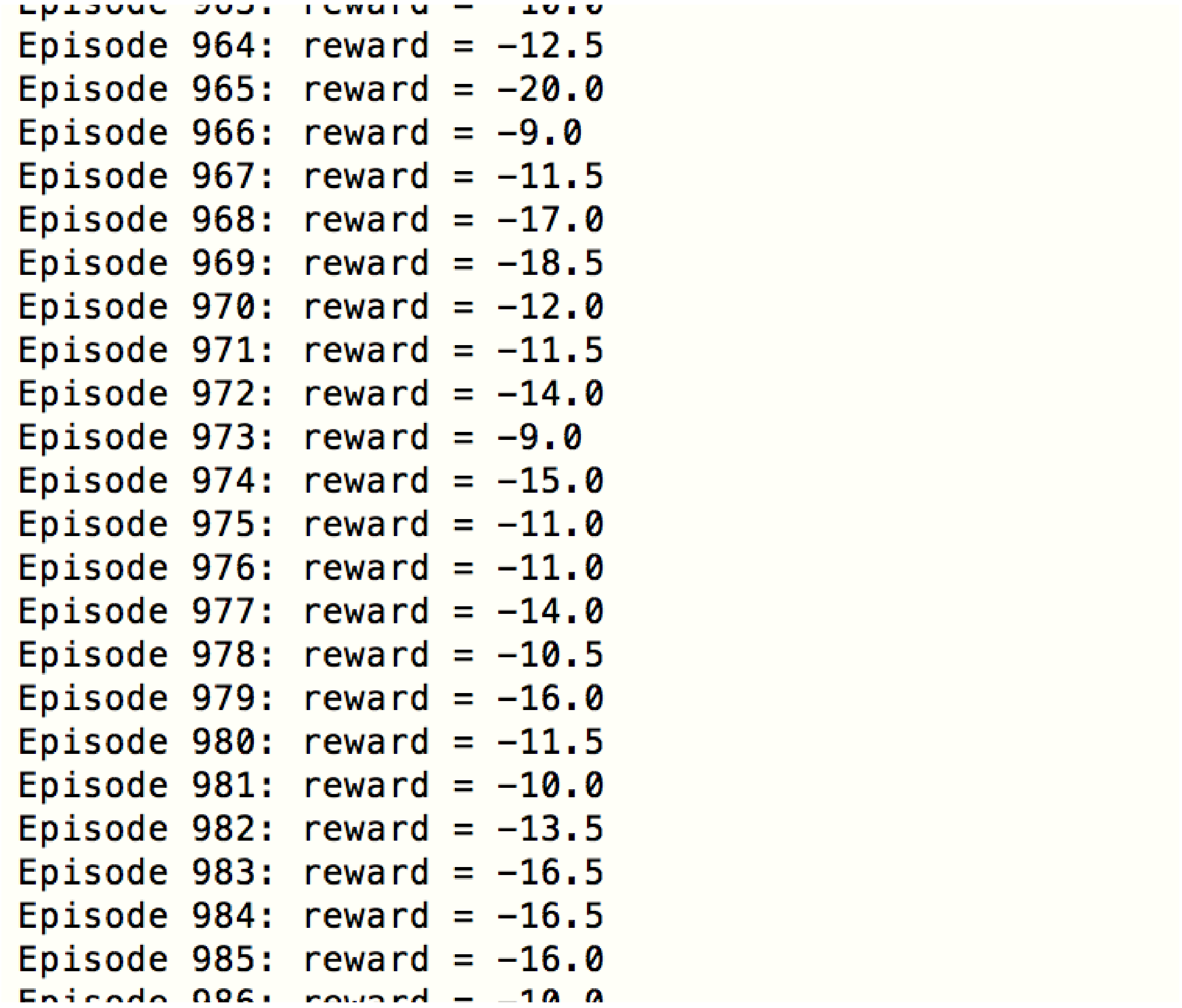
c) The performance is actually worse than the epsilon 0.0 trial in which most of the rewards after each episode is negative. It's because our Q Learning takes too much risk than it should. The epsilon is too high that it doesn't follow the best policy that it should follow after a few episodes of training. It makes our agent quickly converges into a large poor region of π^*

The Vinh Ha
CS 440
MP1



```
Episode 961: reward = -29.5
Episode 962: reward = -44.0
Episode 963: reward = -3.0
Episode 964: reward = -49.5
Episode 965: reward = -37.0
Episode 966: reward = -60.0
Episode 967: reward = -49.0
Episode 968: reward = -32.0
Episode 969: reward = -26.0
Episode 970: reward = -41.5
Episode 971: reward = -15.5
Episode 972: reward = -22.0
Episode 973: reward = -32.0
Episode 974: reward = -44.0
Episode 975: reward = -34.0
Episode 976: reward = -48.5
Episode 977: reward = -46.0
Episode 978: reward = -26.0
Episode 979: reward = -50.5
Episode 980: reward = -46.5
Episode 981: reward = -23.5
Episode 982: reward = -31.0
Episode 983: reward = -30.0
Episode 984: reward = -52.5
Episode 985: reward = -51.0
Episode 986: reward = -40.0
Episode 987: reward = -28.5
Episode 988: reward = -42.5
```

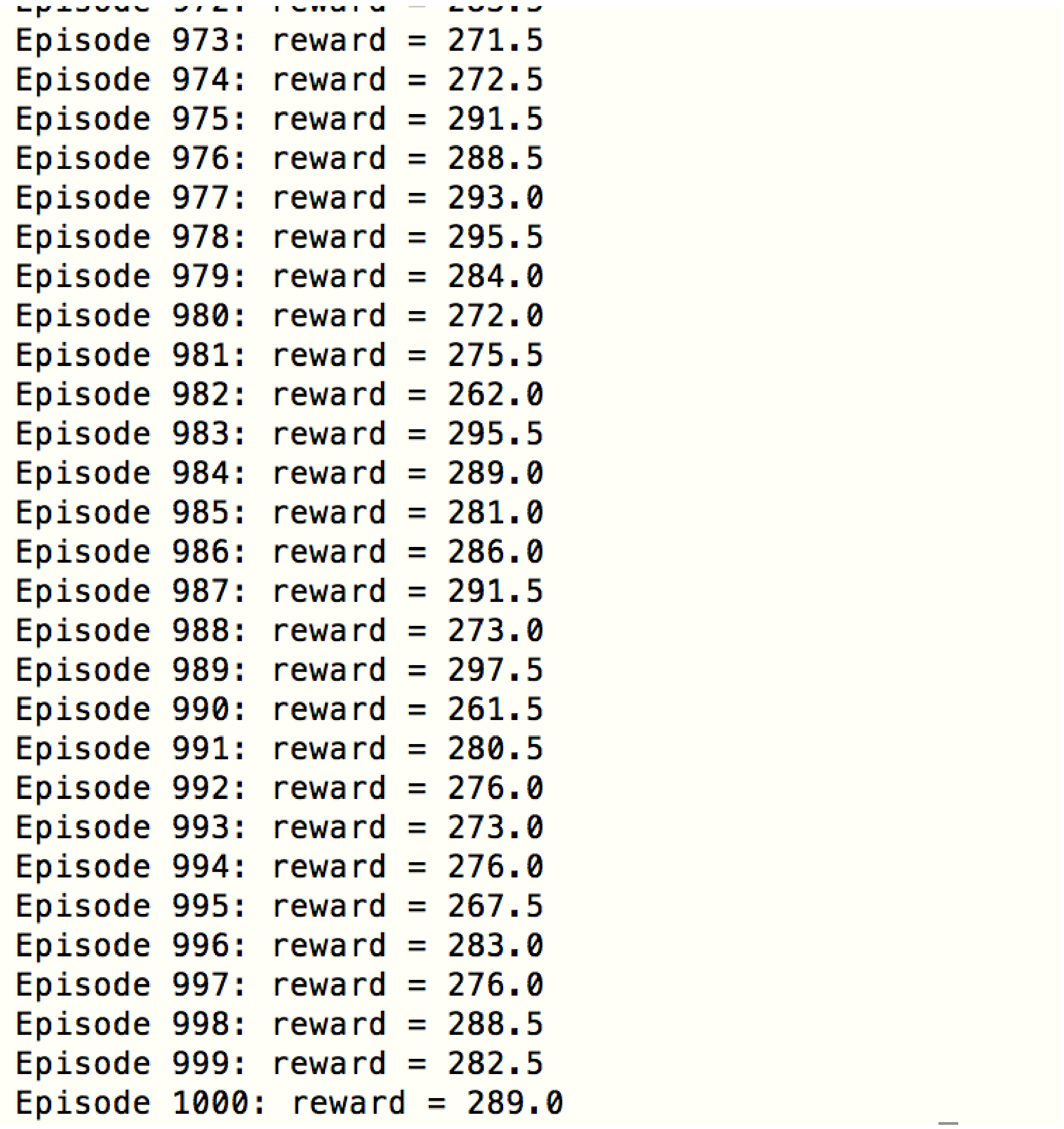
2) a) The reward after each episode in this case is mostly less than 0. Our Q-learning agent is still learning, however, it is not that effective because our agent is not sensitive to the position of the thief. In other words, in the file PolicySimulator.java. If the learning knows_thief is n, then it would not call the function getThiefRow() which allows our agent to know the current position of the thief. Therefore, it has higher chance of bumping into the thief. The learning algorithm is not working because it can't know the pattern of this thief's position.



```
Episode 963: reward = -10.0
Episode 964: reward = -12.5
Episode 965: reward = -20.0
Episode 966: reward = -9.0
Episode 967: reward = -11.5
Episode 968: reward = -17.0
Episode 969: reward = -18.5
Episode 970: reward = -12.0
Episode 971: reward = -11.5
Episode 972: reward = -14.0
Episode 973: reward = -9.0
Episode 974: reward = -15.0
Episode 975: reward = -11.0
Episode 976: reward = -11.0
Episode 977: reward = -14.0
Episode 978: reward = -10.5
Episode 979: reward = -16.0
Episode 980: reward = -11.5
Episode 981: reward = -10.0
Episode 982: reward = -13.5
Episode 983: reward = -16.5
Episode 984: reward = -16.5
Episode 985: reward = -16.0
Episode 986: reward = -10.0
```

b) The performance of the agent is much better as the rewards after each episodes are mostly positive. Now knows_thief is set to yes, so the agent would call the function getThiefRow() to avoid the thief by knowing the current location of the thief. Therefore, it now has much less chance to bump into the thief and receive negative rewards, and the learning of the robot is also getting better because it knows the pattern of the thief.

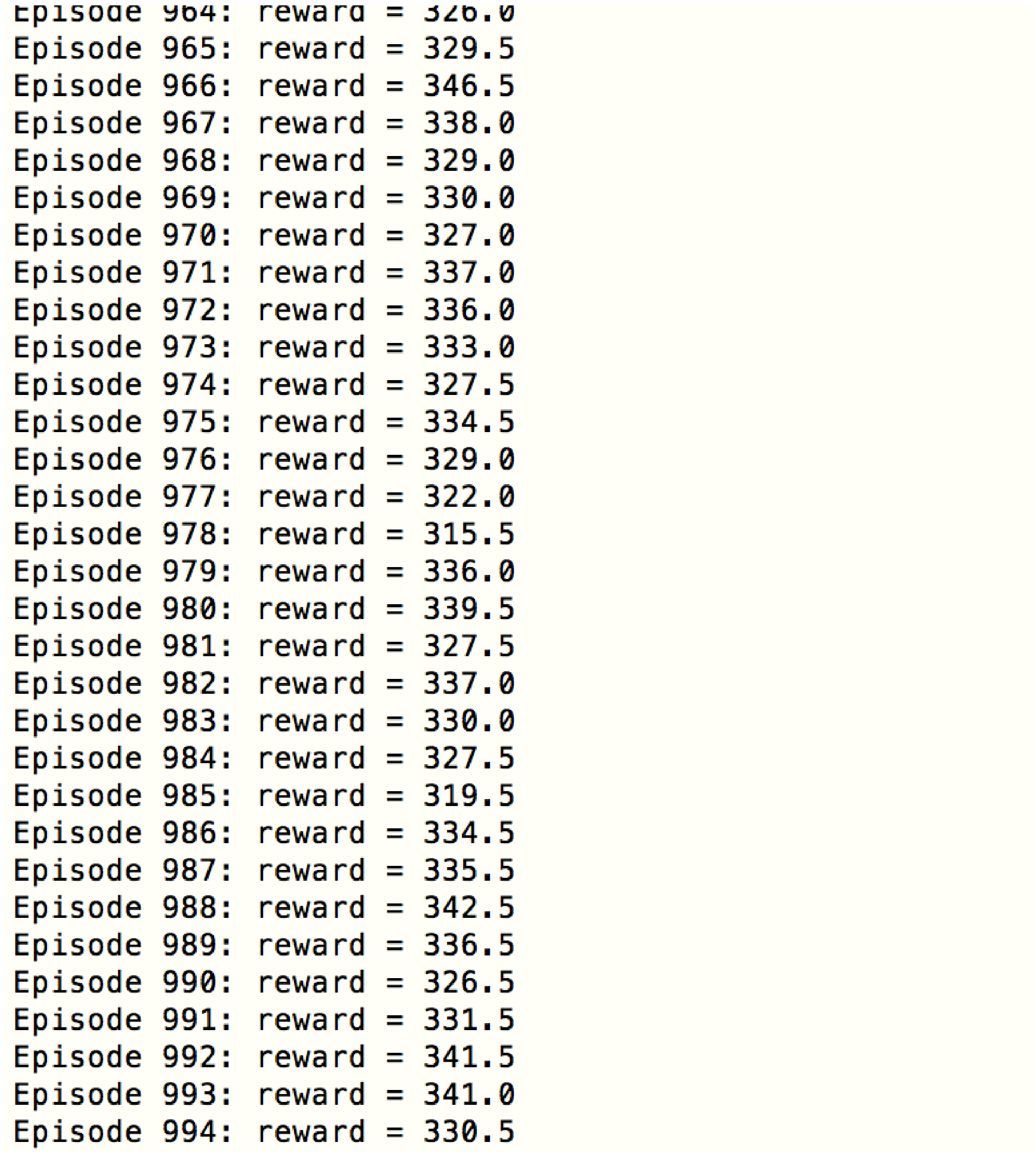
The Vinh Ha
CS 440
MP1



```
Episode 973:  reward = 271.5
Episode 974:  reward = 272.5
Episode 975:  reward = 291.5
Episode 976:  reward = 288.5
Episode 977:  reward = 293.0
Episode 978:  reward = 295.5
Episode 979:  reward = 284.0
Episode 980:  reward = 272.0
Episode 981:  reward = 275.5
Episode 982:  reward = 262.0
Episode 983:  reward = 295.5
Episode 984:  reward = 289.0
Episode 985:  reward = 281.0
Episode 986:  reward = 286.0
Episode 987:  reward = 291.5
Episode 988:  reward = 273.0
Episode 989:  reward = 297.5
Episode 990:  reward = 261.5
Episode 991:  reward = 280.5
Episode 992:  reward = 276.0
Episode 993:  reward = 273.0
Episode 994:  reward = 276.0
Episode 995:  reward = 267.5
Episode 996:  reward = 283.0
Episode 997:  reward = 276.0
Episode 998:  reward = 288.5
Episode 999:  reward = 282.5
Episode 1000:  reward = 289.0
```

c) My best learning rate and epsilon is 0.15 and 0.01, respectively. My investigation at first is to increase epsilon rate. But I find the results getting lower each time I increase epsilon. Then I decide, probably I should decrease the epsilon rates, It is quite surprising that the rewards after each episode increase when I decrease the epsilon rate. So it hit the highest average rewards about over 300 when my epsilon is 0.01. After that I play around with the learning rate, I increase the learning rate with 0.05 step. At the end, I feel like learning rate which is 0.15, gives me the most, so I choose 0.15 as my learning rate.

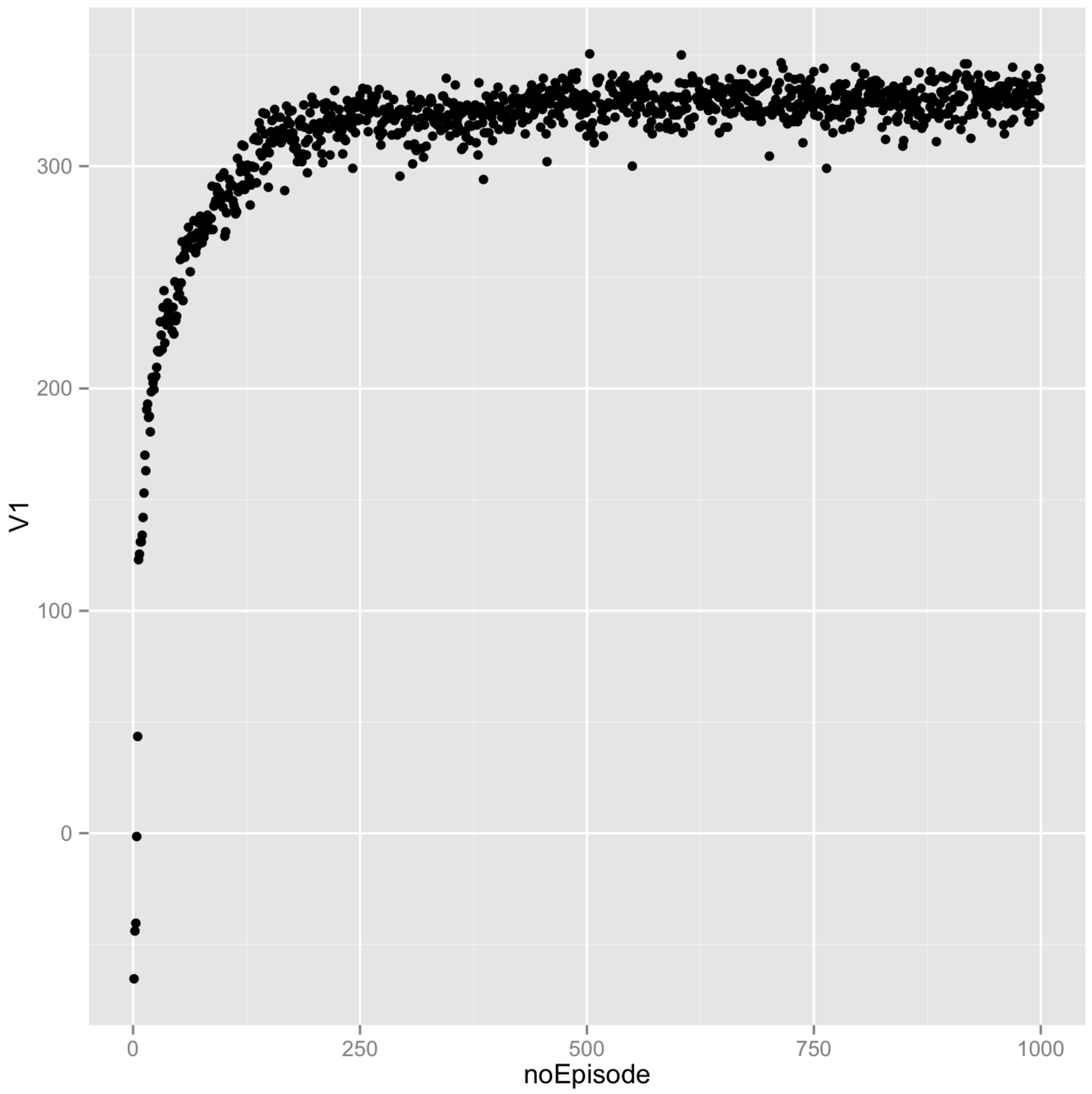
The Vinh Ha
CS 440
MP1



```
Episode 964:  reward = 326.0
Episode 965:  reward = 329.5
Episode 966:  reward = 346.5
Episode 967:  reward = 338.0
Episode 968:  reward = 329.0
Episode 969:  reward = 330.0
Episode 970:  reward = 327.0
Episode 971:  reward = 337.0
Episode 972:  reward = 336.0
Episode 973:  reward = 333.0
Episode 974:  reward = 327.5
Episode 975:  reward = 334.5
Episode 976:  reward = 329.0
Episode 977:  reward = 322.0
Episode 978:  reward = 315.5
Episode 979:  reward = 336.0
Episode 980:  reward = 339.5
Episode 981:  reward = 327.5
Episode 982:  reward = 337.0
Episode 983:  reward = 330.0
Episode 984:  reward = 327.5
Episode 985:  reward = 319.5
Episode 986:  reward = 334.5
Episode 987:  reward = 335.5
Episode 988:  reward = 342.5
Episode 989:  reward = 336.5
Episode 990:  reward = 326.5
Episode 991:  reward = 331.5
Episode 992:  reward = 341.5
Episode 993:  reward = 341.0
Episode 994:  reward = 330.5
```

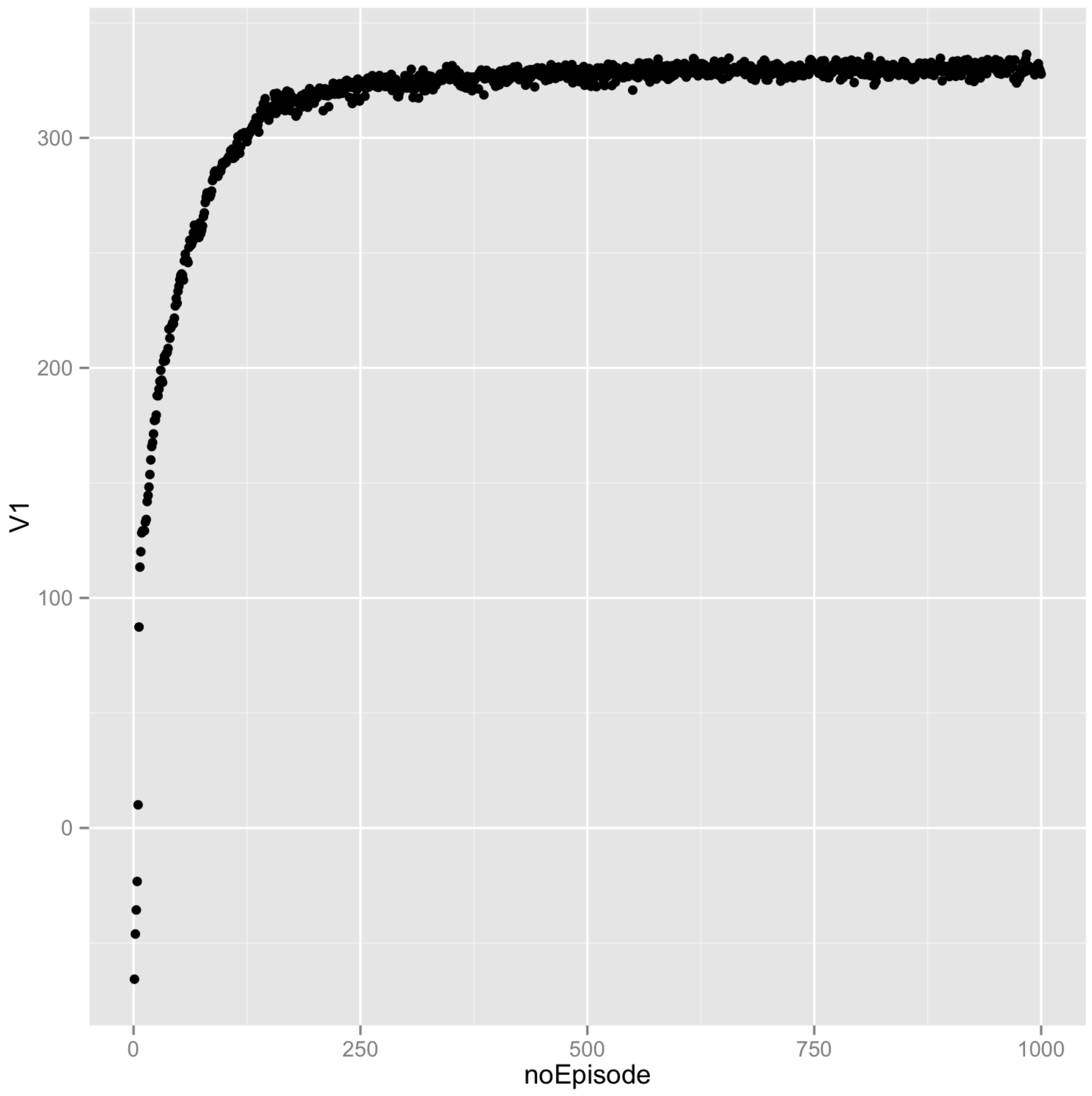
The Vinh Ha
CS 440
MP1

3) The graph for the first time running the code with epsilon 0.01 and learning rate 0.15 is:



V1 here is the total rewards after each episode.
Sometimes, in one episode, there is some sort of outliers. This is probably when the Agent decides to pick the random direction when the random rate is less than epsilon. Overall, after a few turns of learning the data, most of the episodes end up with rewards larger than 300.

The Vinh Ha
CS 440
MP1



V1 is the total rewards after each episode.
The graph seems thinner. There are less outliers than the first graph. This is after ten graphs so, average of the rewards after each episode is gonna make the graph look cleaner which is great for our searching for the best epsilon and learning rate. It’s like, 10 times average graph has cancelled out most of the outliers we saw in the first graph.