최종과제 구현보고서

Simple과 original 메소드를 구현하였다.

모델 설명

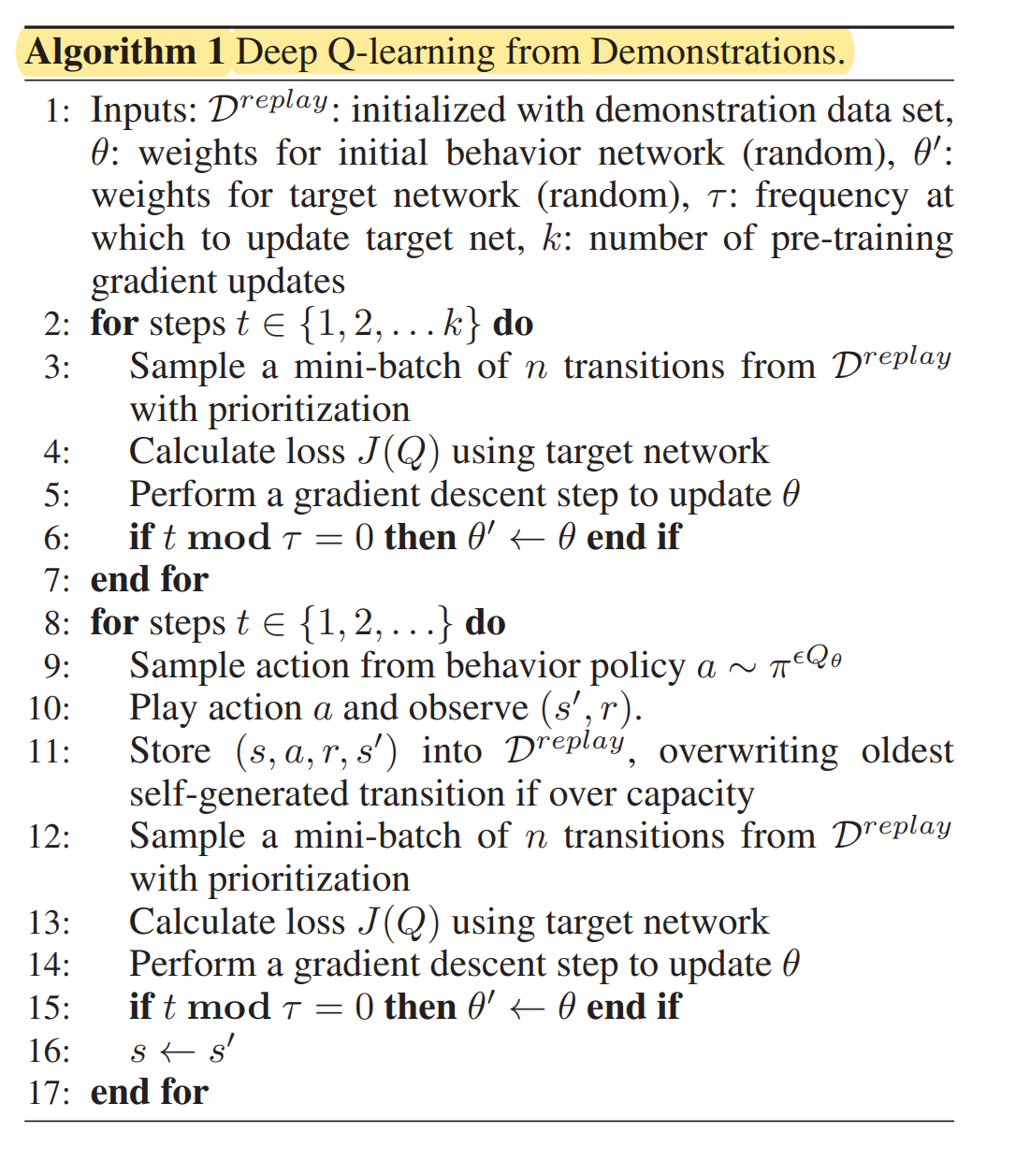
Linear 층 세개로 이루어진 다층 퍼셉트론

은닉층의 뉴런 개수는 30개로 정함

Xavier 균등분포 초기화를 사용하였음

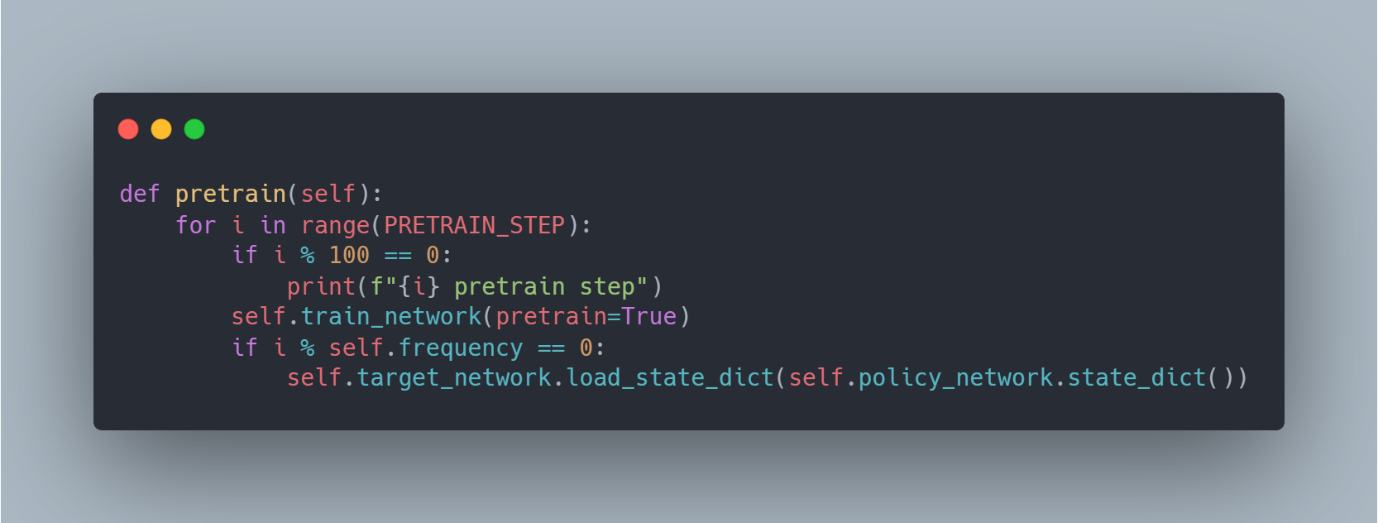
Adam 최적화 함수를 사용하고, L2 손실 대신 MSE 손실 함수를 사용하였음

알고리즘과 코드 비교



2-6줄 pretrain 단계:

100회마다 출력, 정책 네트워크를 훈련시키고 frequency회마다 타겟 네트워크를 업데이트



손실 함수 계산이 포함된 학습 과정

미니배치를 샘플링하고 손실 함수 네가지를 계산한다.

기댓값을 의미하는 Q 함수를 신경망으로 근사하는 DQN 방식을 사용

람다값은 임의로 0.23으로 정하고, 마진 함수의 양수값은 임의로 0.2로 정했음

Q값 계산은 논문의 식 그대로 사용

n-step returns는 미니배치에 있는 다음 값들을 사용

loss 함수는 람다 계수를 취한 모든 손실 함수들의 합

loss 함수를 구한 뒤 오차역전파법으로 backward

그다음에 최적화 함수로 학습한다

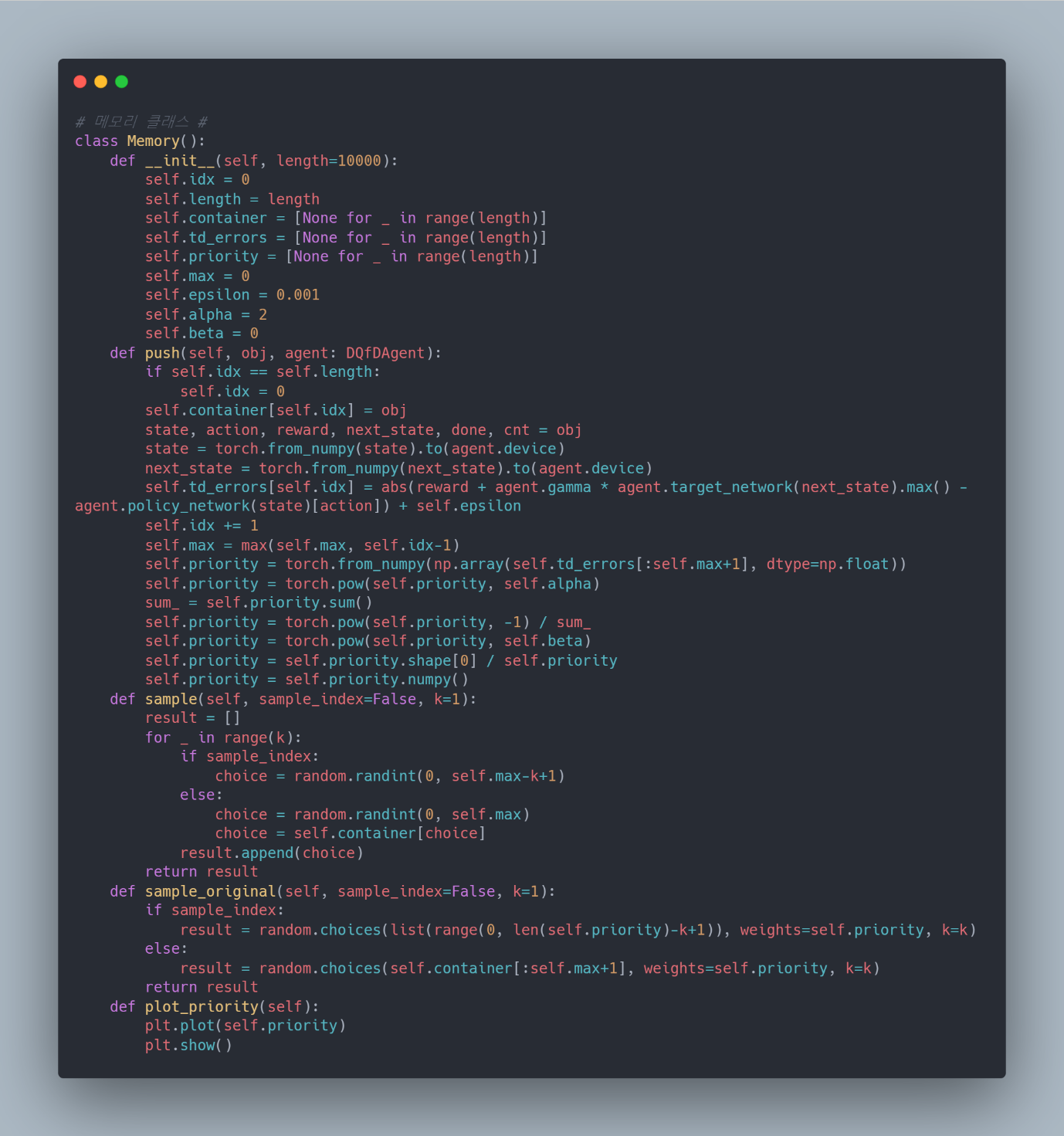


미니배치 샘플링, 메모리 클래스

버퍼를 만들고 인덱싱하는데, 버퍼가 꽉 찼을시에는 처음 인덱스부터 덮어쓴다

push할때 PER를 계산한다.

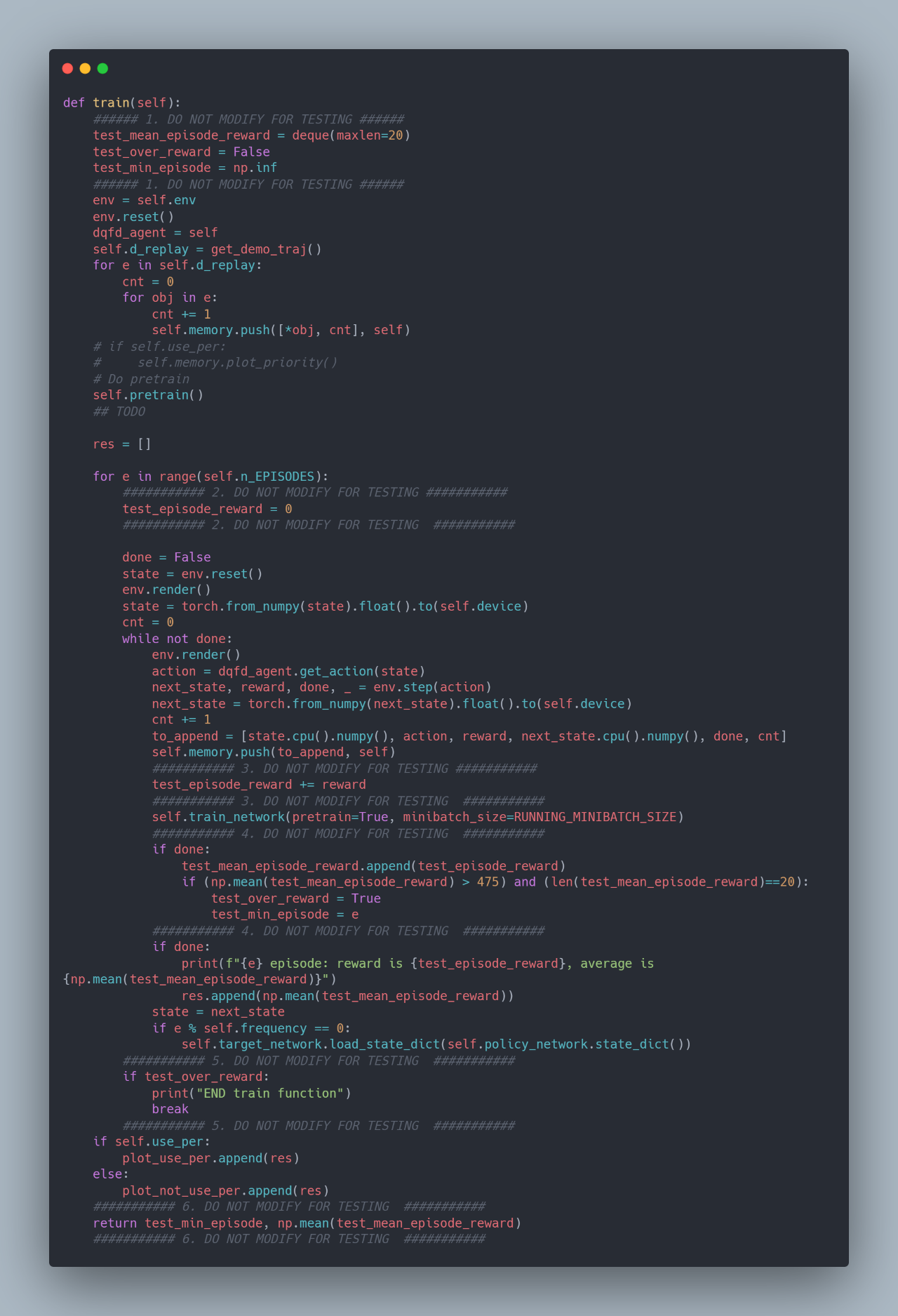
Original일 경우에는 sample\_original, simple일 경우에는 sample 메소드로 샘플링을 하는데 PER을 가중치로 쓰는 것의 유무라고 볼 수 있다



train 메소드

액션을 취하고 메모리에 삽입, 일부 로깅을 적용함

에피소드 횟수만큼 반복하면서 액션을 가져오고 미니배치 학습을 한다.

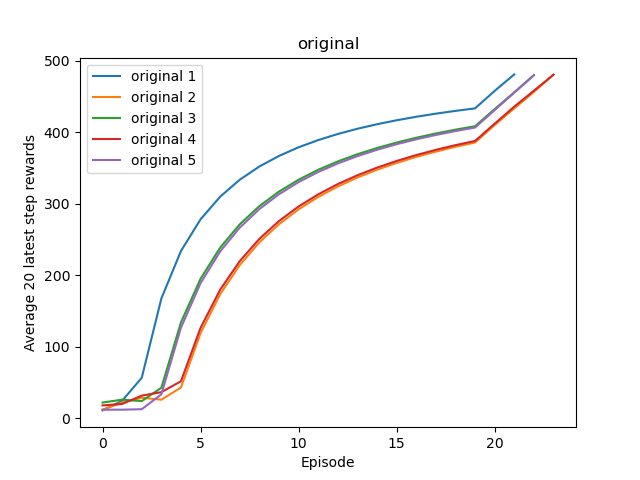
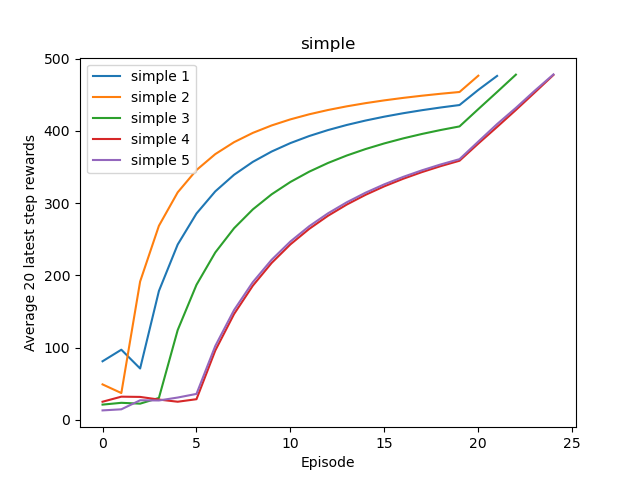


PER 알고리즘에 관해

PER은 미니배치 샘플링을 할 때 영향을 미치는 weight 가중치를 계산하는 알고리즘으로, simple의 경우에는 PER이 미적용된 단순 균등분포 샘플링

Original은 weight에 따라서 확률 가중치가 적용된다.

그래프는 pyplot을 agent\_1에 메인 함수를 생성하고 연동해 작성하였다.



PER을 적용한 original이 simple 메소드보다 제감가능한 성능 향상이 있음을 확인할 수 있었다.