# **[과제 #1] 수행시간 비교 – 분석 레포트**

202002397 통계학과 이도은

수행 환경은 구글 colab을 이용하였다.

텍스트이(가) 표시된 사진

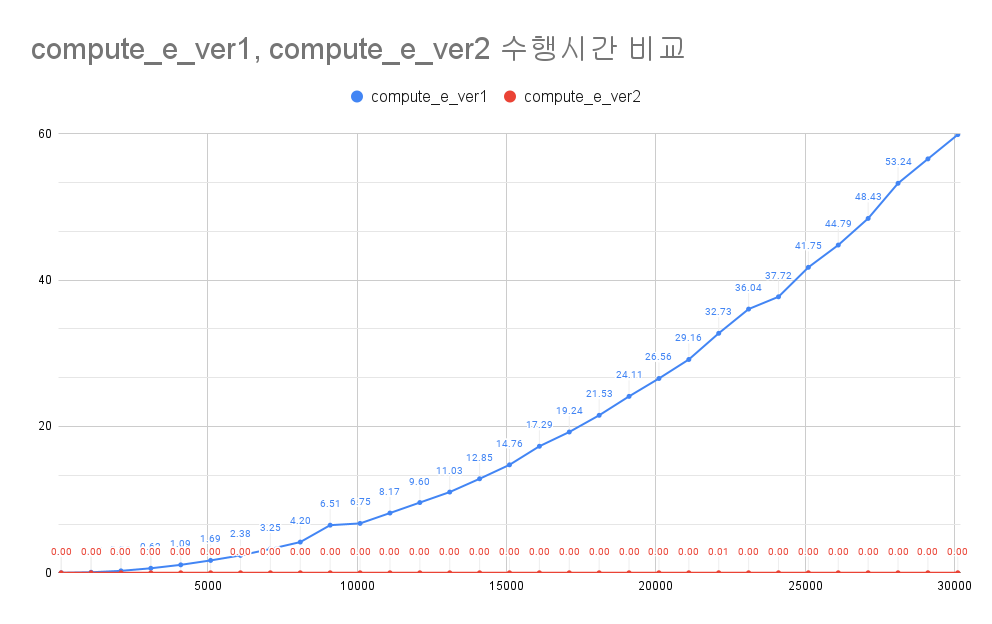
자동 생성된 설명

Compute\_e\_ver\_1는 이중 for루프문이 포함되어 있다. Worst case running time을 계산해보았을 때 수행시간의 최고 차항은 이다. 따라서 이를 Big-O표기법으로 표현하면 으로 표기할 수 있다.

Compute\_e\_ver\_2는 단일 for 루프문이 포함 되어있다. 위와 같이 Worst case running time을 계산해보았을 때 수행시간의 최고 차항은 이고 이를 Big-O 표기법으로 표현하면으로 표기할 수 있다.

이제 각 함수를 실행하여 n 값이 변함에 따라 실제 수행시간이 어떻게 바뀌는지 실습해본다. 문제에서 제시한대로 n = 100~500000까지 범위 내에서 시행했다. for문을 이용하여 100부터 시작하여 1000 단위로 증가하며 60초를 초과하지 않는 n까지 수행시간을 측정했다.

수행시간을 측정해보았을 때, n = 30100일 때 compute\_e\_ver1의 수행시간이 59.88로 이후의 n에서는 60초를 초과하게 되어 측정을 중단했다. 이를 시각화한 차트를 보았을 때 n이 증가함에 따라, compute\_e\_ver1의 수행시간이 폭발적으로 증가하는 것을 볼 수 있다. 즉, n이 증가할 수록 수행시간이 인 compute\_e\_ver1이 compute\_e\_ver2 보다 극명하게 수행시간이 차이남을 확인할 수 있었다.





[느낀점]

이를 통해 코드를 짜는 사람이 얼마나 효율적으로 짜는 가는 입력 값의 크기가 커질 때 더욱 중요해짐을 알 수 있었다. 이때까지 코딩하면서 입력 값의 크기는 크지 않았기 때문에 수행시간이 크게 문제가 된 적은 없었다. 그러나 앞으로 개발자로써 성장하면서 입력 값이 매우 큰 상황들을 마주하게 될 것이고 이런 상황에 대비하여 효율적으로 코드를 짜는 법을 길러야 한다는 생각이 들게 해준 유익한 실습이었다.