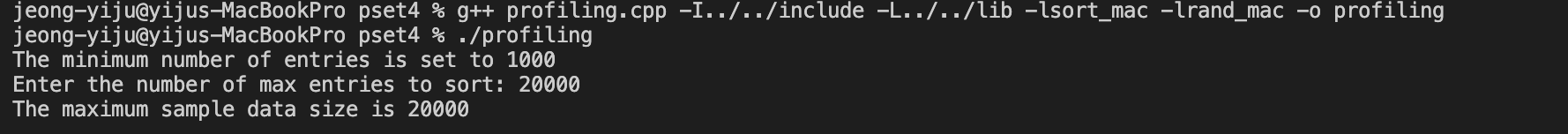
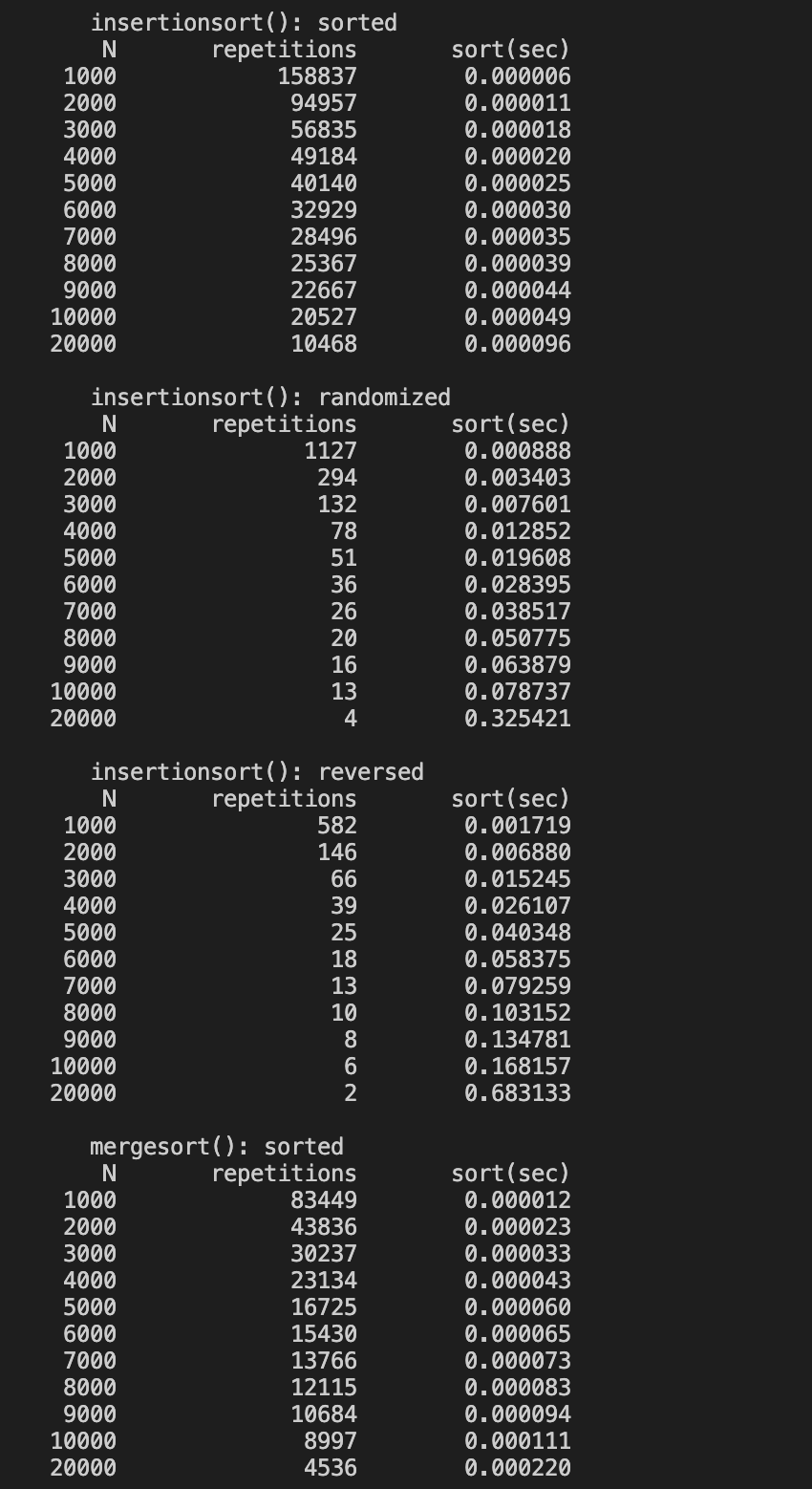
// On my honour, I pledge that I have neither received nor provided improper assistance in the completion of this assignment. Signed: \_\_\_\_Jungyiju\_\_\_\_\_\_ Student Number: \_\_\_\_22000690\_\_\_\_\_

1. Screen capture of profiling.exe output.





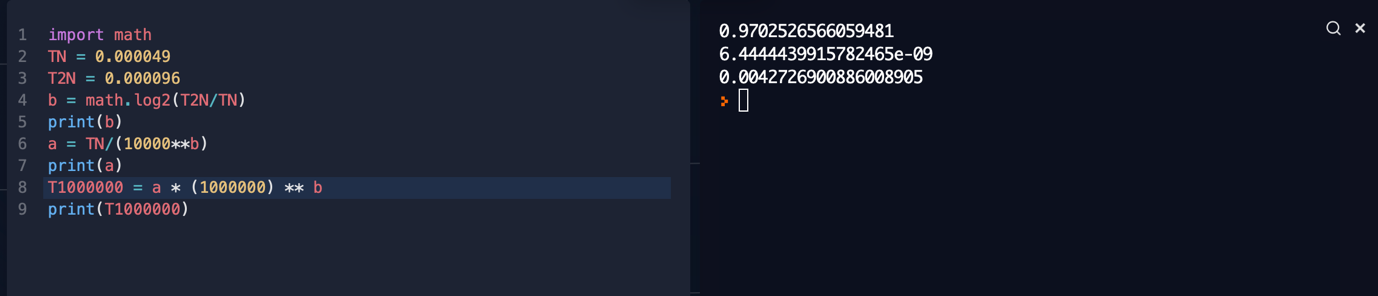


텍스트이(가) 표시된 사진

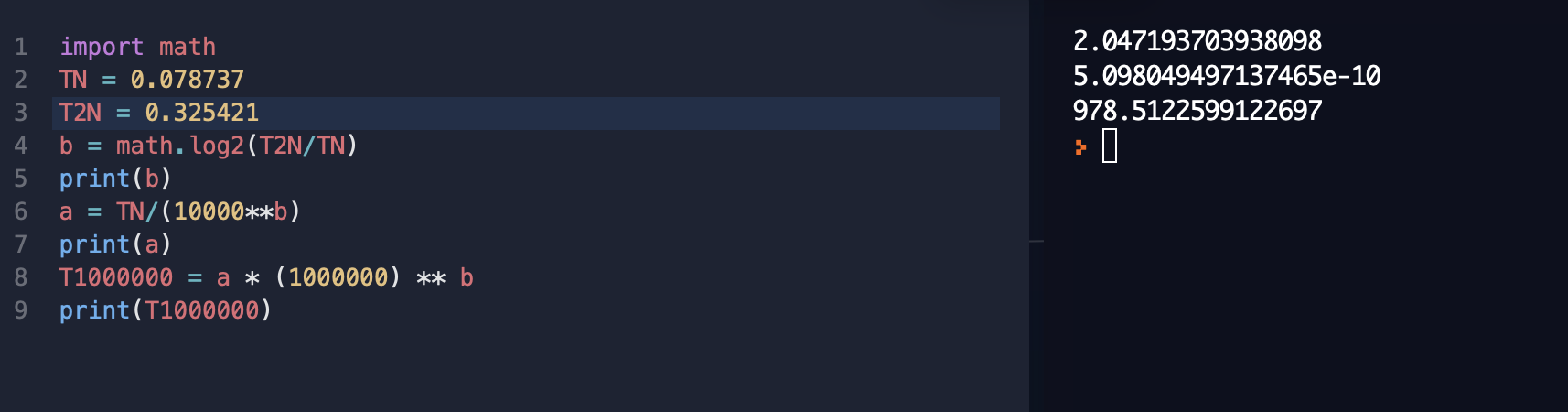
자동 생성된 설명

1. Complete the performance analysis tables with your data – estimated and measured. Show your computation steps for a, b and estimated time for all **five cases** (worst, average and best cases for insertionsort, and two average cases for mergesort and quicksort)

Insertionsort – Best



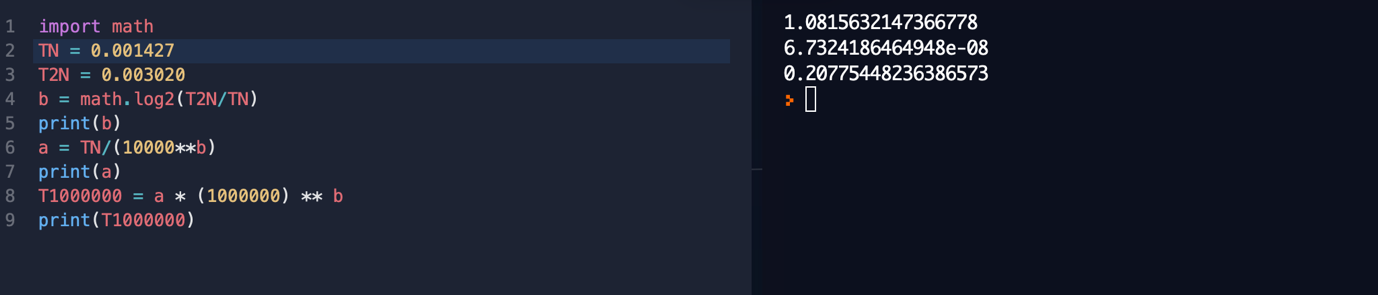
Insertionsort -Average



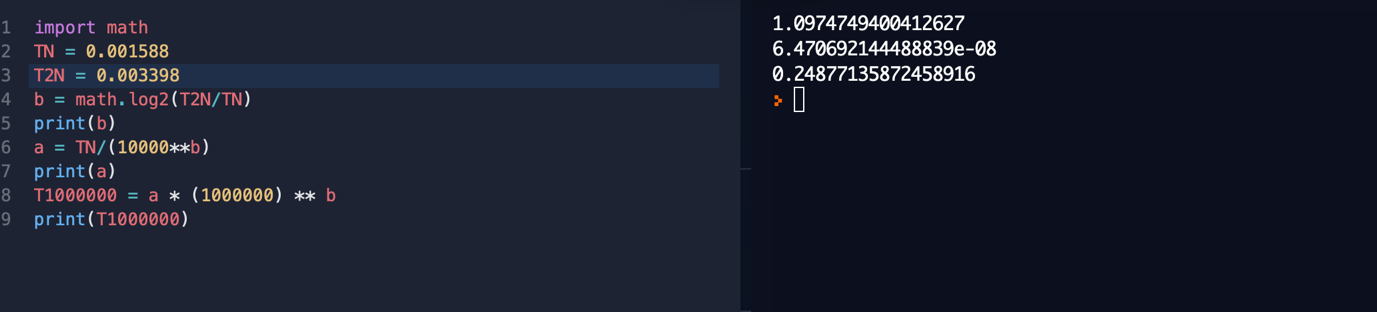
Insertionsort – Worst



Quicksort – Average

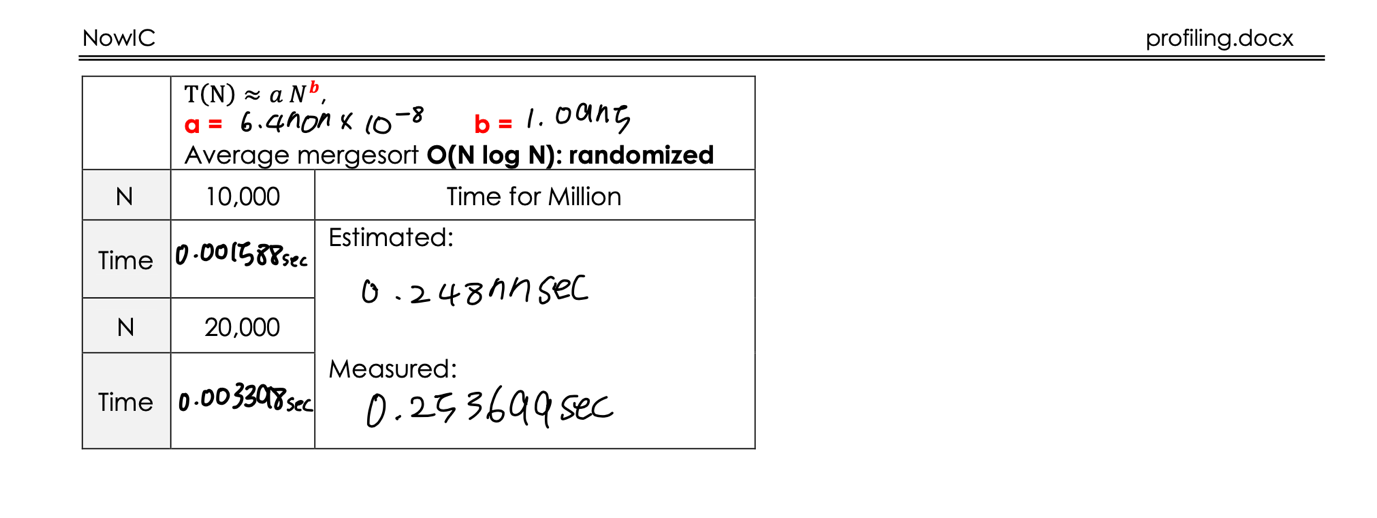


Mergesort – Average



테이블이(가) 표시된 사진

자동 생성된 설명



1. Include the excel chart and graph for comparing those **five cases.**

테이블이(가) 표시된 사진

자동 생성된 설명

1. **Describe your observation** of the time complexity of the five sorting algorithm cases.

Insertionsort에서 시간 복잡도 수치가 가장 작은 순은 sort < randomized < reversed 순이 였다. 이미 sort가 되어있는 값들이 시간 best였고 reversed는 가장 오랜 시간이 걸렸다. 위의 그래프를 보면 quicksort와 mergesort의 평균값은 insertionsort의 best값을 나타내는 그래프와 거의 유사한 점을 보아 insertionsort가 시간 복잡도 수치가 가장 높고 값을 계산하는데 오랜 시간이 걸린다는 것을 알 수 있다. 더불어, 위의 그래프 상에서는 quicksort와 mergesort의 평균값은 insertionsort의 best값을 나타내는 그래프와 거의 유사한 것처럼 보이지만 계산해야하는 값이 커질수록 시간의 차이는 커진다는 것을 1000,000을 계산하는데 걸리는 시간을 보면 알 수 있다.