

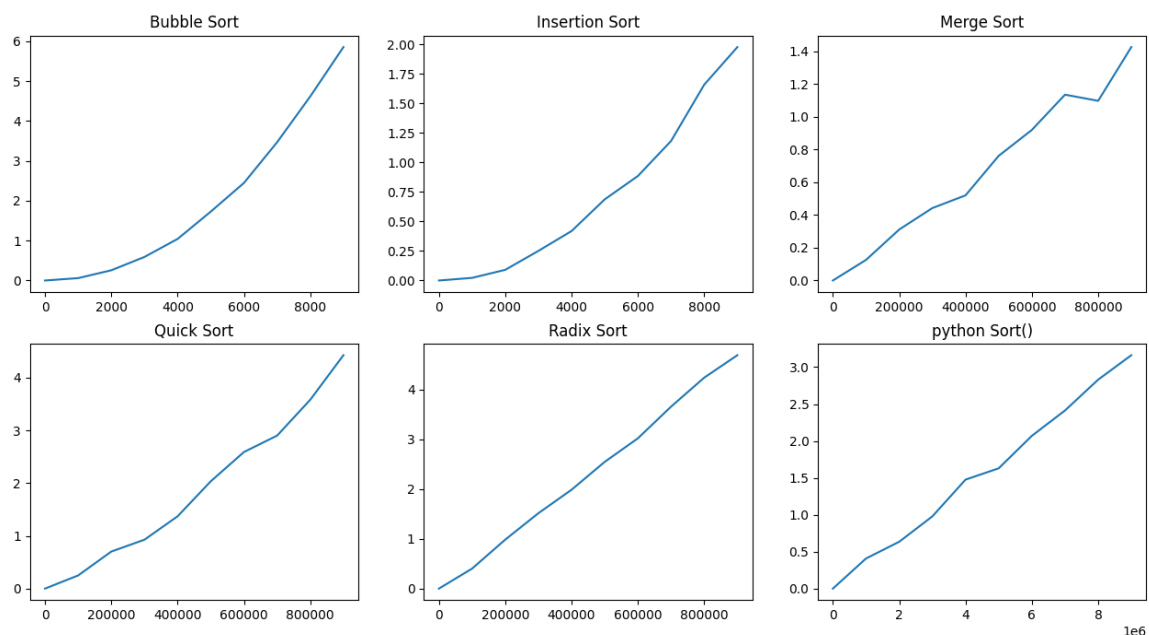
HW 3 성능확인 보고서

2021-27764 안지수
(spds078)

실습환경 : Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz

메모리 : 48GB

파이썬 버전: 3.6.9



Bubble Sort

time complexity : $O(N^2)$

위 그래프를 볼 때 time complexity에 맞추어 증가함을 확인할 수 있다.

Insertion Sort

time complexity :

average: $O(N \log N)$

worst : $O(N^2)$

위 그래프를 볼 때 time complexity에 맞추어 증가함을 확인할 수 있다.

Merge Sort, Quick Sort

time complexity : $O(N \cdot \log N)$

다음 그래프를 볼 때, $O(N \cdot \log N)$ 과 조금 다른 양상을 확인할 수 있다. 이렇게 관찰되는 까닭은 여러가지 요인이 있다고 분석된다. 그 하드웨어 적인 요인으로는 컴퓨터에 실행중인 Process로 잦은 context switching인한 성능저하, cpu cache에서 발생하는 fault로 인한 성능저하 (메모리의 크기는 48기가로 넉넉하기 때문에 virtual memory에 의한 성능 저하는 발생하지 않을 것이라고 예상된다) 등이 있을 것이다. 소프트웨어 적인 요인으로는 random하게 생성된 리스트의 모양에 따라 걸리는 시간의 차이가 있을 것이라고 판단된다.

Radix Sort

time complexity : $O(N \cdot D)$

다음 그래프를 볼 때, $O(N \cdot D)$ 과 조금 다른 양상을 확인할 수 있다. 이렇게 관찰되는 까닭은 여러가지 요인이 있다고 분석된다. 그 하드웨어 적인 요인으로는 컴퓨터에 실행중인 Process로 잦은 context switching인한 성능저하, cpu cache에서 발생하는 fault로 인한 성능저하 (메모리의 크기는 48기가로 넉넉하기 때문에 virtual memory에 의한 성능 저하는 발생하지 않을 것이라고 예상된다) 등이 있을 것이다. 소프트웨어 적인 요인으로는 random하게 생성된 리스트의 모양에 따라 걸리는 시간의 차이가 있을 것이라고 판단된다.

Python Sorted(timesort)

time complexity : $O(N \cdot \log N)$

다음 그래프를 볼 때, $O(N \cdot \log N)$ 과 조금 다른 양상을 확인할 수 있다. 이렇게 관찰되는 까닭은 여러가지 요인이 있다고 분석된다. 그 하드웨어 적인 요인으로는 컴퓨터에 실행중인 Process로 잦은 context switching인한 성능저하, cpu cache에서 발생하는 fault로 인한 성능저하 (메모리의 크기는 48기가로 넉넉하기 때문에 virtual memory에 의한 성능 저하는 발생하지 않을 것이라고 예상된다) 등이 있을 것이다. 소프트웨어 적인 요인으로는 random하게 생성된 리스트의 모양에 따라 걸리는 시간의 차이가 있을 것이라고 판단된다.

이 실험을 통해 분석된 결과로 정렬 알고리즘의 성능은

파이썬 내장 Sorted() > Merge Sort > Quick Sort == Radix Sort >>> Insertion Sort > Bubble Sort

임을 확인할 수 있었다.

또한, 이 실험에서 주목해야 할 점은 $O(N*N)$ 과 $O(N*\log N)$ 사이의 차이가 극명하다는 것이라고 판단 하였다.