

정렬 방식에 따른 BigOh 분석

정 지 호

I. BubbleSort

II. SelectionSort

III. InsertSort

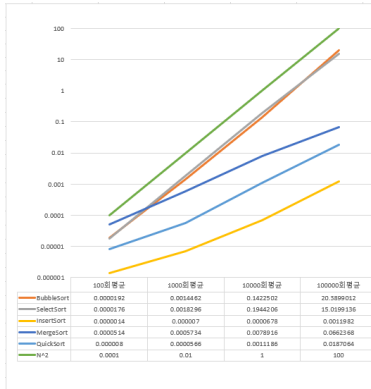
IV. MergeSort

V. QuickSort

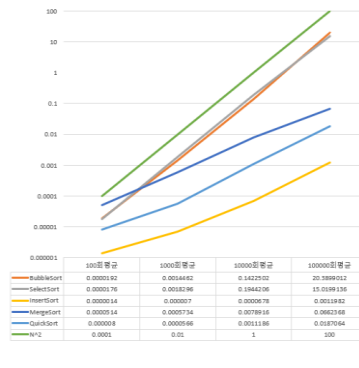
BubbleSort 분석

- 버블 정렬의 비교횟수

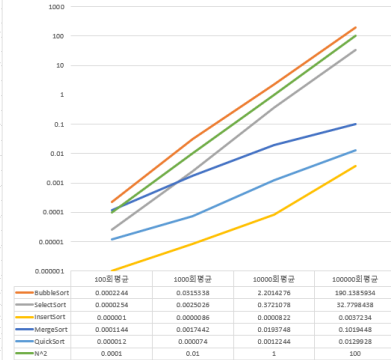
최선, 최악, 평균 모두 시간 복잡도가 일정하다. 무조건 자료의 개수의 제곱만큼 검사 하기 때문에 모두 시간 복잡도 $O(n^2)$ 가진다.



-랜덤데이터-



-정렬데이터-



-역 정렬데이터-

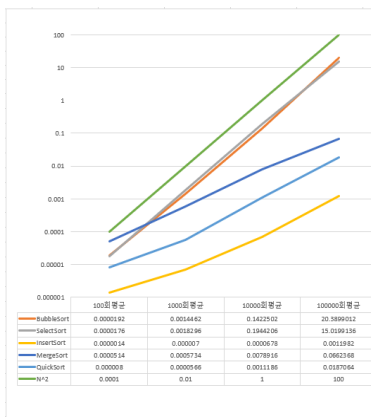
*버블정렬 그래프, * N^2 그래프

위의 그래프와 같이 랜덤, 정렬데이터, 역 정렬 데이터 모두 N^2 의 그래프와 비슷하게 비례하는 것을 볼 수 있다.

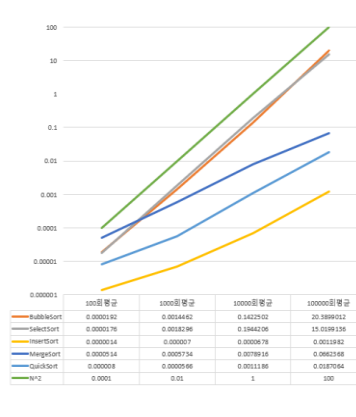
SelectionSort 분석

- 선택 정렬의 비교횟수

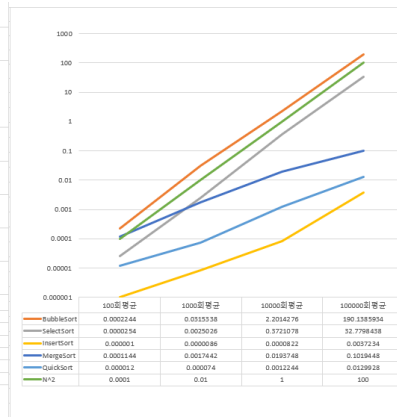
선택 정렬은 두개의 루프문을 이용해 실행해 최선, 최악, 평균 모두 상관없이 시간복잡도 $O(n^2)$ 을 가진다.



-랜덤데이터-



-정렬데이터-



-역 정렬데이터-

*선택 정렬 그래프, * N^2 그래프

위 그래프와 같이 랜덤데이터, 정렬 데이터, 역 정렬 데이터 어떤 것이 들어와도 N^2 그래프가 비례하는 것을 볼 수 있다.

InsertionSort 분석

- 삽입 정렬의 비교횟수

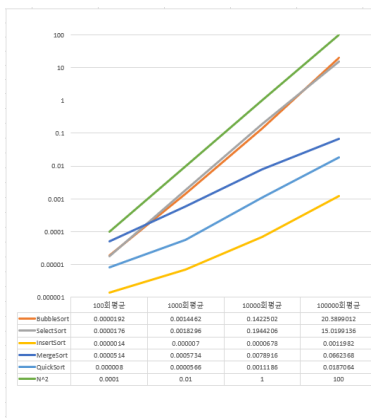
최선의 경우 이동없이 1 번의 비교만 이루어진다.

- $(n-1)번 = O(n)$

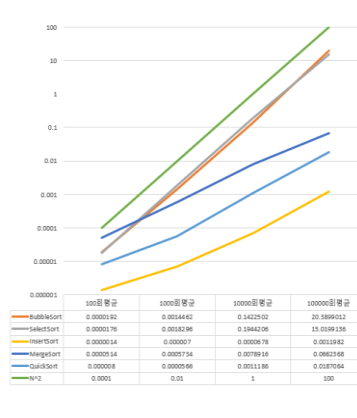
최악의 경우 외부 반복문 안의 각 반복마다 i 번의 비교 수행이 이루어짐.

→ $(n-1) + (n-2) \dots = O(n^2)$

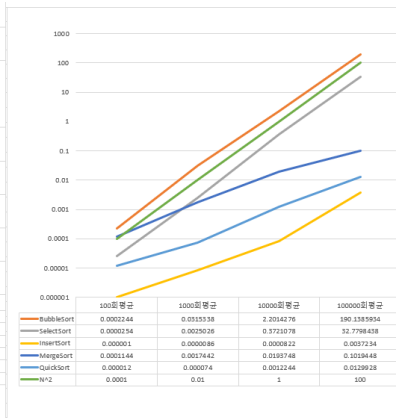
평균적으로는 $O(n^2)$ 정도의 시간 복잡도를 가진다.



-랜덤 데이터-



-정렬 데이터-



-역 정렬데이터-

이 정렬은 최악과 최선이 확실하게 구분 되어있는데, 최선의경우는 정렬된 데이터, 최악의 경우는 역으로 정렬된 데이터이다. 2 번째 그래프를 보면 정렬된 데이터가 들어 왔을 때, 어떠한 정렬 방식보다 빠른 정렬 속도를 보여준다. 3 번째 그래프는 위의 시간 복잡도와 같이 N^2 의 복잡도를 가지고 랜덤데이터 역시 N^2 의 시간 복잡도를 갖는 것을 알 수 있다.

MergeSort 분석

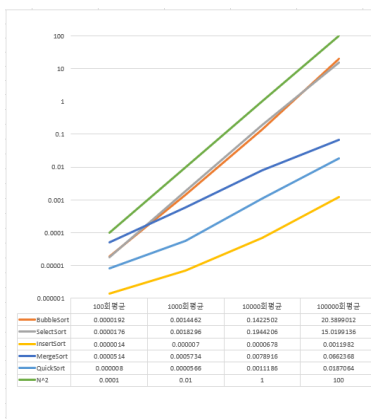
- 합병 정렬의 비교횟수

합병 정렬은 어느 상황에서도 분할하고 다시 합치는 과정을 한다.

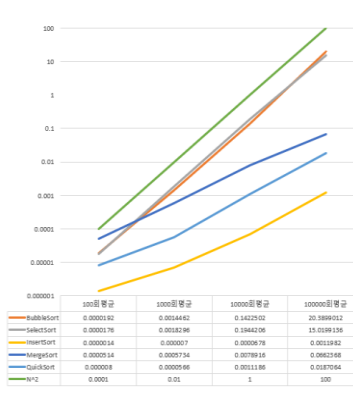
호출하여 분할하는 단계 = $\log_2 N$;

하나의 합병단계에서는 최대 n 번의 비교연산을 수행한다.

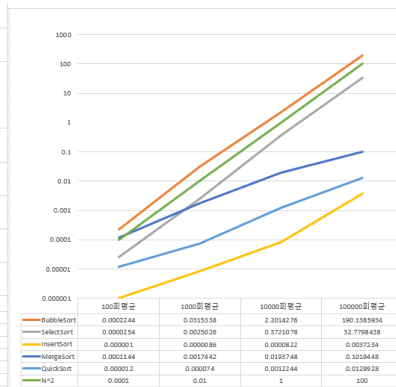
$$\log_2 n * n = n \log_2 n$$



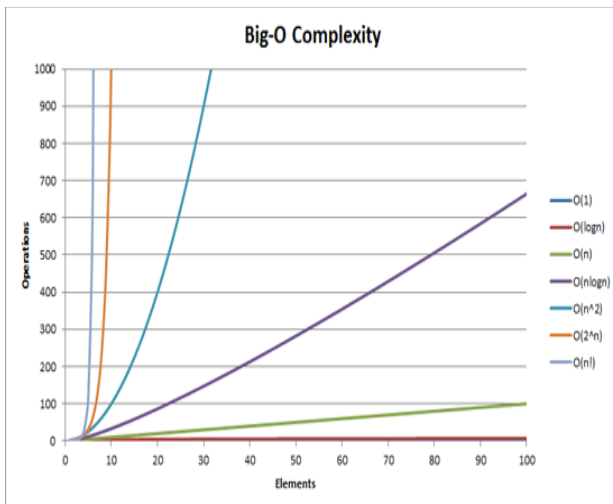
-랜덤 데이터-



-정렬 데이터-



-역 정렬 데이터-



*왼쪽 그래프는 시간 복잡도 그래프이다

위의 랜덤데이터, 정렬 데이터, 역 정렬 데이터를 넣었을 때의 그래프를 보면 시간 복잡도 그래프의 $n \log n$ 의 그래프와 매우 유사한 것을 볼 수있다.

QuickSort 분석

- 퀵 정렬의 비교횟수

최선의 경우 합병 정렬과 같이 분할하는 과정.

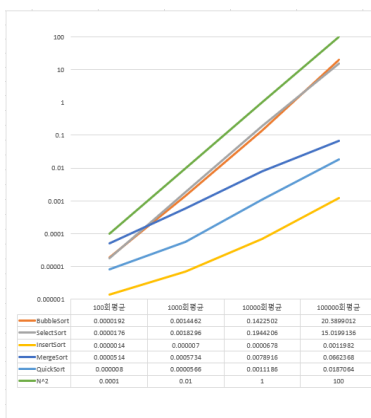
$\log_2 n$.

또한 호출하는 단계에서 대부분의 배열을 비교해야 하므로 평균 n 번정도의 비교가 이루어진다. 그러므로 $n * \log_2 n = n \log_2 n$

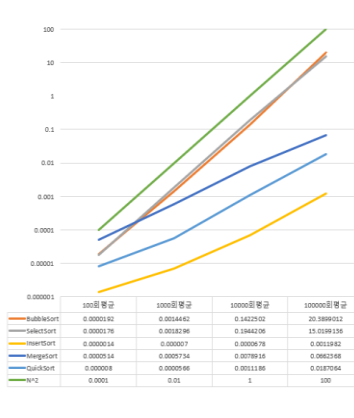
최악의경우는 이미 정렬된 데이터가 들어왔을 양쪽이 불균형 하게 분할이 되는 경우이다. 이 경우에는 n 번의 분할이 이루어진다.

호출해 비교하는 단계는 위와 같으므로 결국 $n * n = n^2$ 의 시간 복잡도를 가짐.

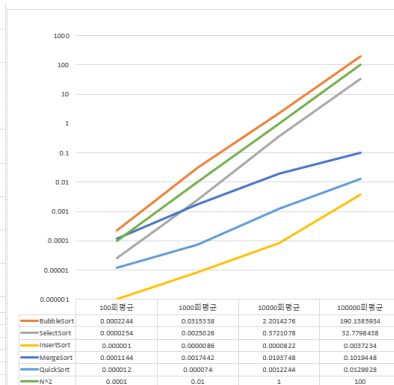
평균은 $n \log_2 n$ 의 시간 복잡도.



-랜덤 데이터-



-정렬 데이터-



-역 정렬 데이터-

위에 그래프와 같이 랜덤 데이터를 보았을 때 $n \log_2 n$ 의 시간 복잡도를 가지는 것을 볼 수 있다. 하지만 위에 언급한 것과는 다르게 정렬데이터가 n^2 의 시간 복잡도를 가지지 않는 것을 볼 수 있다. 그 이유는 최대한의 효율을 내기위해 피벗을 첫 값이 아닌 중간 값으로 지정 했기 때문에 최악의 경우도 n^2 이 아니게 된 것 이다.