

# 삽입 정렬, 선택 정렬, 버블 정렬, 퀵 정렬의 성능 측정 및 비교

김태지

중원대학교 컴퓨터공학과

## *Measure and compare the performance of insertion sort, selection sort, bubble sort, and quick sort algorithm*

*Tae-Ji Kim*

*Department of Computer Engineering, Jungwon University*

### 초록

정렬 알고리즘은 데이터가 대량으로 저장된 데이터베이스에서 이진 탐색 알고리즘을 사용해 데이터를 조회하는 성능을 높이기 위해 사용한다. 정렬 알고리즘의 성능평가는 시간 복잡도를 계산하는 방법과, 직접 여러 형태의 분포를 갖는 데이터를 만들고 각 정렬 알고리즘이 데이터를 정렬하는데 걸리는 시간을 측정해서 실험하는 방법이 있다. 실험 결과 퀵 정렬 알고리즘이 구현이 복잡한 단점이 있지만 가장 좋은 성능을 보여주었다. 삽입 정렬 알고리즘과 버블 정렬 알고리즘은 대체적으로 선택 정렬 알고리즘 보다 좋지 않은 성능을 보여주었는데, 특히 데이터가 역으로 정렬된 경우 최악의 성능을 보여주었다. 따라서 데이터의 양이 많고, 정렬의 속도가 중요한 작업을 하는 경우 퀵 정렬 알고리즘을 사용하는 것이 가장 효율적이고, 반대의 경우 구현이 비교적 간단한 정렬 알고리즘인 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 버블 정렬 알고리즘을 사용하는 것이 적합하다.

주제어: 정렬 알고리즘, 탐색, 선택 정렬, 삽입 정렬, 버블 정렬, 퀵 정렬

## 1. 서론

컴퓨터 분야가 발전하고 정보공학 방법론이 개발된 이후, 컴퓨터는 단순한 연산의 도구가 아닌 데이터를 저장하고 처리하는 데이터베이스의 용도로도 사용되게 되었다. 데이터베이스에 저장되는 데이터의 크기가 커짐에 따라 이미 데이터베이스에 저장된 대량의 데이터 중에서 원하는 데이터를 탐색하는 속도의 중요성이 부각되었다.

데이터를 탐색의 성능에는 데이터의 정렬 여부가 중요하다. 데이터가 정렬이 되어 있지 않으면 데이터를 탐색할 때 순차 탐색 알고리즘 밖에 사용을 할 수 밖에 없기 때문이다. 데이터가 정렬이 되어 있는 경우 이진 탐색이라는 강력한 탐색 알고리즘을 사용할 수 있다. 이진 탐색 알고리즘은  $O(\log_2 n)$ 의 시간 복잡도를 갖고, 순차 탐색 알고리즘은  $O(n)$ 의 시간 복잡도를 갖는다. 이진 탐색 알고리즘의 사용 여부는 데이터베이스의 데이터가 커질 수록 데이터 조회 기능의 성능에 큰 차이를 만들어낸다.

본 논문은 데이터를 효율적으로 정렬하기 위해 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 버블 정렬 알고리즘, 퀵 정렬 알고리즘의 성능을 비교하기 위해 정렬이 되지 않은 데이터, 정렬이 되어 있는 데이터, 역으로 정렬된 데이터, 부분적으로 정렬된 데이터를 만든 후 각 정렬 알고리즘이 각각의 데이터를 정렬하는데 걸리는 시간을 측정해서 각 정렬 알고리즘의 성능을 측정하고 비교하고자 한다.

## 2. 연구 방법

프로그래밍 언어는 Python을 사용하였고, 개발 환경은 PyCharm을 사용하였다. Python의 random 함수를 사용해서 크기가 0부터 1000까지 10씩 증가하는 정렬이 되어있는 데이터, 역으로 정렬이 되어있는 데이터, 정렬이 되어있지 않은 데이터를 만들었고 크기가 1000이고 부분적으로 정렬이 되어있는 데이터를 만들었다. 만든 데이터를 기준으로 삽입 정렬, 선택 정렬, 버블 정렬, 퀵 정렬 알고리즘이 데이터를 정렬하는 데 걸리는 시간을 측정한 후, 측정 결과를 Python의 open() 함수를 사용해서 csv파일로 저장했다. 저장된 데이터를 Microsoft Excel을 사용해서 선형 그래프와 막대 그래프로 성능을 비교하였다.

### 3. 실험 결과

#### 1. 선택 정렬 알고리즘의 성능 평가

선택 정렬 알고리즘은 첫 번째 데이터부터 시작해서 현재 데이터와 현재 이후의 데이터부터 마지막까지의 데이터 중에서 가장 작은 값을 찾고 이 값과 현재 데이터를 교체하는 것을 반복하는 정렬 알고리즘이다.

선택 정렬 알고리즘의 시간 복잡도는 최고, 최저, 평균 모두  $O(n^2)$  이며, 정렬 속도는 정렬이 되어 있는 데이터, 정렬이 되어 있지 않은 데이터, 역으로 정렬된 데이터 순으로 빨랐으나 결과적으로 유의미한 성능차이를 보이지는 않았다.

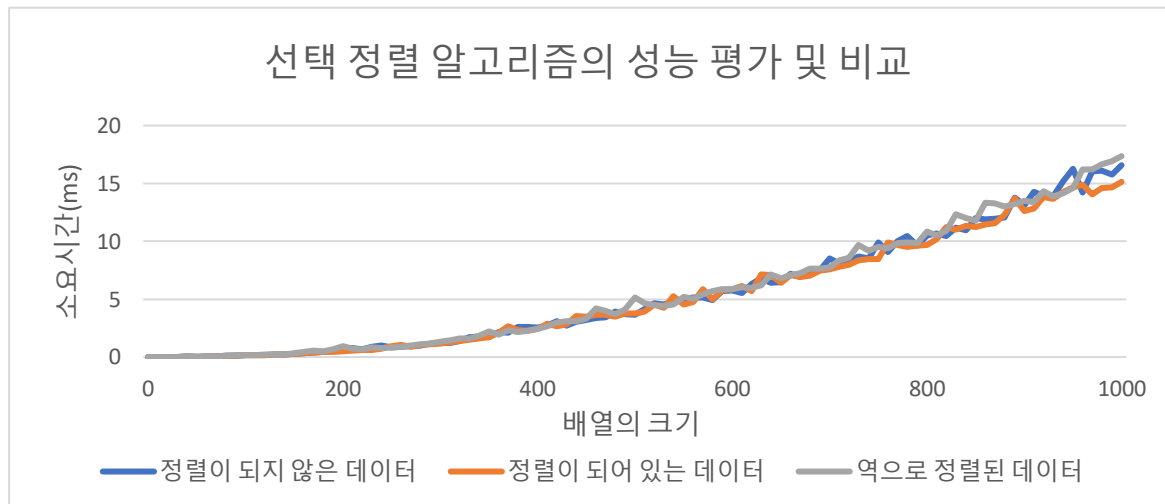


Figure 1. 선택 정렬 알고리즘의 성능비교

#### 2. 삽입 정렬 알고리즘의 성능 평가

삽입 정렬은 두 번째 데이터부터 시작하며, 현재 데이터와 현재 데이터보다 작은 인덱스를 가진 데이터들을 비교하면서 정렬이 되어있는 데이터에 현재 데이터의 삽입 위치를 찾는다. 즉 자료 배열의 모든 요소를 앞에서부터 차례대로 이미 정렬된 배열 부분과 비교하여 데이터가 들어갈 위치를 찾아 삽입하는 정렬 알고리즘이다.

삽입 정렬 알고리즘의 시간 복잡도는 최고  $O(n)$ , 평균  $O(n^2)$ , 최악  $O(n^2)$ 의 시간 복잡도를 갖으며 정렬 속도는 정렬이 되어있는 데이터, 정렬이 되지 않은 데이터, 역으로 정렬된 데이터 순으로 빨랐다.

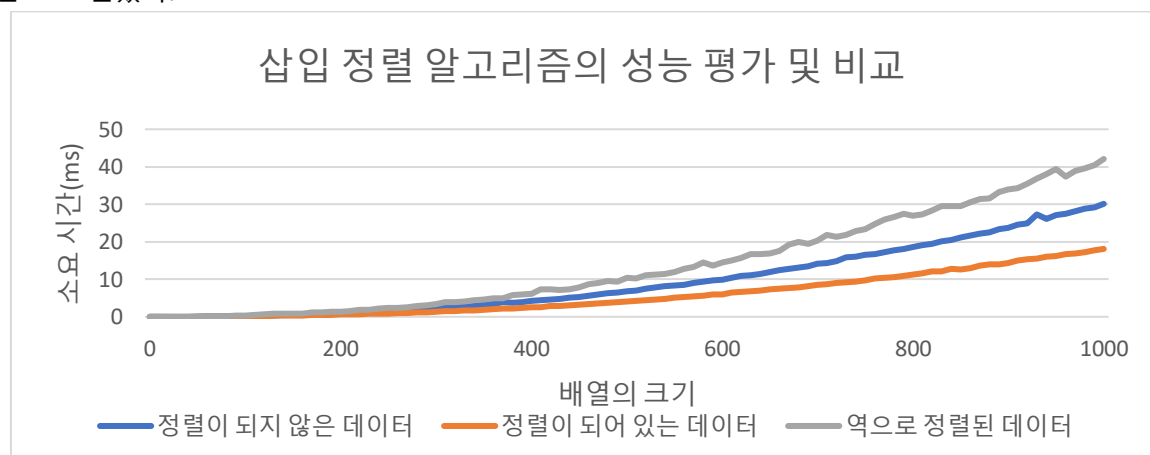


Figure 2. 삽입 정렬 알고리즘의 성능 비교

### 3. 버블 정렬 알고리즘의 성능 평가

버블 정렬 알고리즘은 인덱스를 하나씩 증가시키며 현재 인덱스의 데이터와 현재 인덱스보다 값이 1큰 데이터를 비교하고 작은 값을 현재 인덱스에 위치시키는 작업을 반복하면서 가장 큰 데이터가 마지막 인덱스에 위치하도록 하는 작업을 반복하는 정렬 알고리즘이다.

버블 정렬 알고리즘의 시간 복잡도는 최고, 최저, 평균 모두  $O(n^2)$ 이며, 정렬 속도는 정렬이 되어 있는 데이터, 정렬이 되지 않은 데이터, 역으로 정렬된 데이터 순으로 빨랐다.

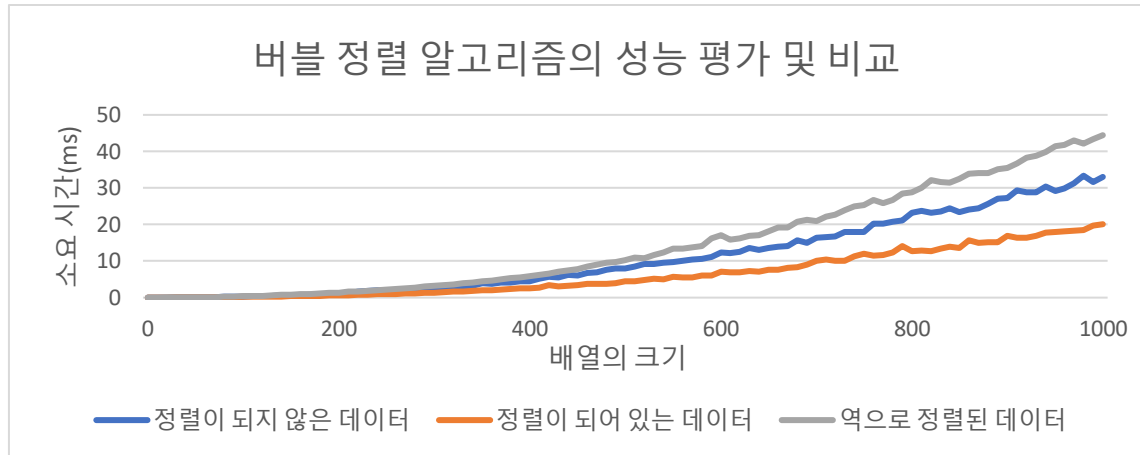


Figure 3. 버블 정렬 알고리즘의 성능 평가

### 4. 퀵 정렬 알고리즘의 성능 평가

퀵 정렬 알고리즘은 분할 정복 알고리즘 중 하나로, 배열을 pivot이라는 원소를 기준으로 pivot보다 작은 원소들은 pivot원소의 인덱스보다 작은 인덱스에 위치하도록 하고, pivot보다 큰 원소들은 pivot원소의 인덱스보다 큰 인덱스를 가지도록 한 후, pivot을 기준으로 배열을 pivot보다 작은 원소들과 pivot보다 큰 원소들로 나눈다. 나뉘어진 배열에서 다시 pivot을 정하고 정한 pivot을 기준으로 pivot보다 큰 원소들과 pivot보다 큰 원소들을 규칙에 맞게 배치시킨 후 pivot을 기준으로 배열을 나누는 과정을 반복하면서 데이터를 정렬한다.

퀵 정렬 알고리즘의 시간 복잡도는 최고  $O(n \log_2 n)$ , 평균  $O(n \log_2 n)$ , 최악  $O(n^2)$ 의 시간 복잡도를 갖는다. 퀵 정렬 알고리즘은 pivot에 의해 배열이 지속적으로 불균형하게 나뉘어지는 경우 최악  $O(n^2)$ 의 시간 복잡도를 갖게 되는 특징이 있다. 정렬 속도는 정렬이 되지 않은 데이터가 가장 느린 결과를 보여주었고, 정렬이 되어있는 데이터와 역으로 정렬되어 있는 데이터에서는 유의미한 성능 차이를 보이지 않았다.

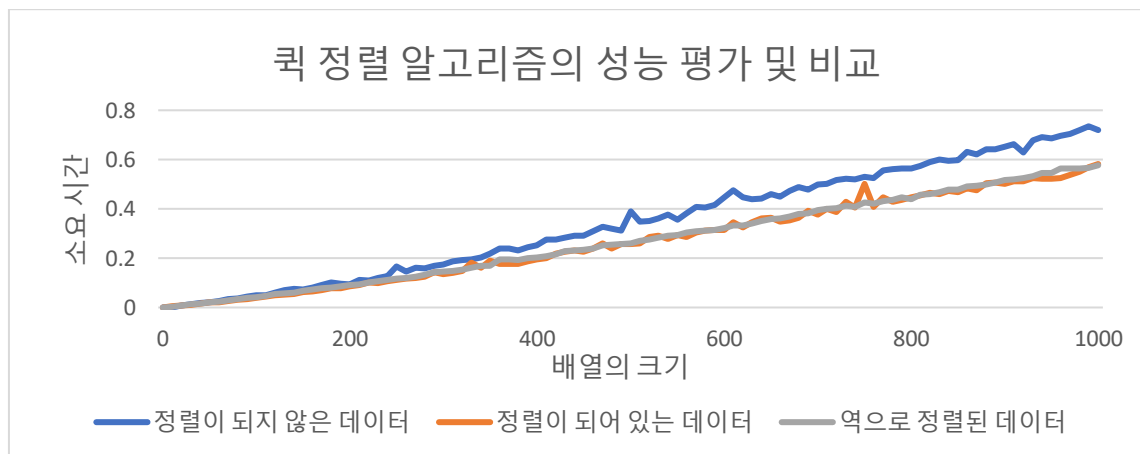


Figure 4. 퀵 정렬 알고리즘의 성능 평가 및 비교

## 5. 종합적인 성능 비교

정렬이 되어 있지 않은 데이터에서는 퀵 정렬 알고리즘, 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 버블 정렬 알고리즘 순으로 빨랐다.

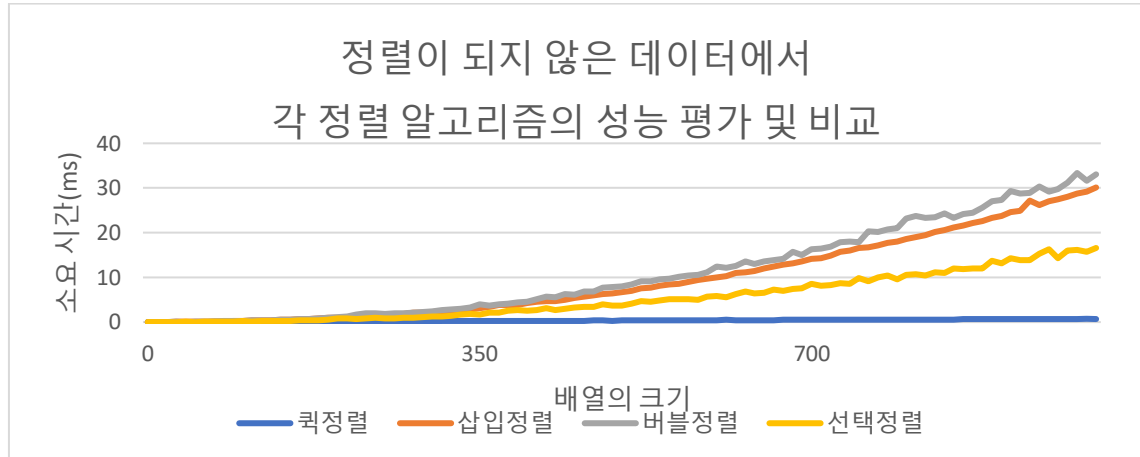


Figure 5 정렬이 되어 있지 않은 데이터에서 각 정렬 알고리즘의 성능 평가 및 비교

정렬이 되어 있는 데이터에서는 퀵 정렬 알고리즘, 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 버블 정렬 알고리즘 순으로 빨랐다.

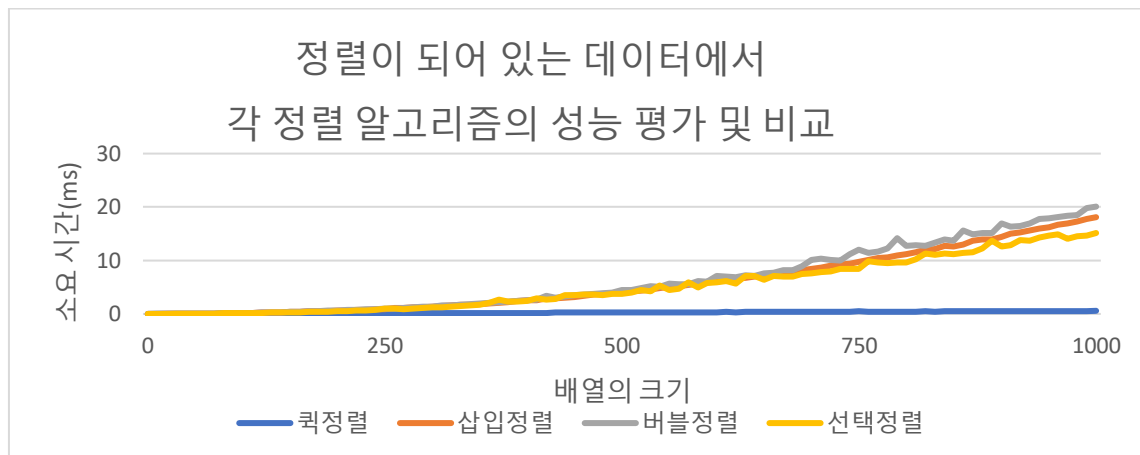


Figure 6. 정렬이 되어 있는 데이터에서 각 정렬 알고리즘의 성능 평가 및 비교

역으로 정렬된 데이터에서는 퀵 정렬 알고리즘, 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 버블 정렬 알고리즘 순으로 빨랐다.

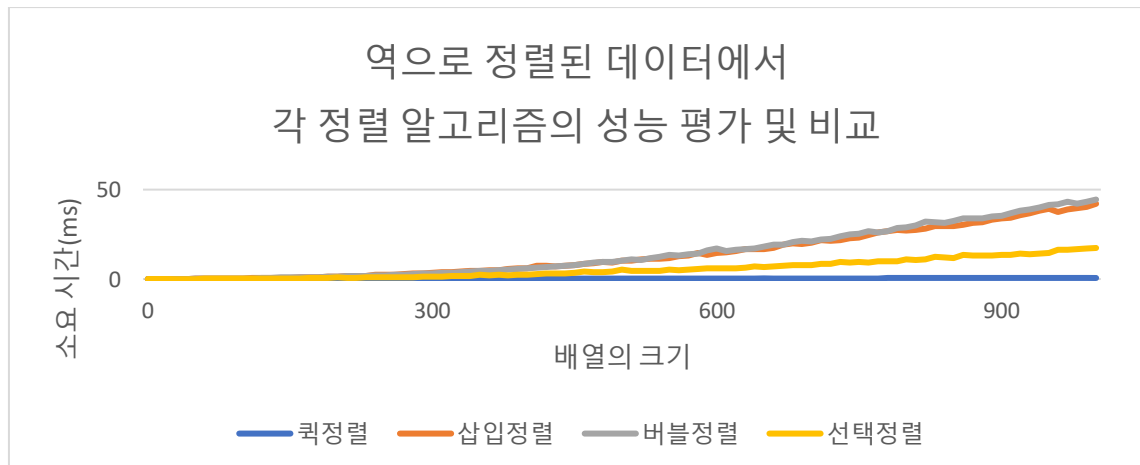


Figure 7. 역으로 정렬된 데이터에서 각 정렬 알고리즘의 성능 평가 및 비교

배열의 크기가 1000이고 연속적이지만 100개씩 정렬이 되어있지 않은 데이터에서는 퀵 정렬 알고리즘, 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 버블 정렬 알고리즘 순으로 정렬 속도가 빨랐다.

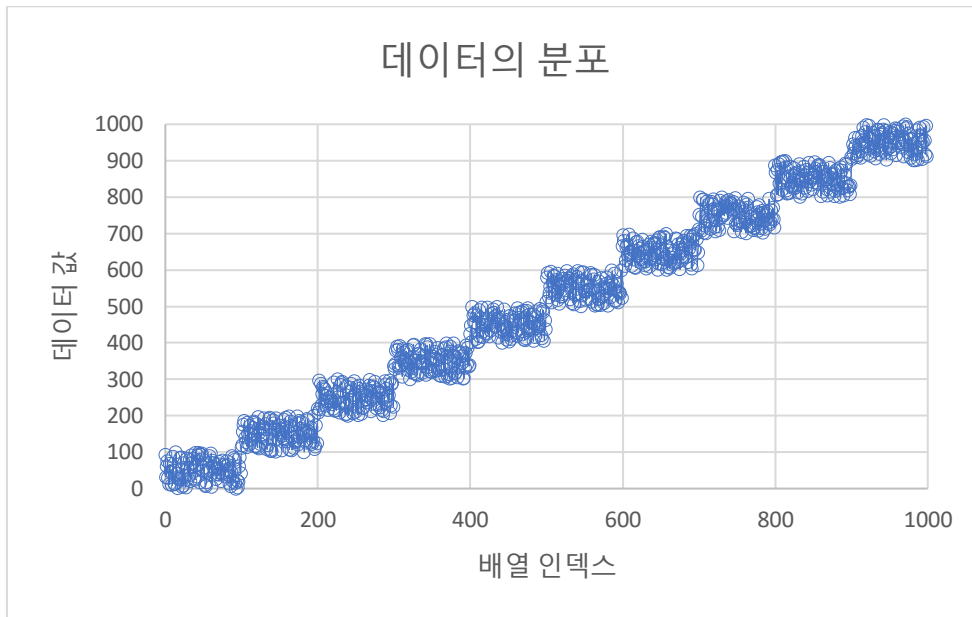


Figure 8. 부분적으로 정렬된 데이터의 분포

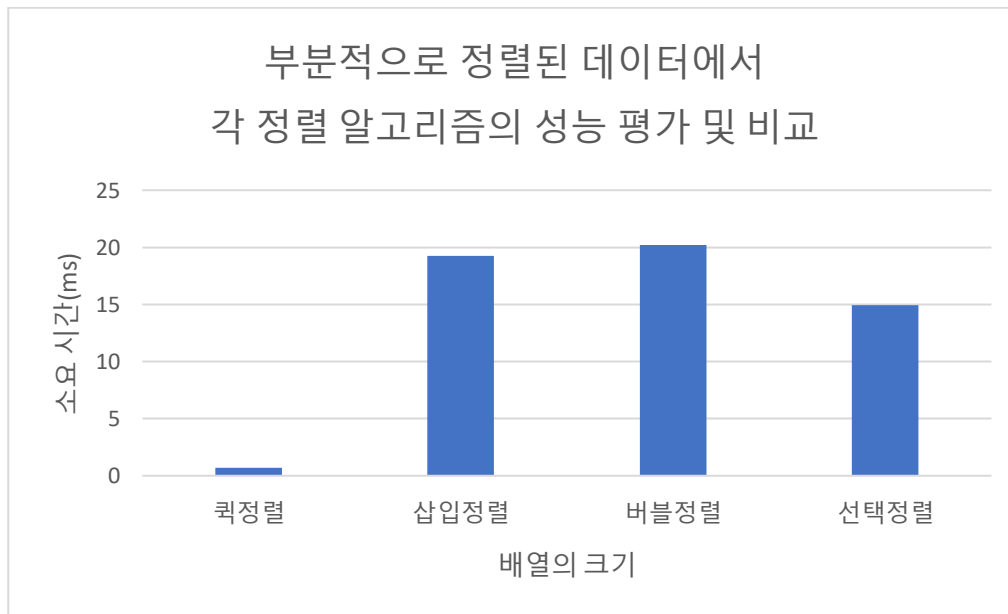


Figure 9. 부분적으로 정렬된 데이터에서 각 정렬 알고리즘의 성능 평가 및 비교

## 4. 결론

본 논문에서 조사한 정렬 알고리즘의 성능은 퀵 정렬 알고리즘, 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 버블 정렬 알고리즘 순으로 좋은 성능을 보여주었다..

선택 정렬 알고리즘은 배열의 데이터의 정렬 형태가 성능에 큰 영향을 미치지 않았다.

삽입 정렬 알고리즘과 버블 정렬 알고리즘은 배열이 정렬이 되어 있는 경우, 정렬이 되지 않은 경우, 역으로 정렬되어 있는 경우 순서로 정렬 소요 시간이 적게 들었다. 따라서 데이터가 어떻게 분포가 되어 있는가에 따라 성능에 영향이 있다.

퀵 정렬 알고리즘은 배열이 정렬이 되어 있지 않은 경우가 배열이 정렬이 되어 있는 경우와 배열이 역으로 정렬되어 있는 경우보다 정렬 소요 시간이 길었다. 따라서 퀵 정렬은 정렬이 되어있는 데이터와 역으로 정렬된 경우의 데이터의 경우 비슷한 성능을 보여주지만 데이터의 분포가 정렬이 되어 있지 않은 경우 정렬 소요 시간이 길어질 수 있다. 퀵 정렬은 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 버블 정렬 알고리즘과 달리, 정렬해야 할 데이터의 크기가 커지면 커질수록 좋은 성능을 보여주었다.