

## 1. 과제 제목

정렬 알고리즘

## 2. 과제 목표

삽입 정렬, 버블 정렬, 선택 정렬을 어셈블리를 이용하여 구현한다. 여러 개의 16진수로 나타낸 수를 32-bit floating point로 저장하고, 각 정렬알고리즘을 사용하여 내림차순으로 정렬하는 작업을 한다. 각 정렬 알고리즘의 동작을 확인하고 성능비교를 진행한다.

## 3. 일정

	11/14	15	16	17	18	19	20	21
설계								
삽입정렬								
버블정렬								
선택정렬								
테스트								
보고서								

프로젝트 기간은 11/14부터 11/21까지 7일로 설정하였다. 전체 프로젝트에 대한 설계를 끝낸 뒤 삽입 정렬, 버블 정렬, 선택 정렬을 순서대로 구현하고, 테스트한 뒤 피드백을 진행한다. 보고서 작성은 프로젝트 구현이 종료된 후에 진행한다.

## 4. 각 function 별 알고리즘

## (1) 삽입 정렬

삽입 정렬은 나열된 자료를 맨 앞에서부터 하나씩 비교하여 삽입하고, 삽입지점 이후의 자료를 뒤로 미루는 과정을 반복하여 정렬하는 알고리즘이다. 이미 정렬되어 있는 자료에 새로운 항목을 삽입하는 경우에 가장 적은 연산을 한다. 반면 자료가 역순서로 정렬되어 있는 경우 처음부터 뒤쪽의 자료까지 모든 자료를 비교하게 되므로 가장 많은 연산이 필요하다.

## (2) 버블 정렬

버블 정렬은 나열된 자료에서 연속적인 위치의 값을 서로 비교하여 정렬순서에 맞도록 적절하게 교환한다. 마지막 자료까지 교환하면 가장 마지막 자료가 정렬된다. 이 과정을 반복하면 자료가 뒤에서부터 정렬된다. 이미 정렬되어 있는 자료에 새로운 항목을 삽입하는 경우에 가장 적은 연산을 하고, 자료가 역순서로 정렬되어 있을 때 가장 많은 연산을 한다.

## (3) 선택 정렬

선택 정렬은 앞에서부터 자료를 선택하여 선택한 자료 뒤에 남아있는 자료 중 정렬 순서에 맞는 것과 교환한다. 선택 정렬은 이미 정렬되어 있거나 역순서로 정렬되어 있는 경우에도 수행하는 연산의 수가 동일하다.

#### (4) floating point 변환

16진수 값을 floating point로 변환하는 과정은 16진수 값을 2진수로 변환한 뒤, 2진수 값을 floating point로 변환하는 것이다. 16진수를 2진수로 변환하면 1자리의 16진수 값이 4자리의 2진수 값으로 변환된다. 16진수 값을 2로 나눈 나머지를 나열하는 것으로 변환할 수 있다. 2진수를 floating point로 변환하는 과정은 2진수를  $1.xxx \times 2^{\text{지수}}$ 의 형태의 형태로 변형하여 1.xxx의 소수부분과 2의 지수를 저장하는 것이다. IEEE 754에서는 소수부분인 Mantissa에 23bit, 지수부분인 Exponent에 8bit를 사용한다. 이때 Exponent는 2의 지수에 127을 더한 값이다.

### 5. 예상되는 문제점, 실행시간 예상

C언어와 파이썬 등에서 이용한 경험이 있는 익숙한 정렬 알고리즘이지만, 각 단계에 대한 명령어를 세부적으로 작성해야 하는 어셈블리어언어이기 때문에 원하는 동작을 코드로 구현하는 것에 어려움을 겪을 수 있을 것 같다.

각 알고리즘의 실행시간에 대한 예측은 데이터가 충분히 무작위적이라고 가정했을 때, 버블 정렬이 가장 느리고, 선택 정렬이 가장 빠를 것이라고 예상한다. 버블 정렬은 삽입 정렬, 선택 정렬과 마찬가지로 하나의 요소에 대해 전체 자료를 살펴보지만, 비교만 실행하는 삽입 정렬과 달리 비교의 결과에 따라 값을 교환하는 부분이 반복적으로 나타나기 때문이다. 반면 선택 정렬은 값을 삽입하기 할 때 두 개의 값을 교환하는 방법을 사용하기 때문에, 값을 삽입하기 위해 모든 값을 한 칸씩 미루는 삽입 정렬보다 속도가 더 빠를 것이라고 생각한다.

### 6. 검증 전략

구현한 각 알고리즘에 대해 data.txt로 제시된 값을 이용하여 알고리즘 별 정렬에 소요된 state를 확인하고 비교한다. 이후 이미 정렬된 자료에 특정 값을 새로 삽입하여 정렬하는 명령을 알고리즘별로 추가 실행하여 소요된 state를 확인하고 비교한다.