

# Programming Methodology

## Instructor: Kyuseok Shim

### Project #1: Sorting & Selection

Due Date: 0:0 a.m. between 2007-10-2 & 2007-10-3

#### Introduction

Sorting(정렬):

주어진 데이터의 순서를 정해진 규칙에 따라 재배치한다.

예를들어 27, 83, 11, 74, 59의 숫자를 오름차순으로 정렬한다면 11, 27, 59, 74, 83이 된다

Selection :

주어진 데이터에서 정해진 규칙에 따른 k번째 값을 구한다.

예를 들어 27, 83, 11, 74, 59의 숫자 중 2번째로 작은 값을 구한다면 27이 된다

가장 손쉬운 방법으로는 정렬을 한 후 k번째 값을 찾는 법이 있다.

#### Implementation

sorting algorithm.

##### 1. Insertion sort

- 함수의 이름은 반드시 insertion-sort으로 한다
- 오름차순, 내림차순을 선택할 수 있도록 한다

##### 2. Merge sort

- 함수의 이름은 반드시 merge-sort으로 한다
- 오름차순, 내림차순을 선택할 수 있도록 한다

##### 3. Quick sort

- 함수의 이름은 반드시 quick-sort으로 한다
- 오름차순, 내림차순을 선택할 수 있도록 한다

selection algorithm

##### 1. Quick select

- 함수의 이름은 반드시 quick-selection으로 한다
- k번째 작은 값이나, k번째 큰 값을 선택하도록 구현한다

\*모든 구현은 수업시간에 배운 문법만을 사용하도록 한다

\*사용자 임의로 함수를 만드는 것은 가능하다

## Input

### 1. sort

정렬할 데이터는 `List`로 주어지고 오름차순은 `<`, 내림차순은 `>`으로 주어진다

예를 들어, 27, 83, 11, 74, 59의 숫자를 정렬하고 싶다면 다음과 같은 데이터를 입력한다  
(27 83 11 74 59)

물론 함수를 호출할 때는 다음과 같이 해야 할 것이다(오름차순 `<`)

```
(merge-sort '(27 83 11 74 59) <)
```

혹은

```
(define x '(27 83 11 74 59))
```

```
(quick-sort x <)
```

### 2. select

k-번째 값을 선택할 데이터 또한 `list`로 주어지고, k는 숫자로 주어진다

k-번째 작은 값을 선택할 때는 `<`, k-번째 큰 값을 선택할 때는 `>`을 입력한다

예를 들어 27, 83, 11, 74, 59의 숫자 중 2번째로 작은 값을 선택하는 경우는 다음과 같다

```
(quick-selection '(27 83 11 74 59) 2 <)
```

혹은

```
(define x '(27 83 11 74 59))
```

```
(quick-selection x 2 <)
```

## Output

sorting에 대한 결과는 `list`로 selection에 대한 결과는 숫자로 표현한다

```
(merge-sort '(27 83 11 74 59) <)
```

```
>>(11 27 59 74 83)
```

```
(quick-selection '(27 83 11 74 59) 2 <)
```

```
>>27
```

## 참고

3가지 종류의 sorting algorithm과 한가지의 selection algorithm은 게시판에 있는 강의슬라이드의 설명 및 pseudo code를 참조

## 프로그램 실행 예시

```
> (define input '(27 83 11 74 59))

> (insertion-sort input <)
(11 27 59 74 83)

> (merge-sort input <)
(11 27 59 74 83)

> (quick-sort input <)
(11 27 59 74 83)

> (quick-selection input 2 <)
27
```

## 제출사항

### 1. 제출 방법 및 내용

#### 1) 프로젝트 구현 파일 제출

- 파일은 ee 홈페이지에 과제 제출을 이용하여 제출한다.
- 각 제출 파일 이름은 다음의 형식을 따른다.

pro\_1\_학번.scm

예) pro\_1\_2007-12345.scm

- 각 소스 파일 상단에 **E-mail**주소와 작성자 학번 이름을 적는다.

#### 2) 보고서 제출

간단한 설명과 discussion을 하드카피로 제출한다.

### 2. 제출 일시

**파일 제출 마감일 : 2007년 10월 2일과 3일 사이 새벽 0:00**

- 구현 파일 제출 시각은 홈페이지 과제 제출 시각에 준한다

**보고서 제출 마감일 : 2007년 10월 4일 오전 12:00(정오)**

**302동 516-2호 앞 과제제출함**

## 채점 규칙

몇 가지 다양한 숫자(수십개에서 수백,수천개 이상)의 list들을 입력하여 각각의 데이터들이 정확히 정렬되는지 확인하여 채점

딜레이는 하루에 10%씩 감점

Copy 발견 시 모든 숙제 0점 처리