

[511643] 자료구조**실습 #13 보고서**

이름	곽영주
학번	20175105
소속 학과/대학	빅데이터
분반	03 (담당교수: 김태운)

<주의사항>

- 개별 과제입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
- **각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.**
 - 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
- 스마트캠퍼스 제출 데드라인: **2020. 6. 10.~2020. 6. 16. (화요일) 23:59 // 1 주 과제**
 - 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)
 - 주말/휴일/학교행사 등 모든 날짜 카운트 함
 - 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함
 - 예외 없음
- 스마트캠퍼스에 아래의 파일을 제출 해 주세요
 - 보고서(**PDF 파일로 변환 후 제출**)
 - 보고서 파일명에 이름과 학번을 입력 해 주세요.
 - 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (미 제출시 감점)

<개요>

이번 과제는 정렬 알고리즘을 구현하는 내용입니다.

<실습 과제>

[Q 0] 요약 [10 점]

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.

답변: 강의노트와 자료검색, 교수님의 메일 답변을 통해 과제를 완성했습니다. 그리고 과제를 통해 다양한 버블, 선택, 삽입, 합병, 퀵 정렬 알고리즘을 배우고 각각의 정렬 알고리즘의 수행시간을 직접 측정해볼 수 있어서 정렬 알고리즘을 이해하는데 큰 도움이 되었습니다.

[Q 1] 정렬할 데이터 생성하기 [10 점]

N 개의 int 값을 무작위로 생성하고, 생성한 숫자를 배열로 리턴하는 GenData 클래스를 다음과 같이 구현하세요.

- `int[] getRandIntArr(int N, int min, int max) :`
 - N 개의 int 값을 무작위로 생성하여 배열에 저장한 뒤, 배열을 리턴.
 - N은 생성할 int 형 값의 개수를 의미하고, N은 항상 2 보다 크거나 같다.
 - 무작위로 생성한 각각의 값은 min 보다 크거나 같으며, max 보다 작거나 같다.
 - 리턴되는 배열에 저장된 int 값들은 정렬되지 않은 상태이다.
 - 메소드를 호출할 때 리턴되는 배열에 저장되는 값들은 항상 바뀌어야 한다. 즉, `getRandIntArr` 을 연속으로 두 번 호출하고 각각 리턴된 배열을 `arr1`, `arr2` 라고 하면, `arr1` 과 `arr2` 는 서로 같지 않다 (일부 숫자는 같을 수 있지만, 배열 전체가 완전히 동일하면 안됨)

GenData 클래스를 테스트 하기 위한 GenDataTest 클래스를 만들고, main 함수에서 아래의 작업을 수행하세요.

- [Task 1] `N=5`, `min=0`, `max=10` 으로 `getRandIntArr` 호출 후, 리턴 된 배열을 터미널에 출력. 터미널 화면을 캡처하고, 본 문서에 첨부하세요.
- [Task 2] `N=10`, `min=0`, `max=100` 으로 `getRandIntArr` 호출 후, 리턴 된 배열을 터미널에 출력. 터미널 화면을 캡처하고, 본 문서에 첨부하세요.

소스코드도 제출해야 합니다.

답변:

소스코드: [Q1] 정렬할 데이터 생성.txt

```
<terminated> GenDataTest [Java App
```

```
[Task 1]
```

```
9 5 4 0 8
```

```
[Task 2]
```

```
37 56 33 66 78 97 9 62 20 85
```

[Q 2] 버블 정렬 (오름차순 정렬) [10 점]

강의자료를 참고하여 **버블 정렬** 알고리즘을 구현하세요 (강의자료의 코드를 일부 수정해서 사용해도 됩니다). N=10, min=0, max=100 으로 설정하고, getRandIntArr 를 호출하여 정렬되지 않은 int 배열 arr 을 생성하세요.

- 정렬하기 전 배열 arr 을 터미널에 출력하세요.
- **버블 정렬**을 이용해 arr 배열을 정렬하고, 정렬된 배열 arr 을 터미널에 출력하세요.

터미널 화면을 캡처하고, 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드: [Q2] 버블 정렬.txt

```
<terminated> BubbleSort [Java Application] C:\Program Files\WJ
```

```
Before : [19, 52, 97, 42, 98, 7, 37, 43, 80, 54]
```

```
After : [7, 19, 37, 42, 43, 52, 54, 80, 97, 98]
```

[Q 3] 선택 정렬 (오름차순 정렬) [10 점]

강의자료를 참고하여 **선택 정렬** 알고리즘을 구현하세요 (강의자료의 코드를 일부 수정해서 사용해도 됩니다). N=10, min=0, max=100 으로 설정하고, getRandIntArr 를 호출하여 정렬되지 않은 int 배열을 생성하세요.

- 정렬하기 전 배열을 터미널에 출력하세요.
 - **선택 정렬**을 이용해 정렬하고, 정렬된 배열을 터미널에 출력하세요.
- 터미널 화면을 캡처하고, 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드: [Q3] 선택 정렬.txt

```
<terminated> SelectionTest [Java Application] C:\Program Files\
Before : [51, 60, 96, 75, 78, 85, 28, 54, 85, 45]
After  : [28, 45, 51, 54, 60, 75, 78, 85, 85, 96]
```

[Q 4] **삽입 정렬** (오름차순 정렬) [10 점]

강의자료를 참고하여 **삽입 정렬** 알고리즘을 구현하세요 (강의자료의 코드를 일부 수정해서 사용해도 됩니다). N=10, min=0, max=100 으로 설정하고, getRandIntArr 를 호출하여 정렬되지 않은 int 배열을 생성하세요.

- 정렬하기 전 배열을 터미널에 출력하세요.
- **삽입 정렬**을 이용해 정렬하고, 정렬된 배열을 터미널에 출력하세요.

터미널 화면을 캡처하고, 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드: [Q4] 삽입 정렬.txt

```
<terminated> InsertionTest [Java Application] C:\Program File:
Before : [60, 52, 54, 11, 59, 18, 2, 3, 38, 96]
After  : [2, 3, 11, 18, 38, 52, 54, 59, 60, 96]
```

[Q 5] 합병 정렬 (오름차순 정렬) [10 점]

강의자료를 참고하여 **합병 정렬** 알고리즘을 구현하세요 (강의자료의 코드를 일부 수정해서 사용해도 됩니다). N=10, min=0, max=100 으로 설정하고, getRandIntArr 를 호출하여 정렬되지 않은 int 배열을 생성하세요.

- 정렬하기 전 배열을 터미널에 출력하세요.
- **합병 정렬**을 이용해 정렬하고, 정렬된 배열을 터미널에 출력하세요.

터미널 화면을 캡처하고, 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드: [Q5] 합병 정렬.txt

```
<terminated> MergeTest [Java Application] C:\Program Files\Wj:
Before : [50, 56, 35, 27, 44, 2, 78, 21, 83, 17]
After  : [2, 17, 21, 27, 35, 44, 50, 56, 78, 83]
```

[Q 6] 퀵 정렬 (오름차순 정렬) [10 점]

강의자료를 참고하여 **퀵 정렬** 알고리즘을 구현하세요 (강의자료의 코드를 일부 수정해서 사용해도 됩니다). N=10, min=0, max=100 으로 설정하고, getRandIntArr 를 호출하여 정렬되지 않은 int 배열을 생성하세요.

- 정렬하기 전 배열을 터미널에 출력하세요.
- **퀵 정렬**을 이용해 정렬하고, 정렬된 배열을 터미널에 출력하세요.

터미널 화면을 캡처하고, 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드: [Q6] 퀵 정렬.txt

```
<terminated> QuickTest [Java Application] C:\Program Files\WJ
Before : [9, 60, 27, 6, 61, 25, 54, 66, 34, 74]
After  : [6, 9, 25, 27, 34, 54, 60, 61, 66, 74]
```

[Q 7] 정렬 알고리즘 비교 (오름차순 정렬) [30 점]

N=10000, min=0, max=N 으로 정렬되지 않은 int 배열을 생성하세요. **깊은 복사**를 사용해서, **동일한 배열**을 5 개 생성하세요 (arr1, arr2, ..., arr5).

- 버블 정렬로 arr1 을 정렬하고, 시간을 측정해서 터미널에 출력하세요
- 선택 정렬로 arr2 를 정렬하고, 시간을 측정해서 터미널에 출력하세요.
- 삽입 정렬로 arr3 을 정렬하고, 시간을 측정해서 터미널에 출력하세요.
- 합병 정렬로 arr4 를 정렬하고, 시간을 측정해서 터미널에 출력하세요.
- 퀵 정렬로 arr5 를 정렬하고, 시간을 측정해서 터미널에 출력하세요.

* 시간을 측정하는 코드는 지난 [실습과제 03, Q 1 실행시간]문제를 참고하세요. 실행시간을 측정하는 코드는 아래에 첨부되어 있습니다.

* 본인이 사용하는 컴퓨터/노트북의 성능이 10,000 개의 데이터를 처리/정렬할 수 없는 수준이라고 생각되면, N 을 더 작은 숫자로 줄여도 됩니다.

터미널 화면을 캡처하고, 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

[예시]

```
<terminated> PerfTest [Java Application] C:\Program Files\WJ
[Bubble] Time diff (ms): 115
[Selection] Time diff (ms): 34
[Insertion] Time diff (ms): 43
[Merge] Time diff (ms): 3
[Quick] Time diff (ms): 4
```

참고: 실행시간을 측정하는 코드

fMyFunction 함수의 실행 시간 측정하기

```

DataGen dg = new DataGen();
int[] arr = dg.getRandIntArr(N,min,max); // create an array
long beforeTime = System.currentTimeMillis(); // start the timer
fMyFunction(arr); //실행시간을 측정하고자 하는 함수를 여기에 넣으세요
long afterTime = System.currentTimeMillis(); // finish the timer
long diffTimeMilliSecond = afterTime-beforeTime; // get the diff
System.out.println("Time diff (ms): " + diffTimeMilliSecond);

```

답변:

소스코드: [Q7] 정렬 알고리즘 비교.txt

* N = 25000

```

<terminated> TimeTest [Java Application]
[Bubble] Time diff (ms): 1258
[Selection] Time diff (ms): 709
[Insertion] Time diff (ms): 963
[Merge] Time diff (ms): 21
[Quick] Time diff (ms): 19

```

끝! 수고하셨습니다 ☺