

[제 8 주 실습]

외부정렬

강 지 훈

jhkang@cnu.ac.kr



[문제 8] External Sort



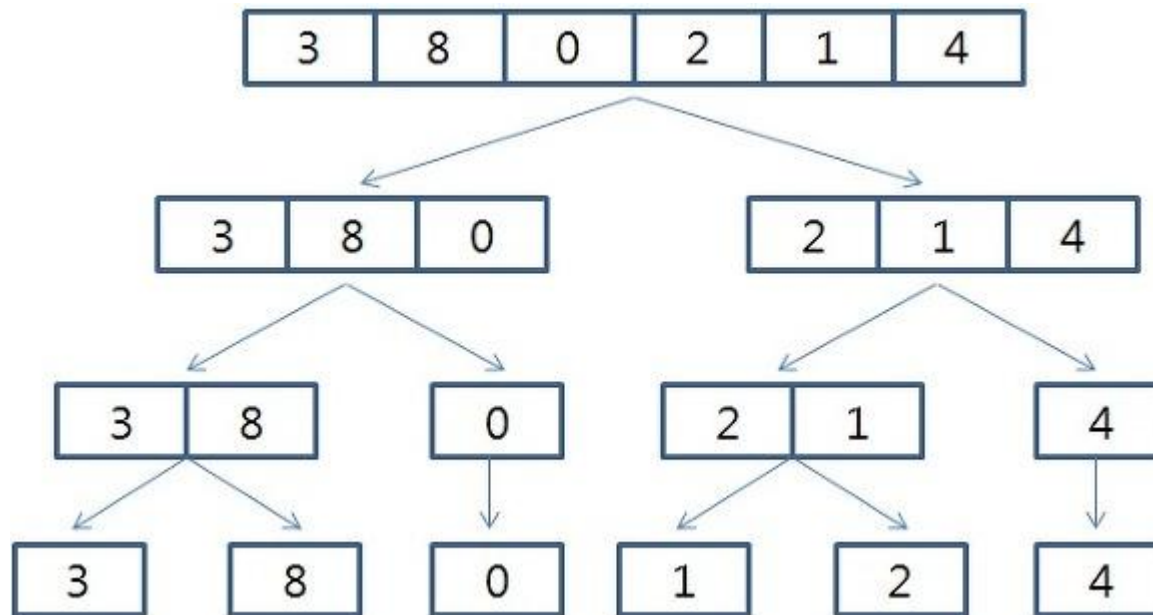
□ 문제 개요

- 2-Way Merge Sort 알고리즘을 이해하고, External Sort를 이용하여 이를 구현한다.



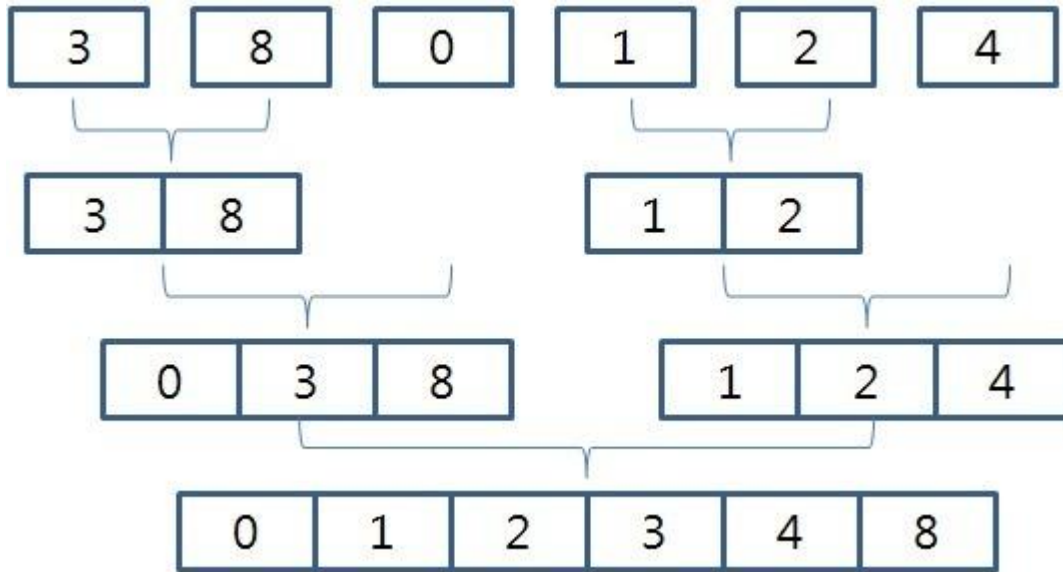
□ 2원 병합 정렬(2-way Merge Sort)

- 전체 원소를 하나의 단위로 분할 한 뒤 분할한 원소를 다시 병합하는 정렬
- 분할



□ 2원 병합 정렬(2-way Merge Sort)

■ 병합



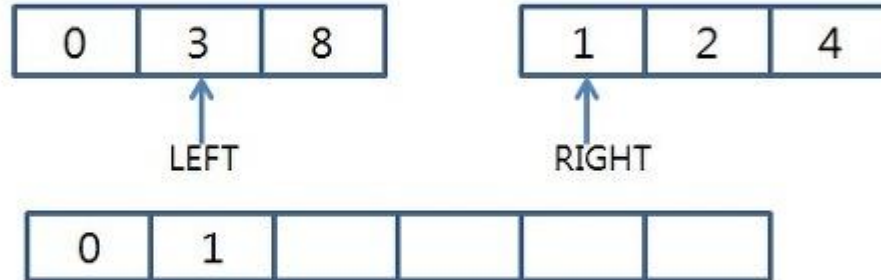
□ 2원 병합 정렬(2-way Merge Sort)

■ 정렬 과정

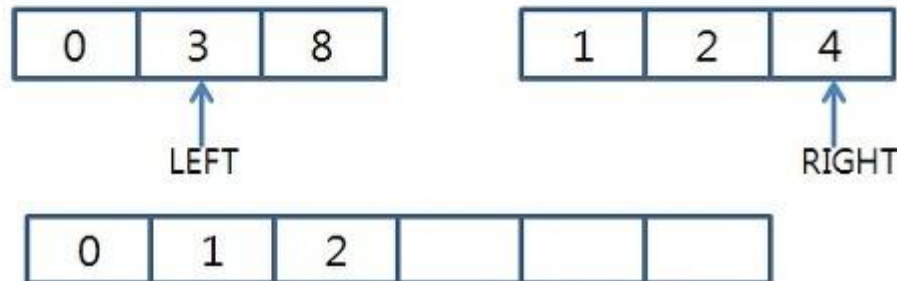
[STEP 1]



[STEP 2]



[STEP 3]



외부 정렬

- 데이터의 양이 많아 메모리를 이용 하지 못하여 디스크를 이용하여 정렬을 하는 방법

file0	1	8	9	16	...
-------	---	---	---	----	-----

file1	2	3	10	15	...
-------	---	---	----	----	-----

file2	7	6	11	14	...
-------	---	---	----	----	-----

file3	4	5	12	13	...
-------	---	---	----	----	-----

Devide files

result	1	2	3	4	5	6	...
--------	---	---	---	---	---	---	-----

외부 정렬

file0	1	8	9	16	...
	↑				
file1	2	3	10	15	...
	↑				
file2	7	6	11	14	...
	↑				
file3	4	5	12	13	...
	↑				

Devided files

result 1

외부 정렬

file0	1	8	9	16	...
		↑			
file1	2	3	10	15	...
	↑				
file2	7	6	11	14	...
	↑				
file3	4	5	12	13	...
	↑				

Devided files

result 1 2

외부 정렬

file0	1	8	9	16	...
		↑			
file1	2	3	10	15	...
		↑			
file2	7	6	11	14	...
	↑				
file3	4	5	12	13	...
	↑				

Devided files

result 1 2 3

외부 정렬

file0	1	8	9	16	...
		↑			
file1	2	3	10	15	...
			↑		
file2	7	6	11	14	...
	↑				
file3	4	5	12	13	...
	↑				

Devided files

result 1 2 3 4

출력의 예

input original file name : OriginalData

input result file name : ResultData

END External Sort OriginalData > ResultData

OriginalData x	
	0 10 20 30
1	1799151400
2	502966176
3	1399633731
4	1713310671
5	920426901
6	1260852296
7	1923446012
8	824975050
9	2021426201
10	373146338
11	46492280
12	2136350936
13	918150932
14	1954642874
15	1172145830
16	1076816066
17	33541624
18	1054870363
19	426478321
20	1434687307
21	286298821
22	1145581958
23	481702986
24	917572717
25	36488988
26	1956793662

ResultData x	
	0 10 20 30
1	2795
2	3812
3	3839
4	6855
5	7361
6	7449
7	7619
8	10075
9	10195
10	13217
11	13530
12	14261
13	14736
14	15379
15	15712
16	16578
17	20056
18	20550
19	25209
20	29564
21	30268
22	30652
23	30812
24	31018
25	31116
26	31547

로컬 디스크 (D:) \ JAVA \ workspace \ DS2_08		
이름	수정한 날짜	크기
.settings	2013-10-17 오후...	4
bin	2013-10-22 오후...	I
src	2013-10-22 오후...	I
.classpath	2013-10-17 오후...	C
.project	2013-10-17 오후...	F
OriginalData	2013-10-23 오전...	I
ResultData	2013-10-23 오전...	I
temp0	2013-10-23 오전...	L
temp1	2013-10-23 오전...	L
temp2	2013-10-23 오전...	L
temp3	2013-10-23 오전...	L
temp4	2013-10-23 오전...	L
temp5	2013-10-23 오전...	L
temp6	2013-10-23 오전...	L
temp7	2013-10-23 오전...	L
temp8	2013-10-23 오전...	L
temp9	2013-10-23 오전...	L
temp10	2013-10-23 오전...	L
temp11	2013-10-23 오전...	L
temp12	2013-10-23 오전...	L
temp13	2013-10-23 오전...	L
temp14	2013-10-23 오전...	L
temp15	2013-10-23 오전...	L
temp16	2013-10-23 오전...	L
temp17	2013-10-23 오전...	L
temp18	2013-10-23 오전...	L
temp19	2013-10-23 오전...	L



이 과제에서 필요한 객체는?

- AppView
- ApplicationController
 - AppView
 - ExternalSort
- ExternalSort
 - QuickSort



□ AppController의 공개 함수는?

■ 사용자에게 필요한 함수 (Public Functions)

- public void run()

□ AppView의 공개 함수는?

■ 사용자에게 필요한 함수 (Public Functions)

- public AppView()
- public void showInputOriginalDataFileName()
- public void showInputResultDataFileName()
- public void showSortingResult(String originalDataFileName, String resultDataFileName)
- public void showMsg(String aString)
- public String inputString()

ExternalSort의 멤버 함수는?

■ 사용자에게 필요한 함수 (Public Functions)

- `public ExternalSort(int givenDiskSize, int givenMemorySize)`
- `public void externalSort(String dataName, String resultName)`

□ QuickSort의 멤버 함수는?

■ 사용자에게 필요한 함수 (Public Functions)

- [실습 7]에서 사용한 QuickSort를 사용
- public QuickSort()
- public void sort(int [] _data, int size)



Class “AppController”



□AppController – 비공개 인스턴스 변수

```
import java.io.FileWriter;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.Random;
```

```
public class AppController {  
    static int N = 2000000; // size of the file in disk  
    static int M = 100000; // max items the memory buffer can hold  
    private AppView _appView;  
    private ExternalSort _externalSort;
```



□ AppController 의 공개 함수 run()의 구현

```
public void run(){
    this._appView = new AppView();
    _externalSort = new ExternalSort(N, M);

    this._appView.showInputOriginalDataFileName();
    String originalDataFileName = this._appView.inputString();

    this._appView.showInputResultDataFileName();
    String resultDataFileName = this._appView.inputString();

    _externalSort.externalSort(originalDataFileName,resultDataFileName);

    this._appView.showSortingResult(originalDataFileName,
resultDataFileName);
}
```


Class "AppView"



□ AppView — 비공개 인스턴스 변수

```
import java.util.Scanner;
```

```
public class AppView {  
    private Scanner _scanner;
```

□ AppView의 Public Method

■ AppView 의 Public Member function의 사용법과 구현

- public AppView()
 - ◆ 생성자
- public void showInputOriginalDataFileName()
 - ◆ "input original file name : "를 화면에 내보낸다.
- public void showInputResultDataFileName()
 - ◆ "input result file name : " 를 화면에 내보낸다.
- public void showSortingResult(String originalDataFileName, String resultDataFileName)
 - ◆ "END External Sort : " + originalDataFileName + " > " + resultDataFileName 를 화면에 내보낸다.

□ AppView의 Public Method

■ AppView 의 Public Member function의 사용법과 구현

- public void showMsg(String aMsg)
 - ◆ String 메시지를 하나 받아 화면에 내보낸다.
- public String inputString()
 - ◆ 문자열을 입력 받아 반환한다.
 - ◆ _scanner의 nextLine()등을 이용한다.

Class “ExternalSort”



Member Functions의 구현

ExternalSort 의 Public Member function의 사용법

- `public ExternalSort(int givenDiskSize, int givenMemorySize)`
 - ◆ ExternalSort의 생성자
- `public void externalSort(String dataName, String resultName)`
 - ◆ Devide와 marge작업을 수행한다.



ExternalSort – 비공개 인스턴스 변수

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
```

```
public class ExternalSort {
    private int _diskSize;
    private int _memoryBufferSize;
    private QuickSort _quickSort;
    private String _tfile = "temp";
```



□ Member Functions의 구현

■ ExternalSort 의 Private Member function의 사용법

- private void devide(String dataName)
 - ◆ 정렬 하려는 파일을 원하는 개수만큼 잘라서 정렬하여 저장
- private void marge(String resultName)
 - ◆ 정렬 되어 있는 파일들을 모아서 하나의 출력 파일을 만들어 냄

자바의 파일처리

■ 자바에서의 스트림(Stream)

- 데이터의 흐름을 형성해 주는 통로
- 파일 또는 각종 I/O 장치와의 데이터 이동에 사용 되는 인스턴스(객체)

■ 자바에서의 입출력 스트림

- 입력 스트림(input Stream) – 프로그램으로 데이터를 읽어들이는 스트림
- 출력 스트림(Output Stream) – 프로그램으로부터 데이터를 내보내는 스트림

□자바의 예외처리

■예외(오류)의 종류

- 문법 오류

- ◆ 잘못된 명령어를 입력했거나, 타이핑의 실수로 발생하는 오류

- 런타임 오류

- ◆ 런타임 오류는 프로그램 작성 후 실행할 때 발생하는 오류

- 알고리즘 오류 :

- ◆ 주어진 문제에 대한 잘못된 해석으로 잘못된 결과를 초래하는 에러를 알고리즘 오류 또는 로직 오류

□자바의 예외처리

- 프로그램 실행도중 예외가 발생하여 프로그램이 중단 되는 것을 미리 예방하는 것
- 예외가 발생한 부분은 어쩔 수 없어도 이외의 영역은 끝까지 수행 될 수 있도록 함

Object - Throwable

- Error
- Exception- IOException
 - ClassNotFoundException
 -
 - RuntimeException
 - ArithmeticException
 - ClassCastException
 - NullPointerException
 -
 - IndexOutOfBoundsException



□자바의 예외처리

■에러 - Error

- 프로그램 코드에 의해서 수습 될 수 없는 심각한 오류

■예외 - Exception

- 프로그램 코드에 의해서 수습 될 수 있는 다소 미약한 오류
- RuntimeException클래스와 그 자손클래스들 - 프로그래머의 실수로 발생하는 예외
- Exception클래스와 그 자손클래스들(RuntimeException과 자손들 제외) - 사용자의 실수와 같은 외적인 요인에 의해 발생하는 예외.

□자바의 예외처리

■ 직접처리방법

```
try{
    예외발생 가능성있는 실행문;
}catch(발생되는 Exception 클래스 타입 객체이름){
    Exception 처리 실행문;
}catch(발생되는 Exception 클래스 타입 객체이름){
    Exception 처리 실행문;
}
..
finally{
    예외발생여부와 상관없이 무조건 실행되는 실행문;
}
```

- catch문 여러번 작성할때 반드시 **서브 클래스**부터 작성한다.
- try는 catch또는 finally와 함께 사용한다.

□자바의 예외처리

■ 자주 볼수있는 예외

- NullPointerException
 - ◆ 실제 객체가 없을때
- ArrayIndexOutOfBoundsException
 - ◆ 배열의 범위를 벗어났을 때
- NumberFormatException
 - ◆ 숫자로 바꿀 수 없는 데이터를 숫자로 변경 할 때

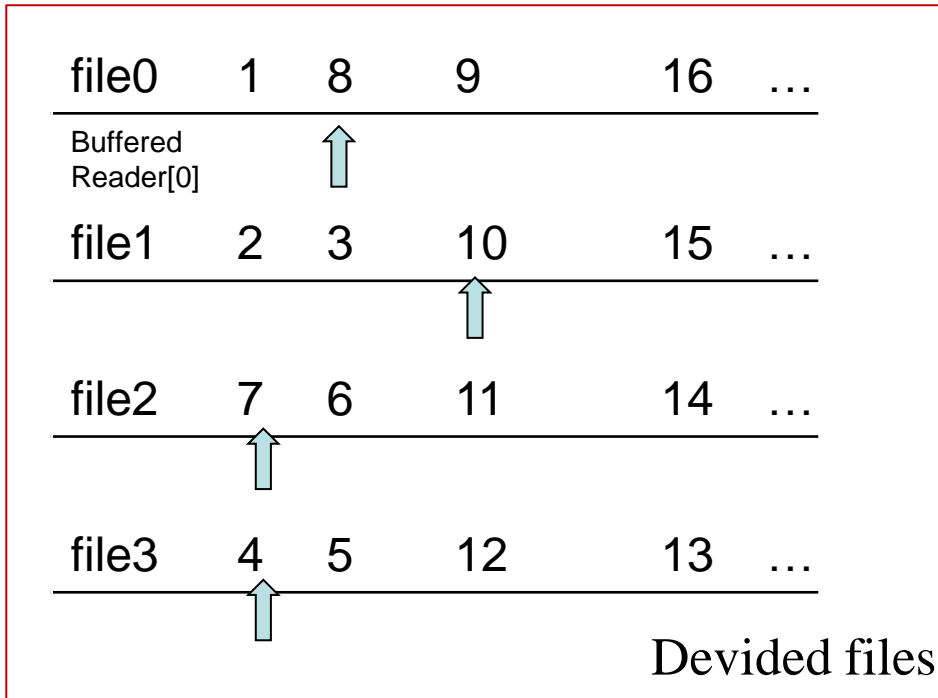
□ Member Functions의 구현

■ ExternalSort 의 Public Member function의 구현

- public ExternalSort(int givenDiskSize, int givenMemorySize)
 - ◆ givenDiskSize를 _diskSize에 저장
 - ◆ givenMemorySize를 _memoryBufferSize에 저장
- public void externalSort(String dataName, String resultName)

```
public void externalSort(String dataName, String resultName) {  
    this.devide(dataName);  
    this.marge(resultName);  
}
```

외부 정렬



topNums	8	10	7	4
result	1	2	3	

Member Functions의 구현

ExternalSort 의 private Member function의 구현

```
private void devide(String dataName) {
    int[] buffer = new int[_memoryBufferSize < _diskSize ? _memoryBufferSize: _diskSize];
    this._quickSort = new QuickSort();
    try {
        FileReader fileReader = new FileReader(dataName);
        BufferedReader bufferedReader = new BufferedReader(fileReader);
        int slices = (int) Math.ceil((double) _diskSize / _memoryBufferSize);

        int i, j;
        i = j = 0;
        for (i = 0; i < slices; i++) {
            for (j = 0; j < (_memoryBufferSize < _diskSize ? _memoryBufferSize : _diskSize); j++) {
                buffer[j] = Integer.parseInt(bufferedReader.readLine());
            }
        }
    }
```

file read

// 만들어진 데이터를 정렬하기 위하여 quick sort를 실행

~~~~~

```
FileWriter fileWriter = new FileWriter(_tfile + Integer.toString(i)+ ".txt");
PrintWriter printWriter = new PrintWriter(fileWriter);
for (int k = 0; k < j; k++){
    printWriter.println(buffer[k]);
}
printWriter.close();
fileWriter.close();
```

File write

```
}
bufferedReader.close();
fileReader.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
```



# Member Functions의 구현

## ExternalSort 의 private Member function의 구현

### private void marge(String resultName) [1] – 정렬 준비

- ◆ slices를 설정
  - ◆ open된 파일들의 가장 앞 숫자를 가지고 있는 배열 topNums를 slices 만큼 생성
  - ◆ 분할 되어 있는 파일들의 정보를 저장 할 BufferedReader 배열 bufferedReader배열을 slices 만큼 생성
  - ◆ Slices의 개수만큼 분할 되어 있는 파일을 각각 열어서 bufferedReader 배열의 각 위치에 저장
    - devide에서 분할 파일의 이름을 생성한 방법을 참고하여 작성
- ```
FileWriter fileWriter = new FileWriter(_tfile + Integer.toString(i)+ ".txt");
PrintWriter printWriter = new PrintWriter(fileWriter);
```
- ◆ Open한 파일들의 각각의 앞에 있는 값을 하나 읽어서 topNums에 저장
  - ◆ 결과를 저장할 resultName을 가지는 FileWriter와 PrintWriter를 생성

# □ Member Functions의 구현

## ■ ExternalSort 의 private Member function의 구현

- private void marge(String resultName) [2] – 실제 정렬
  - ◆ \_diskSize만큼 값을 찾아 내어야 함
  - ◆ 각 topNums중에 가장 작은 값을 찾음
  - ◆ 찾아낸 가장 작은 값을 resultName을 가지는 파일에 저장
  - ◆ 저장된 값이 있는 file에서 가장 앞에 있는 값을 다시 읽어서 topNums에 저장
- ◆ 모든 작업에는 try/catch로 이루어진 예외처리 작업이 있어야 함

## □[문제 8] 요약

### ■ 외부정렬

#### ● 외부 정렬이란 무엇인가?

- ◆ 외부 정렬에 대하여 이해하고 외부정렬이 이루어지는 그림으로 그려서 설명하시오.

#### ● 외부 정렬의 시간과 공간 복잡도

- 병합될 파일의 크기
- N개의 레코드에 대하여 수행되는 단계의 숫자
- 두개의 파일을 병합하는데 걸리는 시간
- 총 소요시간
- 공간 복잡도

■ 각 요약에 대한 내용을 보고서에 작성하여 제출하세요.



# 과제 제출



# □ 과제 제출

■ [pineai@cnu.ac.kr](mailto:pineai@cnu.ac.kr)

- 메일 제목 : [0X]DS2\_08\_학번\_이름
  - ◆ 양식에 맞지 않는 메일 제목은 미제출로 간주됨
  - ◆ 앞의 0X는 분반명 ( 오전10시 : 00반 / 오후4시 : 01반 )

## ■ 제출 기한

- 10월 29일(화) 23시59분까지
- 시간 내 제출 엄수
- 제출을 하지 않을 경우 0점 처리하고, 숙제를 50% 이상 제출하지 않으면 F 학점 처리하며, 2번 이상 제출하지 않으면 A 학점을 받을 수 없다.

# □ 과제 제출

## ■ 파일 이름 작명 방법

● DS2\_08\_학번\_이름.zip

● 폴더의 구성

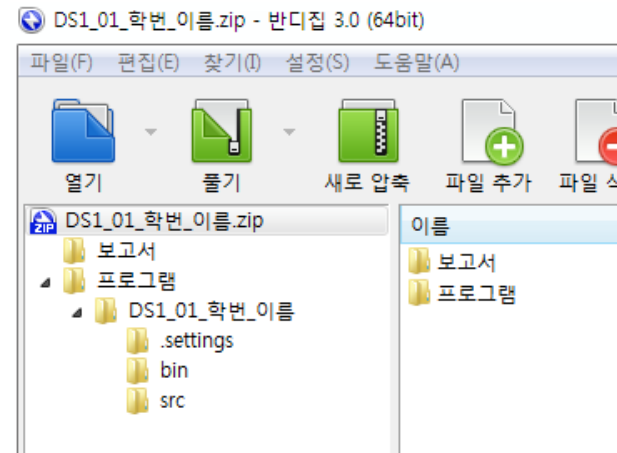
◆ DS2\_08\_학번\_이름

■ 프로그램

- 프로젝트 폴더 / 소스
- 메인 클래스 이름 : DS2\_08\_학번\_이름.java

■ 보고서

- 이곳에 보고서 문서 파일을 저장한다.
- 입력과 실행 결과는 화면 image로 문서에 포함시킨다.
- 문서는 pdf 파일로 만들어 제출한다.



# □ 보고서 작성 방법

## ■ 겉장

- 제목: 자료구조 실습 보고서
- [제xx주] 숙제명
- 제출일
- 학번/이름

## ■ 내용

### 1. 프로그램 설명서

1. 주요 알고리즘 /자료구조 /기타
2. 함수 설명서
3. 종합 설명서 : 프로그램 사용방법 등을 기술

### 2. 구현 후 느낀 점 : 요약의 내용을 포함하여 작성한다.

### 3. 실행 결과 분석

1. 입력과 출력 (화면 capture : 실습예시와 다른 예제로 할 것)
2. 결과 분석

----- 표지 제외 3장 이내 작성 -----

### 4. 소스코드 : 화면 capture가 아닌 소스를 붙여넣을 것 소스는 장수 제한이 없음.

# [제 8 주 실습] 끝

