Data Analysis

(Data modelling, collecting, and analyses 1)

Fall, 2020

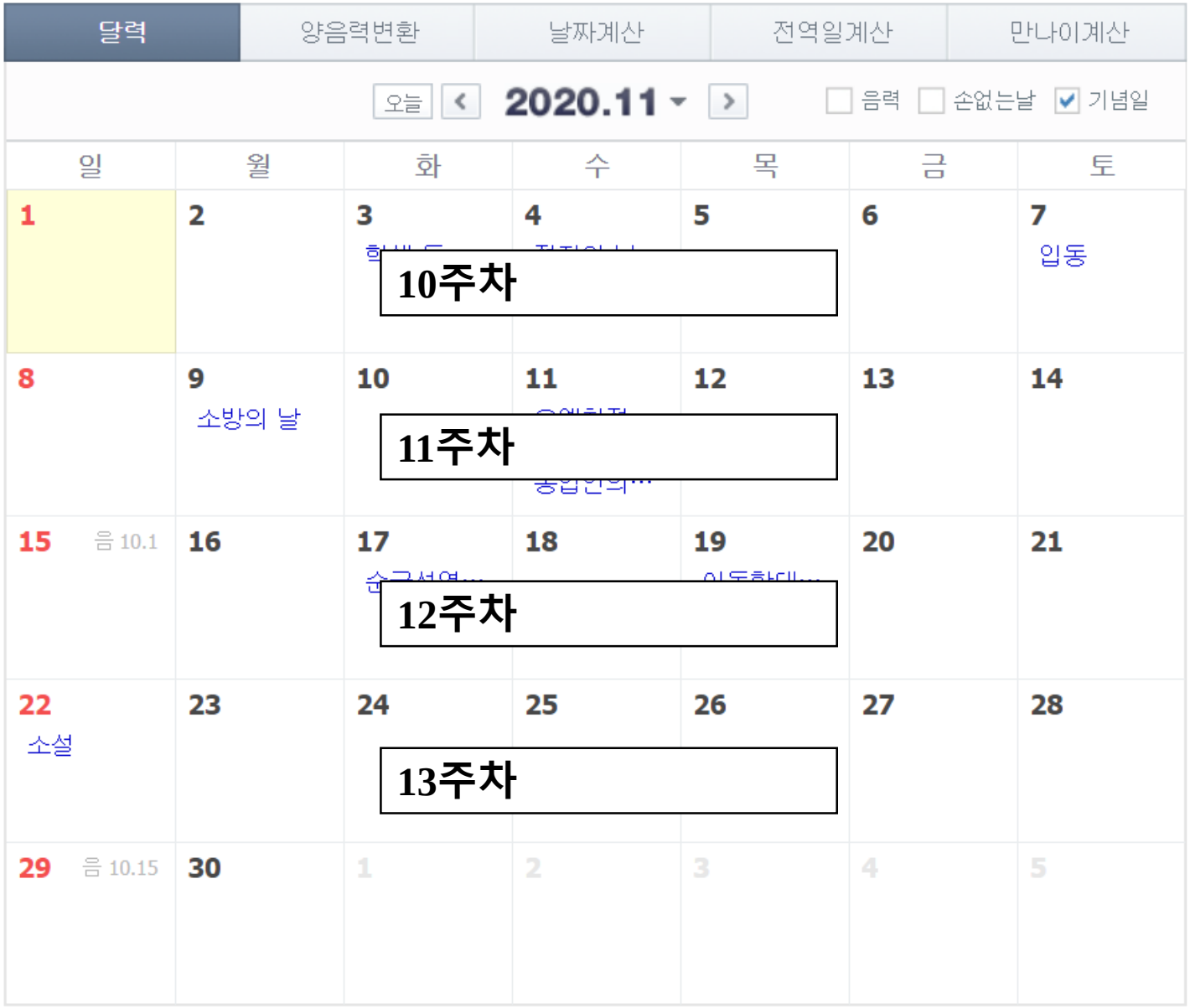
Calendar


달력
양음력변환
날짜계산
전역일계산
만나이계산
오늘
2020.09
음력
손없는날
기념일
일
월
화
수
목
금
토
30
31
1
소개
2
음 7.15
3
환경
세팅
4
지식재산…
5
6
7
백로
8
복습
1
9
10
9. 1
복습
2
11
12
13
14
15
16
17
음 8.1
18
19
청년의 날
3주차
20
21
치매극복…
22
23
24
25
26
4주차
27
28
29
30
1
2
3
5주차

Calendar


달력
양음력변환
날짜계산
전역일계산
만나이계산
오늘
2020.10
음력
손없는날
기념일
일 월 화 수 목 금 토
27 28 29 30 1 음 8.15 추석 국군의 날 2 노인의 날 3 개천절
4 5 세계 한… 6 7 8 9 한글날 10
11 12 13 14 15 16 부마민주… 17 음 9.1 문화의 날
18 19 20 21 22 23 상강 24 국제연합일
25 독도의날 중양절 26 27 28 29 지방자치… 30 31 음 9.15
6주차
7주차
8주차: 중간고사
9주차

Calendar


달력
양음력변환
날짜계산
전역일계산
만나이계산
오늘
2020.11
음력
손없는날
기념일
일
월
화
수
목
금
토
1
2
3
4
5
6
7
입동
10주차
8
9
소방의 날
10
11
12
13
14
11주차
15
음 10.1
16
17
18
19
20
21
12주차
22
소설
23
24
25
26
27
28
13주차
29
음 10.15
30
1
2
3
4
5

Calendar


달력
양음력변환
날짜계산
전역일계산
만나이계산
오늘
2020.12
음력
손없는날
기념일
일 월 화 수 목 금 토
29 30 1 2 3 4 5
무역의 날
14주차
6 7 8 9 10 11 12
대설
15주차
13 14 15 음 11.1 16 17 18 19
16주차: 기말고사 주간
20 21 22 23 24 25 26
동지
성탄절
27 28 29 음 11.15 30 31 1 2
원자력의…

Table of Contents

- Some practice!
- Collecting, modelling, and analyses 1

Question #1

- Compute the number of lines

- Make a variable to keep the number of line → int numLine = 0;
- Increase the variable for each loop
- Print out the variable after the loop

- https://github.com/JaewookByun/plecture/blob/master/data_processing/src/main/java/kr/ac/sejong/advanced_programming/week3/Assignment1.java

Question #2

- Compute the number of lines only if each line does not start with '#'

- Make a variable to keep the number of line → int numLine = 0;
- **Skip the loop whenever the line starts with #**
- Increase the variable for each loop
- Print out the variable after the loop

- https://github.com/JaewookByun/plecture/blob/master/data_processing/src/main/java/kr/ac/sejong/advanced_programming/week3/Assignment2.java

Question #3

- Compute the number of identifiers (The redundancy is allowed)

- Make a variable to keep the number of identifiers → int numID = 0;
- Skip the loop whenever the line starts with #
- Increase the variable for each loop
- Split each line with a delimiter and add its length to numID
- Print out the variable after the loop

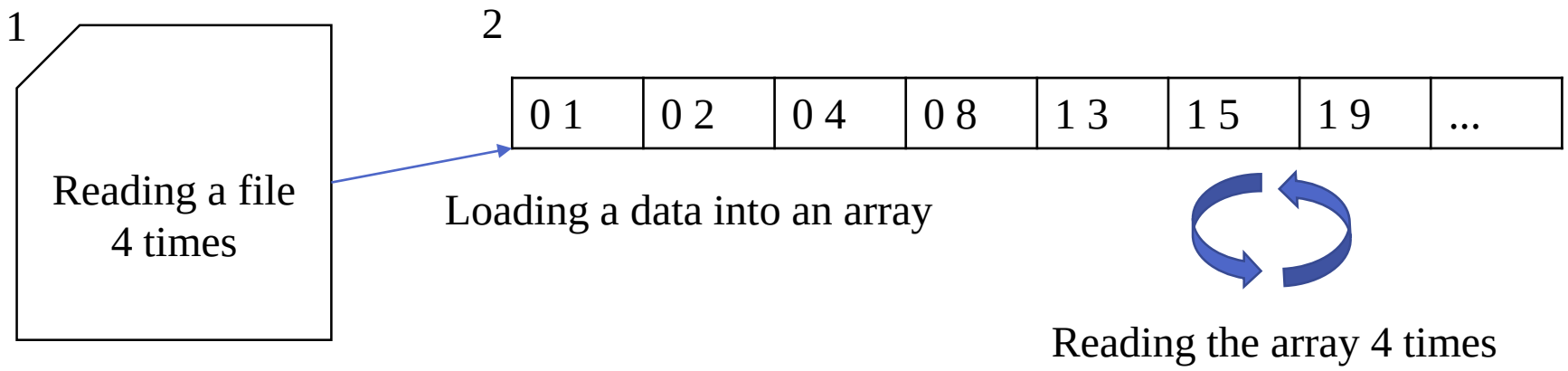
- https://github.com/JaewookByun/plecture/blob/master/data_processing/src/main/java/kr/ac/sejong/advanced_programming/week3/Assignment3.java

Question #4

- Compute the maximum value of left identifier
- https://github.com/JaewookByun/plecture/blob/master/data_processing/src/main/java/kr/ac/sejong/advanced_programming/week3/Assignment6.java
- Compute the maximum value of right identifier
- Compute the minimum value of left identifier
- Compute the minimum value of right identifier

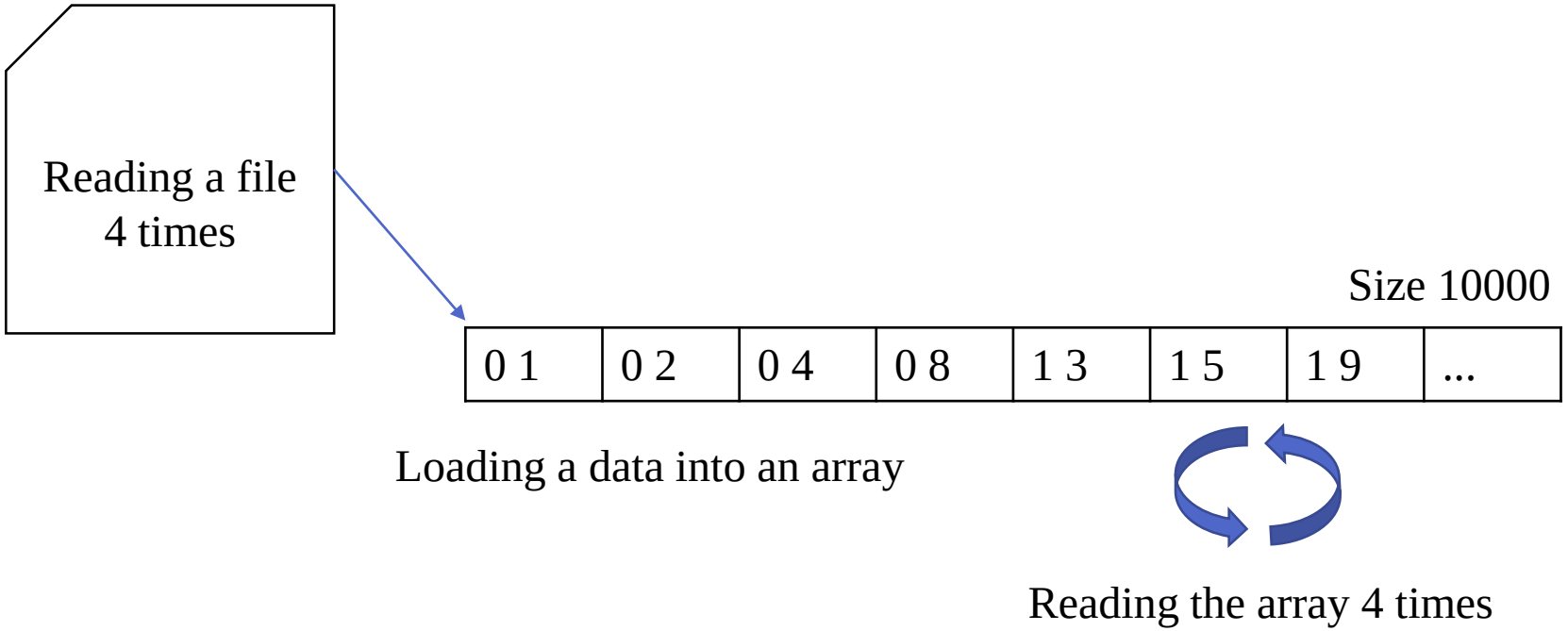
How about collecting the data and reusing it

- Compute the maximum value of left identifier
- Compute the maximum value of right identifier
- Compute the minimum value of left identifier
- Compute the minimum value of right identifier



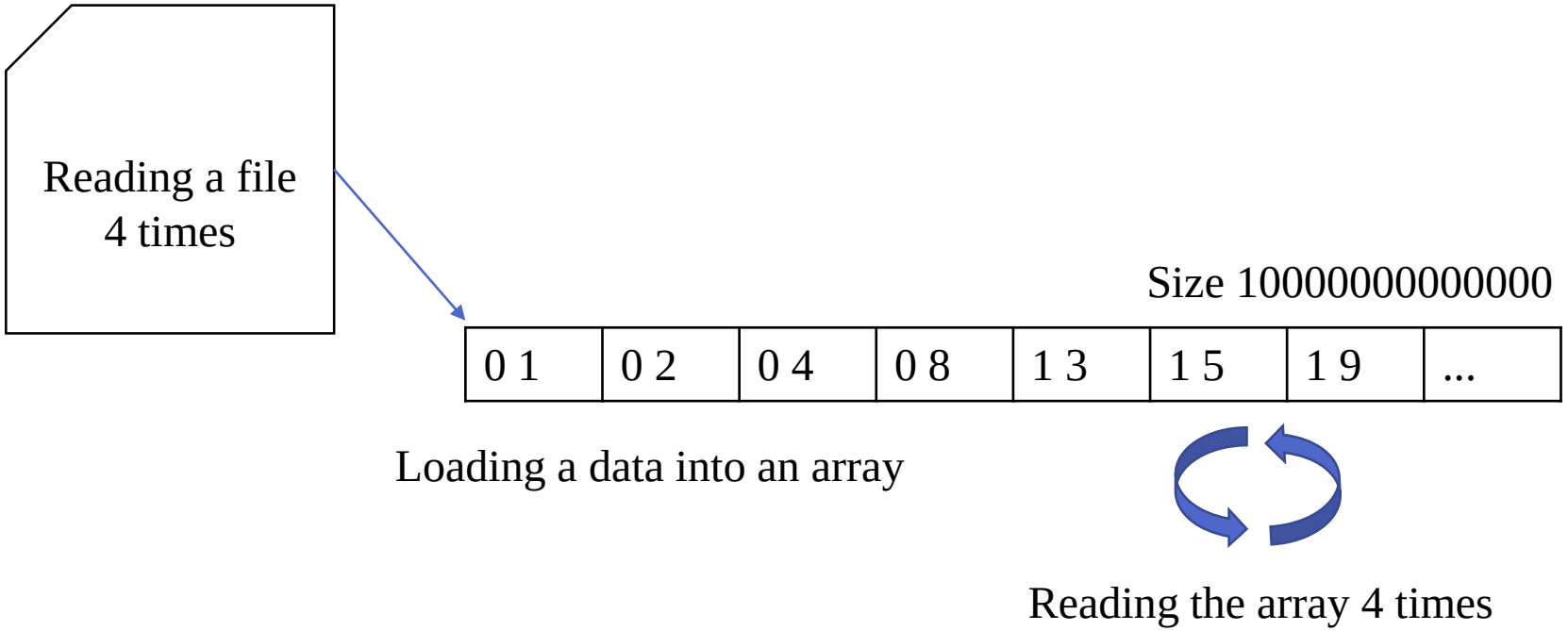
How about collecting the data and reusing it

- Try and Consider
  - Declare String[] data = new String[10000];
    - Pros
    - Cons



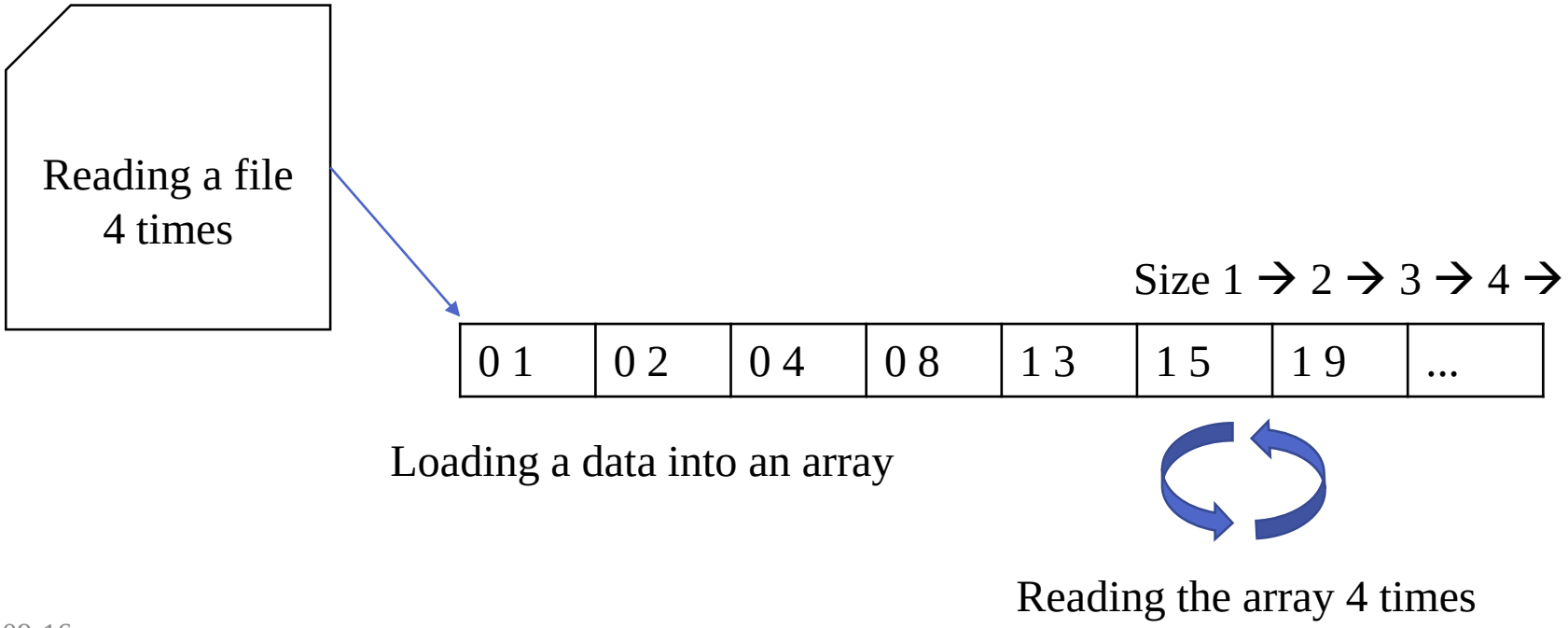
How about collecting the data and reusing it

- Try and Consider
  - Declare String[] data = new String[10000000000000];
    - Pros
    - Cons



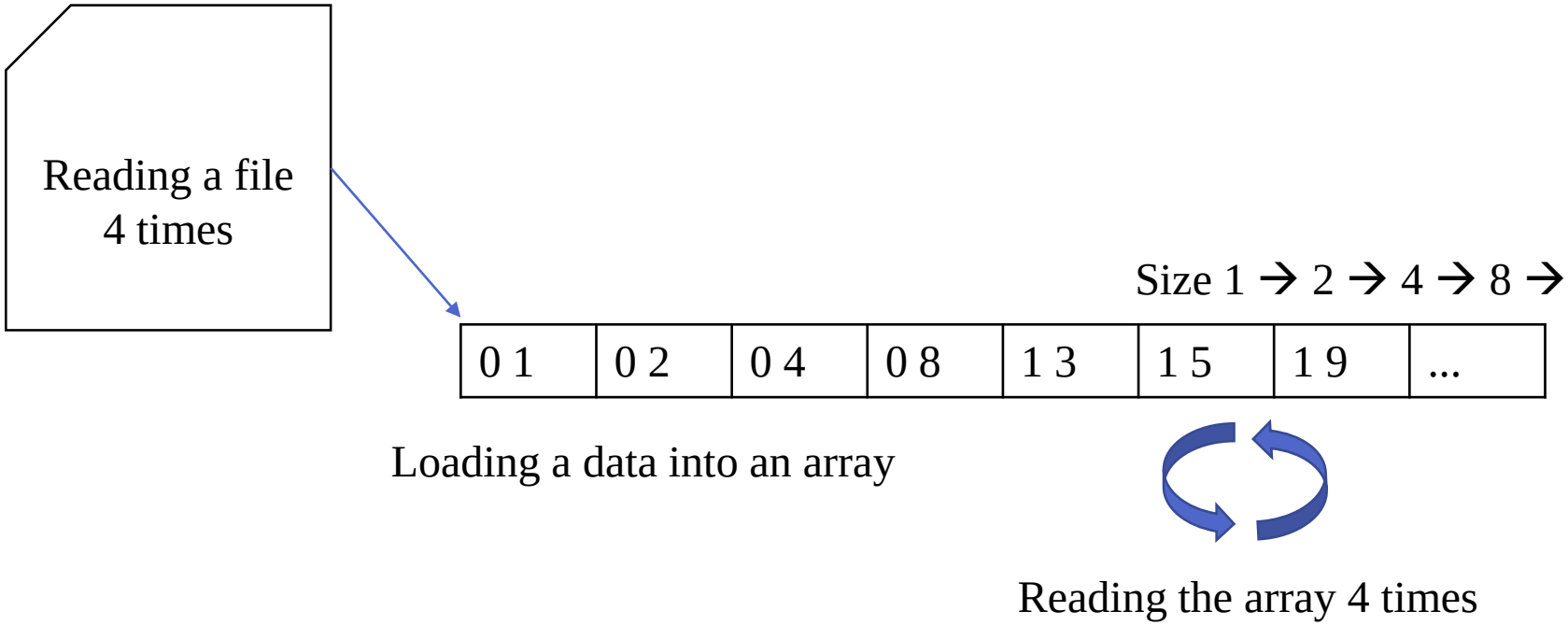
How about collecting the data and reusing it

- Try and Consider
  - Declare String[] data = new String[];
    - Increase the size one by one;
      - if a new record may yield array out of bound exception?
        - declare an array with current length + 1
        - copy data
        - append new record



How about collecting the data and reusing it

- Try and Consider
  - Declare String[] data = new String[];
    - Array doubling



Collection Framework

- Collection
  - a group of elements (instances)

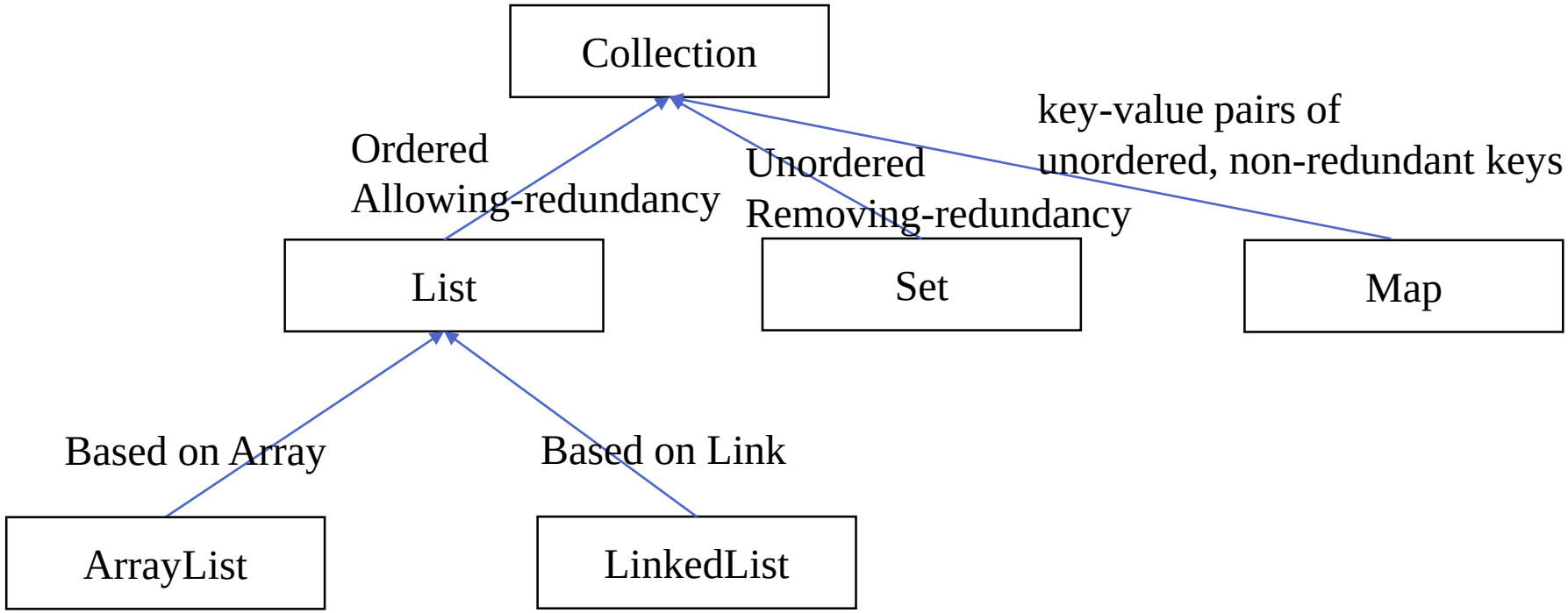
- Framework
  - An abstraction in which software providing generic functionality

- Collection Framework
  - An abstraction of how to manipulate a group of elements
  - Useful classes or interfaces
  - in java.util package since JDK1.2

  - ArrayList
    - Ordered, allowing-redundancy dynamic array

Collection Framework

- Collection hierarchy



Collection Framework

- ArrayList
  - Unordred, Non-redundant elements E
  - Based on Array
  - CRUD by index

```
public class ArrayList<E> extends AbstractList<E>
        implements List<E>, RandomAccess, Cloneable,
java.io.Serializable
{
            …
            Object[] elementData;
            …
}
```

Collection Framework

- ArrayList
  - extending Collection, List

Collection

메서드	설 명
boolean add(Object o) boolean addAll(Collection c)	지정된 객체(o) 또는 Collection(c) 의 객체들을 Collection에 추가한다.
void clear( )	Collection의 모든 객체를 삭제한다.
boolean contains(Object o) boolean containsAll(Collection c)	지정된 객체(o) 또는 Collection의 객체들이 Collection에 포함되어 있는지 확인한다.
boolean equals(Object o)	동일한 Collection인지 비교한다.
int hashCode( )	Collection의 hash code를 반환한다.
boolean isEmpty( )	Collection이 비어있는지 확인한다.
Iterator iterator( )	Collection의 Iterator를 얻어서 반환한다.
boolean remove(Object o)	지정된 객체를 삭제한다.
boolean removeAll(Collection c)	지정된 Collection에 포함된 객체들을 삭제한다.
boolean retainAll(Collection c)	지정된 Collection에 포함된 객체만을 남기고 다른 객체들은 Collection 에서 삭제한다. 이 작업으로 인해 Collection에 변화가 있으면 true를 그렇지 않으면 false를 반환한다.
int size( )	Collection에 저장된 객체의 개수를 반환한다.
Object[ ] toArray( )	Collection에 저장된 객체를 객체배열(Object[ ])로 반환한다.
Object[ ] toArray(Object[ ] a)	지정된 배열에 Collection의 객체를 저장해서 반환한다.

Collection Framework

- ArrayList
  - extending Collection, List

List

메서드	설 명
void add(int index, Object element) boolean addAll(int index, Collection c)	지정된 위치(index)에 객체(element) 또는 컬렉션에 포함된 객체들을 추가한다.
Object get(int index)	지정된 위치(index)에 있는 객체를 반환한다.
int indexOf(Object o)	지정된 객체의 위치(index)를 반환한다. (List의 첫 번째 요소부터 순방향으로 찾는다.)
int lastIndexOf(Object o)	지정된 객체의 위치(index)를 반환한다. (List의 마지막 요소부터 역방향으로 찾는다.)
ListIterator listIterator( ) ListIterator listIterator(int index)	List의 객체에 접근할 수 있는 ListIterator를 반환한다.
Object remove(int index)	지정된 위치(index)에 있는 객체를 삭제하고 삭제된 객체를 반환한다.
Object set(int index, Object element)	지정된 위치(index)에 객체(element)를 저장한다
void sort(Comparator c)	지정된 비교자(comparator)로 List를 정렬한다.
List subList(int fromIndex, int toIndex)	지정된 범위(fromIndex부터 toIndex)에 있는 객체를 반환한다.

Collection Framework

- ArrayList
  - Practice ArrayList
    - CRUD
    - Generics
    - Iterator
    - Comparator
  - Refer to one of reference JAVA의 정석 for more details

Question #5

- Loading a dataset into ArrayList<String> and reuse it four times for
  - Compute the maximum value of left identifier
  - Compute the maximum value of right identifier
  - Compute the minimum value of left identifier
  - Compute the minimum value of right identifier

In-class assignment 1

- Implement your own MyArrayList implementing List<E>
  - Implementing all the methods
  - Data Abstraction
    - Object[]
  - Strategy
    - Array Doubling

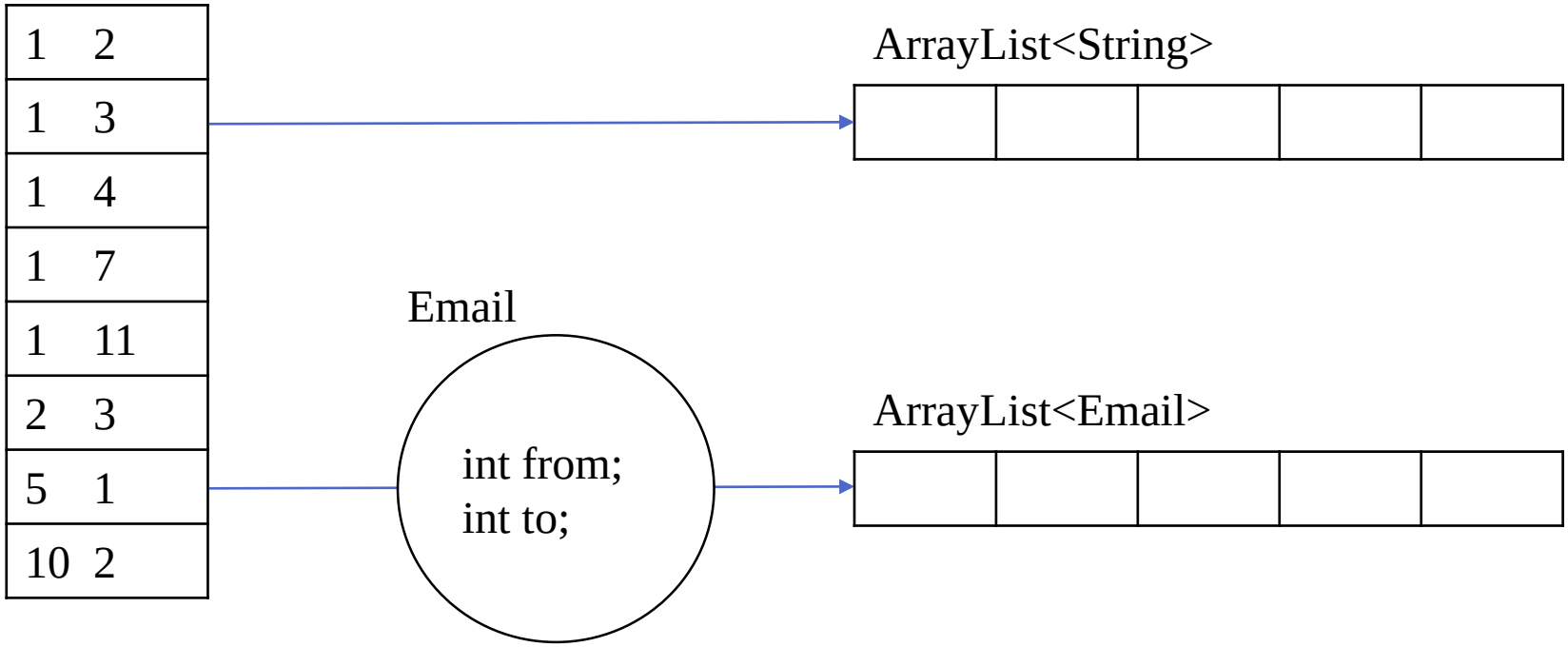
- You have to show me the size of Object[] is doubled whenever exceeding a limit.
  - When loading the line one by one
- You have to show me the size of Object[] becomes half.
  - When removing the element one by one

- Solve Question #5 with your collection

Modelling

- Modelling a class for abstracting a dataset
  - Increasing the accessibility of the concept
  - Practice: get the maximum, minimum identifier

An email dataset



Question #6

- Compute the number of identifiers (The redundancy is **not** allowed)
  - Keep the identifiers without redundancy into an array

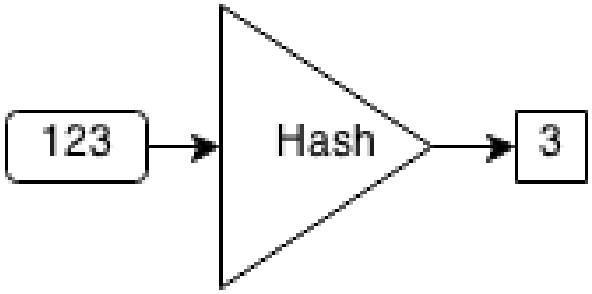
Remaining: 1 2 3 1 4 2									
1									
Remaining: 2 3 1 4 2									
1	2								
Remaining: 3 1 4 2									
1	2	3							
Remaining: 1 4 2									
1	2	3							
Remaining: 4 2									
1	2	3	4						
Remaining: 2									
1	2	3	4						

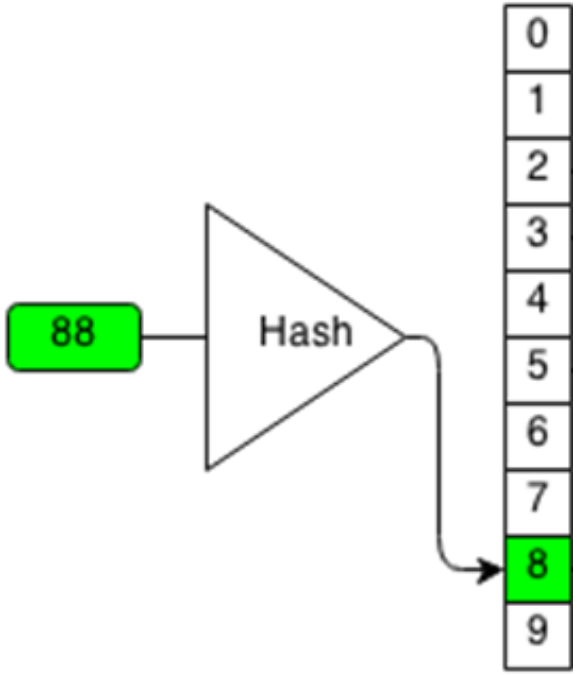
Question #6

- Compute the number of identifiers (The redundancy is **not** allowed)
  - See how the trend of the computation time for computing each line

HashSet

- HashSet consists of non-redundant elements
- Rethink the importance of ‘Data Structure’ and ‘Algorithms’
- We can find an element of HashSet O(1)





http://www.thecshandbook.com/Hash_Set

Summary

- Some Practice!
- Collecting, modelling, and analyses based on Array-based collection

- Next Week
  - Collecting, modelling, and analyses based on Hash-based collection