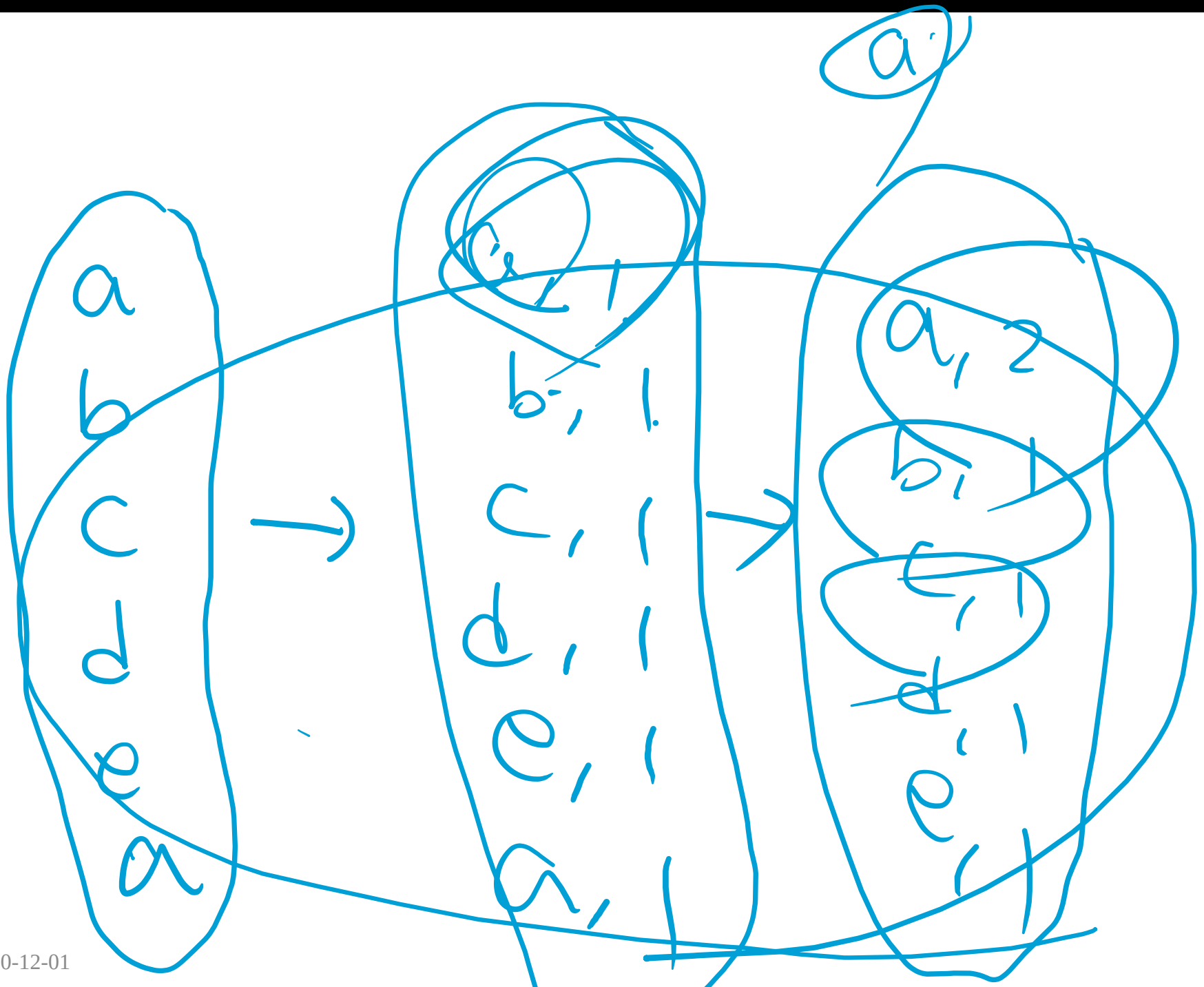
Data Analysis
(Stream and Parallel Processing 4)

Fall, 2020


a
b
c
d
e
a, 1
b, 1
c, 1
d, 1
e, 1
a, 1
a, 2
b, 1
c, 1
d, 1
e, 1

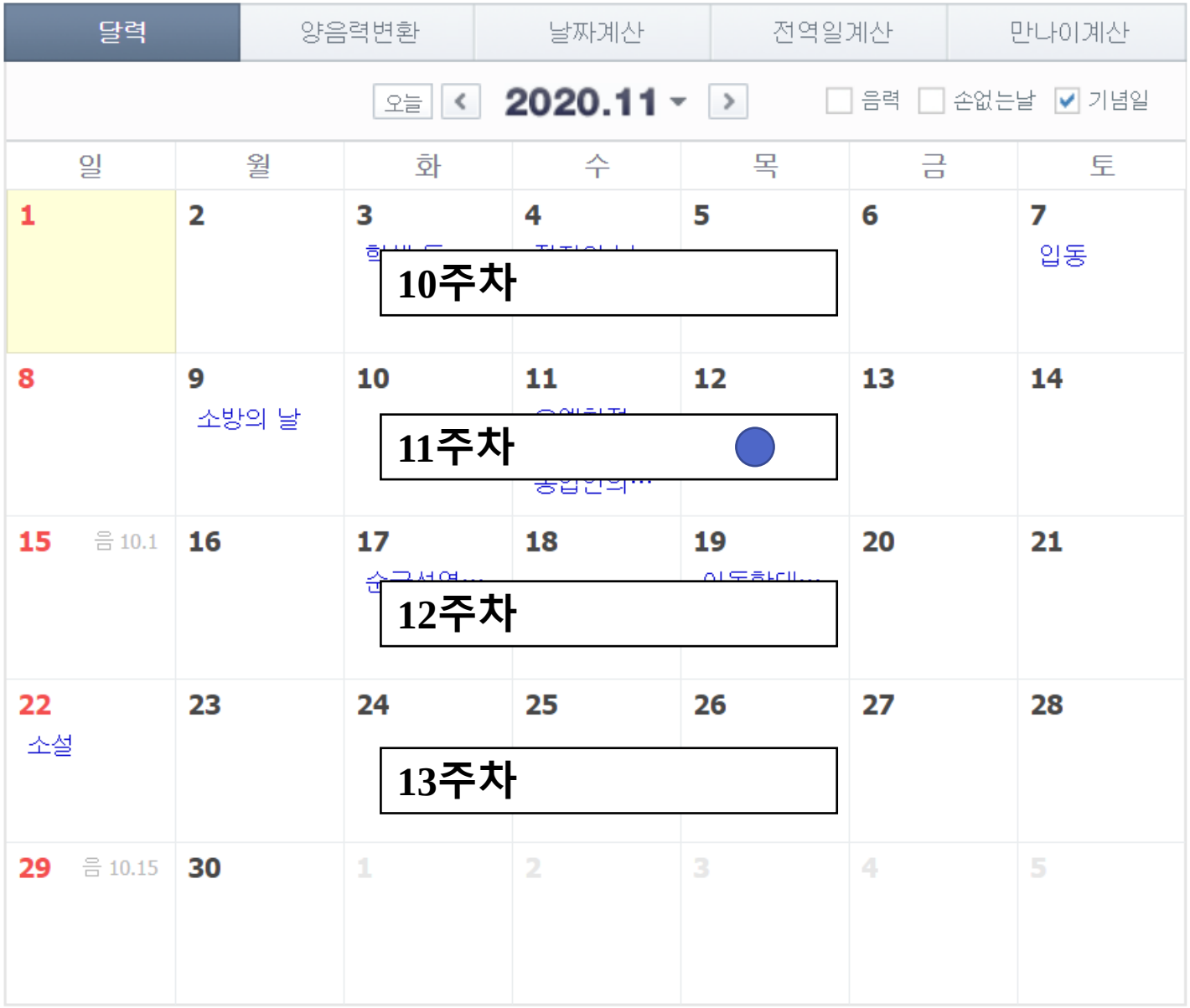
Calendar


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

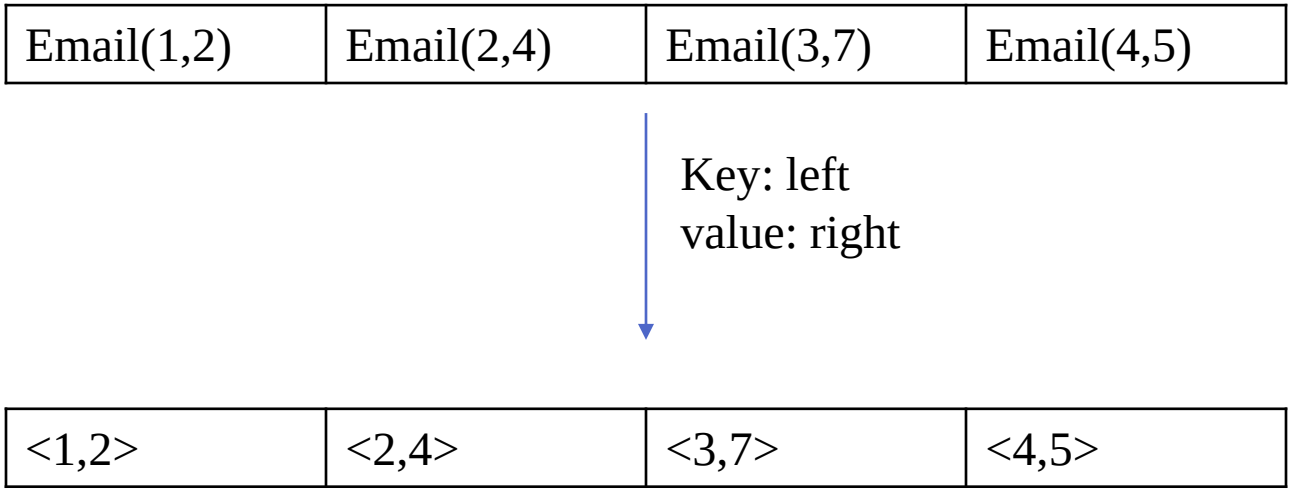
Calendar


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

Stream API Methods

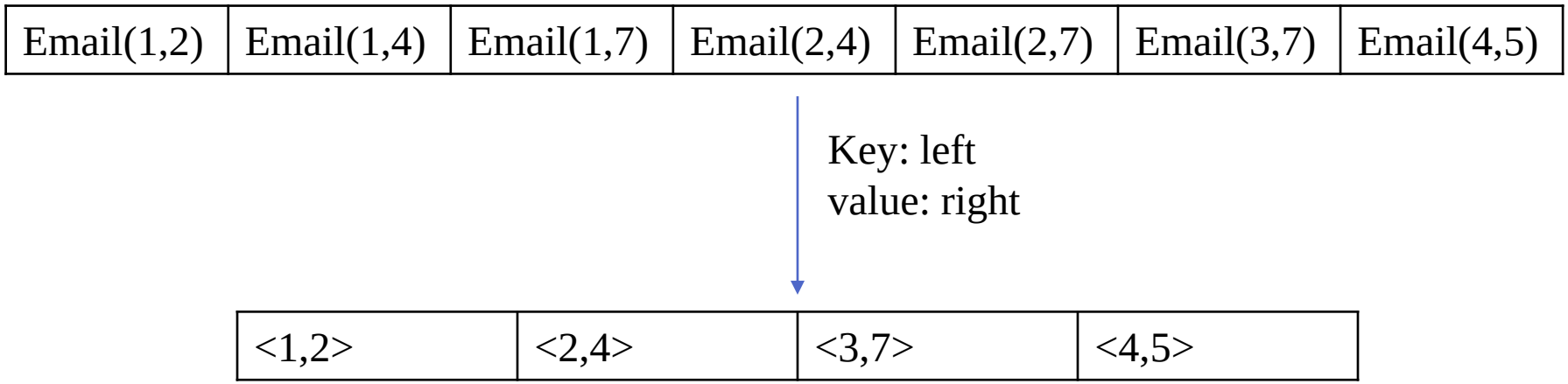
- Map<B, C> Collectors.toMap(Function<A, B> keyMapper, Function<A, C> valueMapper)
  - Returns a Collector that accumulates elements into a Map whose keys and values are the result of applying the providedmapping functions to the input elements.



- Practice #1 – do above

Stream API Methods

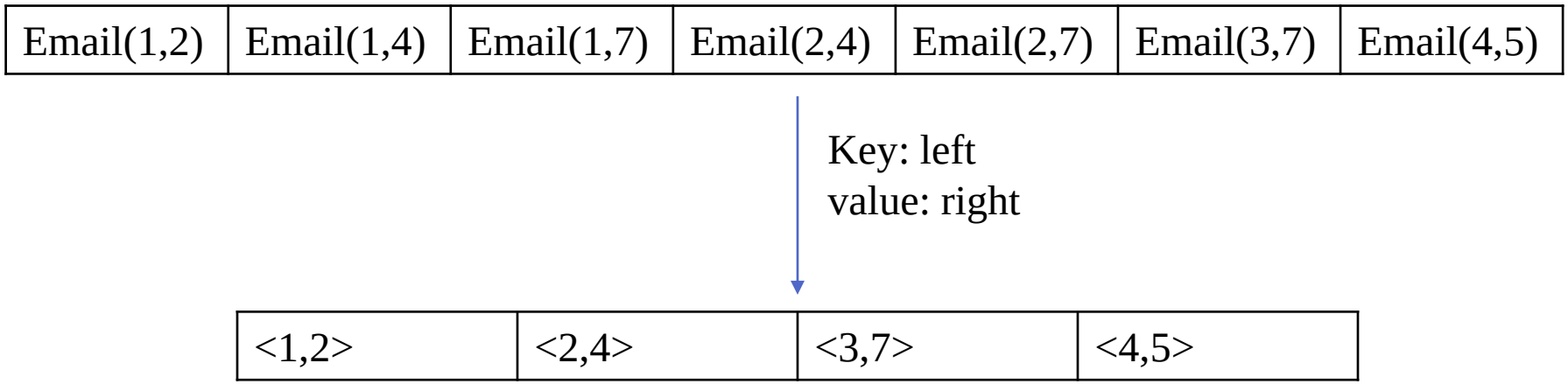
- Map<B, C> Collectors.toMap(Function<A, B> keyMapper, Function<A, C> valueMapper)
  - Returns a Collector that accumulates elements into a Map whose keys and values are the result of applying the providedmapping functions to the input elements.



- Practice #2 – do above yielding Duplicate key exception

Stream API Methods

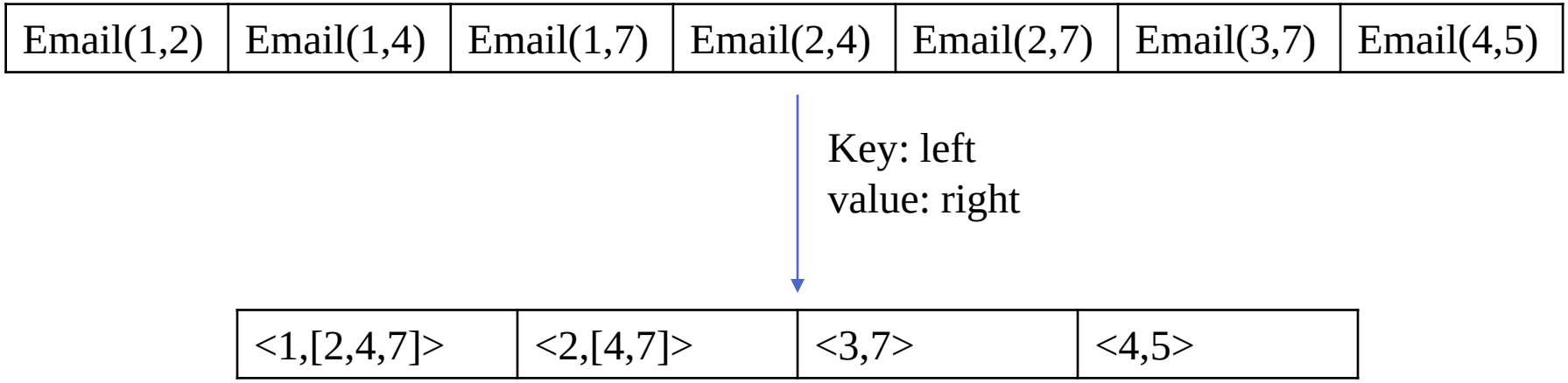
- Map<B, C> Collectors.toMap(Function<A, B> keyMapper, Function<A, C> valueMapper, BinaryOperator<C,C> merge)
  - Returns a Collector that accumulates elements into a Map whose keys and values are the result of applying the providedmapping functions to the input elements.



- Practice #2 – do above yielding Duplicate key exception by just choosing e1 integer

Stream API Methods

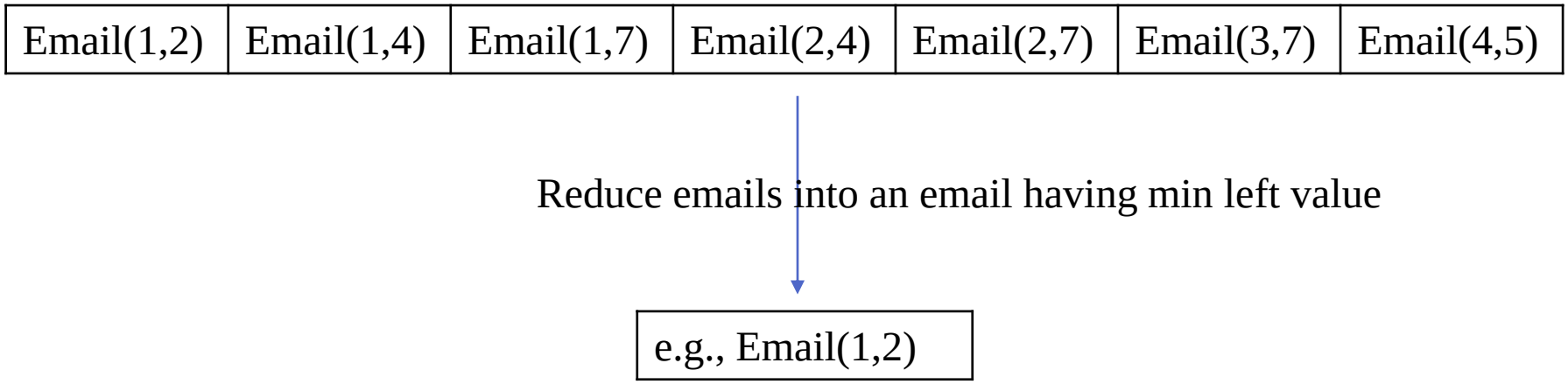
- ??Map<B, C> Collectors.toMap(Function<A, B> keyMapper, Function<A, C> valueMapper, BinaryOperator<C,C> merge. Supplier supplier)
  - Returns a Collector that accumulates elements into a Map whose keys and values are the result of applying the providedmapping functions to the input elements.



- Practice #4 – collect into TreeMap

Stream API Methods

- T Collectors.reducing(BinaryOperator<T> op)
  - Returns a Collector which performs a reduction of itsinput elements under a specified BinaryOperator. The resultis described as an Optional<T>.

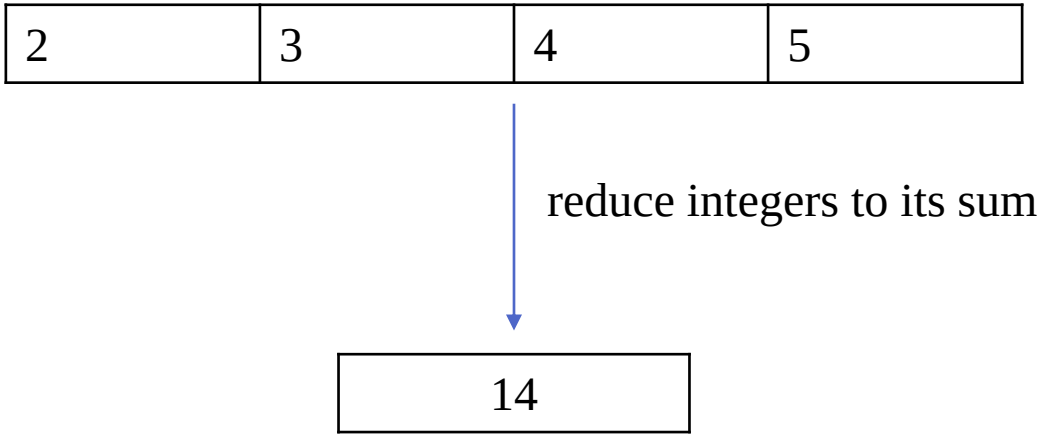


- Practice #5

--> Would like to compare left values?

Stream API Methods

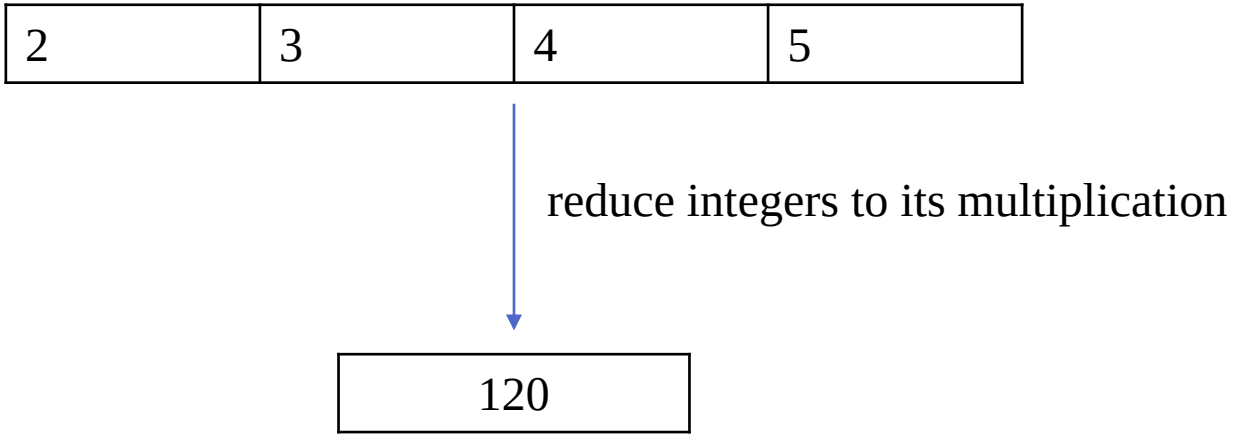
- U Collectors.reducing(U, BinaryOperator<U> op)
  - Returns a Collector which performs a reduction of itsinput elements under a specified BinaryOperator using theprovided identity.



- Practice #6
  - With (BinaryOperator) as well as (U, BinaryOperator)

Stream API Methods

- U Collectors.reducing(U, BinaryOperator<U> op)
  - Returns a Collector which performs a reduction of itsinput elements under a specified BinaryOperator using theprovided identity.

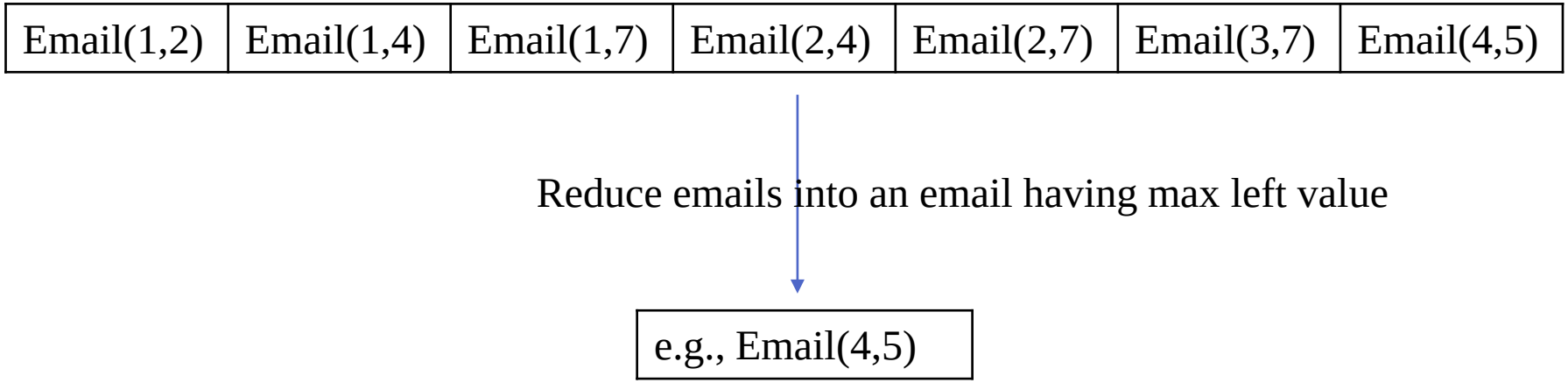


- Practice #7
  - With (U, BinaryOperator)

How it works?
Stream ParallelStream

Stream API Methods

- T Collectors.reducing(U,Funtion<U,T> mapper, BinaryOperator<T> op)
  - Returns a Collector which performs a reduction of itsinput elements under a specified mapping function and BinaryOperator. This is a generalization of reducing(Object, BinaryOperator) which allows a transformationof the elements before reduction



- Practice #8
  - With (U, mapper, BinaryOperator)

Stream API Methods

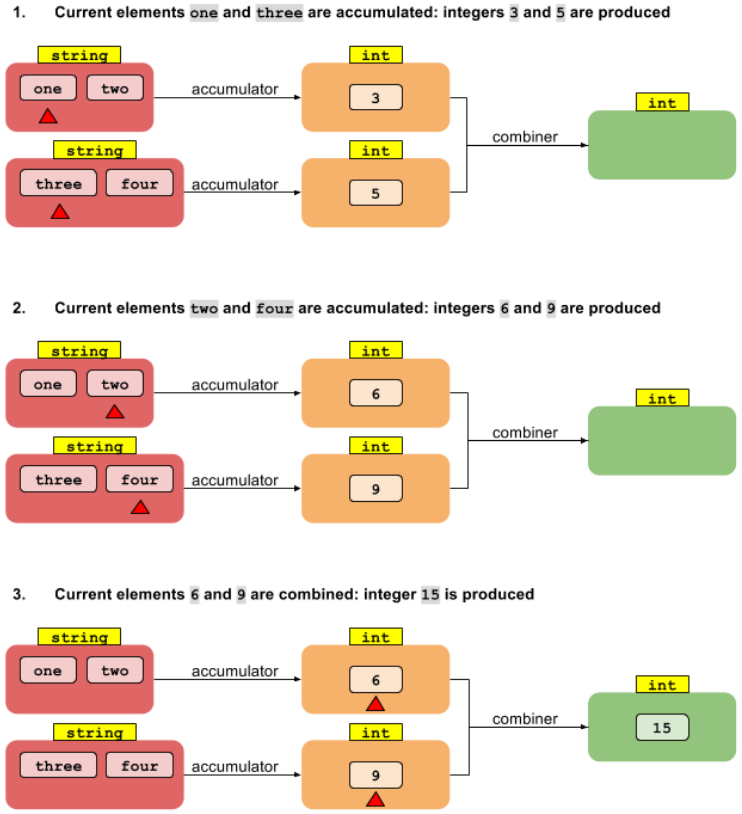
- Optional<Integer> java.util.stream.Stream.reduce(BinaryOperator<Integer> accumulator)
  - Performs a reduction on theelements of this stream, using an associative accumulationfunction, and returns an Optional describing the reduced value,if any. This is equivalent to:

- Practice #9
  - Same with Collect(Collectors.reduce)

Stream API Methods

- Optional<Integer> java.util.stream.Stream.reduce(BinaryOperator<Integer> accumulator)
  - Performs a reduction on theelements of this stream, using an associative accumulationfunction, and returns an Optional describing the reduced value,if any.

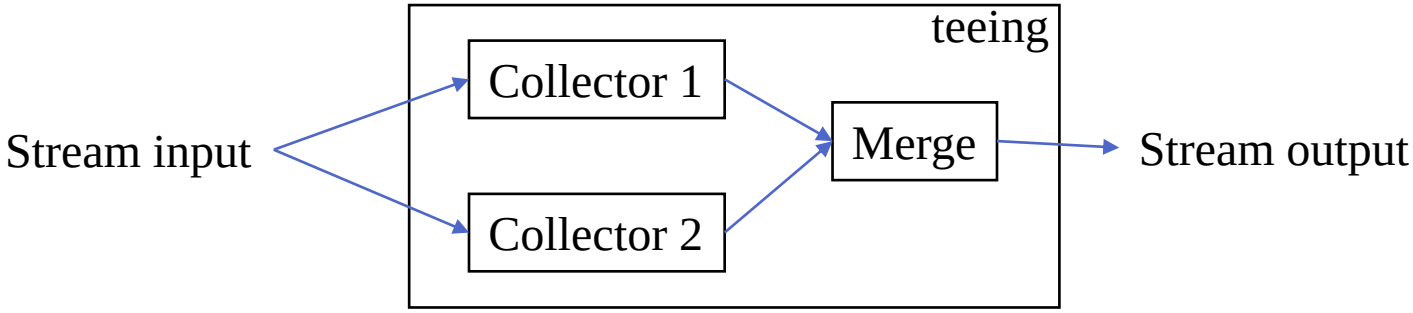
- Works for Parallel Stream
- Practice #10
  - Prepare ArrayList<String>
  - Reduce(0, String->length, length+length)



https://stackoverflow.com/questions/24308146/why-is-a-combiner-needed-for-reduce-method-that-converts-type-in-java-8

Stream API Methods

- <Integer, Optional<Integer>, Optional<Integer>, List<Integer>> Collector<Integer, ?, List<Integer>> java.util.stream.Collectors.teeing(Collector<? super Integer, ?, Optional<Integer>> downstream1, Collector<? super Integer, ?, Optional<Integer>> downstream2, BiFunction<? super Optional<Integer>, ? super Optional<Integer>, List<Integer>> merger)
  - Performs a reduction on theelements of this stream, using an associative accumulationfunction, and returns an Optional describing the reduced value,if any.



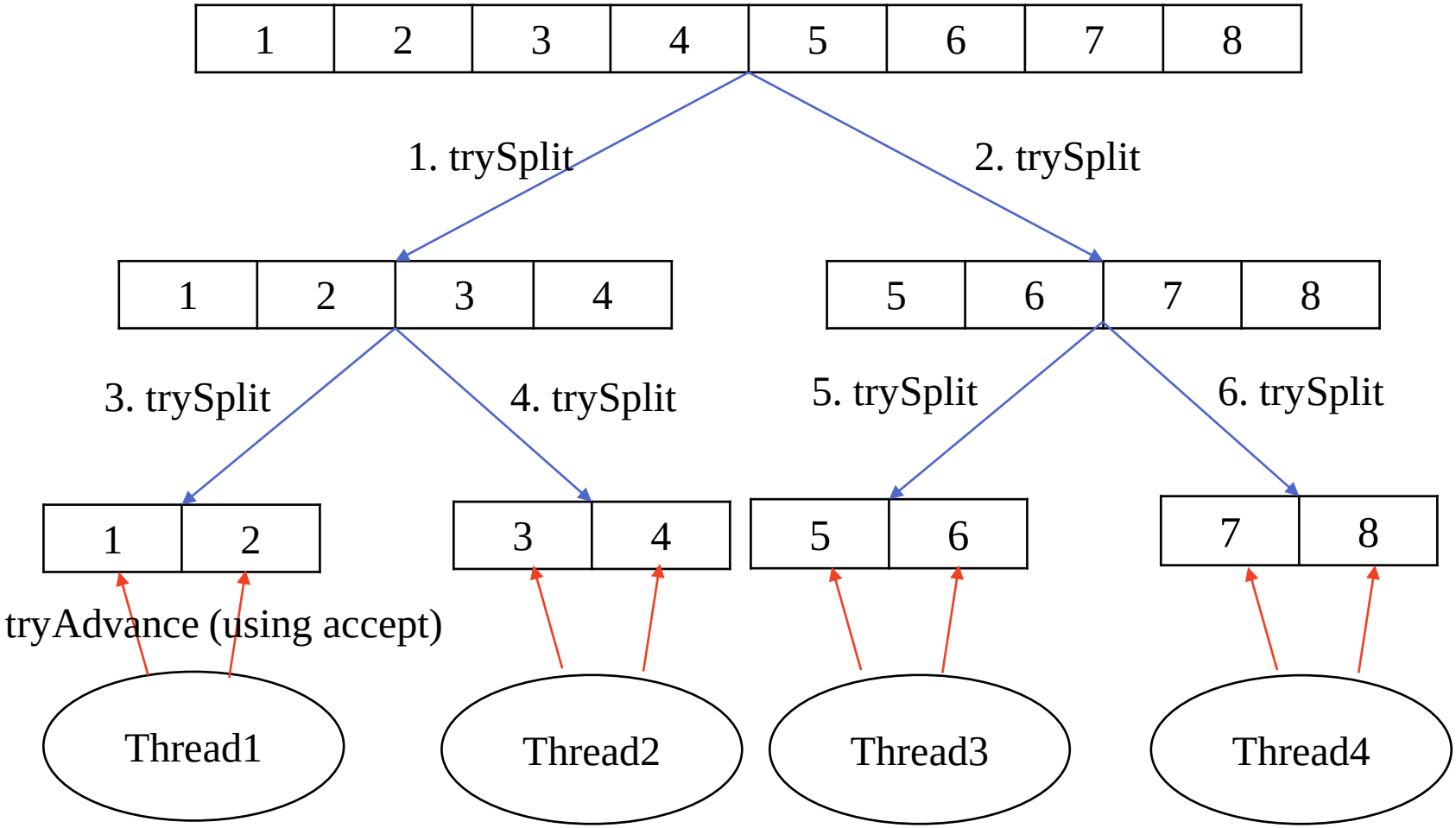
- Practice #11
  - Prepare ArrayList<Integer> of 1,2,3,4,5
  - Collect List of SUM and MUL

Stream API Methods

- Spliterator
  - An object for traversing and partitioning elements of a source. The source of elements covered by a Spliterator could be, for example, an array, a Collection, an IO channel, or a generator function.
  - Boolean tryAdvanced(Consumer action);
    - If a remaining element exists, performs the given action on it,returning true; else returns false. If thisSpliterator is ORDERED the action is performed on thenext element in encounter order. Exceptions thrown by theaction are relayed to the caller.
  - Spliterator trySplit()
    - If this spliterator can be partitioned, returns a Spliteratorcovering elements, that will, upon return from this method, notbe covered by this Spliterator.
  - long estimateSize()
    - If this spliterator can be partitioned, returns a Spliteratorcovering elements, that will, upon return from this method, notbe covered by this Spliterator.
  - int characteristics()
    - Returns a set of characteristics of this Spliterator and itselements. The result is represented as ORed values from ORDERED, DISTINCT, SORTED, SIZED, NONNULL, IMMUTABLE, CONCURRENT, SUBSIZED. Repeated calls to characteristics() ona given spliterator, prior to or in-between calls to trySplit,should always return the same result.

Stream API Methods

- Spliterator (working example)



Stream API Methods

- Spliterator (code review)
  - Practice12
    - Custom Spliterator
    - Debug and feel
    - estimateSize is used for spliterator
      - 0, 1 does not invoke trySplit anymore
    - You can customize minimum elements of a sub spliterator
    - Characteristics
      - `int isOrdered = Spliterator.ORDERED;`
      - `int isDistinct = Spliterator.DISTINCT;`
      - `int isSorted = Spliterator.SORTED;`
      - `int isSized = Spliterator.SIZED;`
      - `int isNonnull = Spliterator.NONNULL;`
      - `int isImmutable = Spliterator.IMMUTABLE;`
      - `int isConcurrent = Spliterator.CONCURRENT;`
      - `int isSubsized = Spliterator.SUBSIZED;`

Wrap-up

- Stream and Parallel Processing 4 - Advanced