BIGDATA
- Problem1

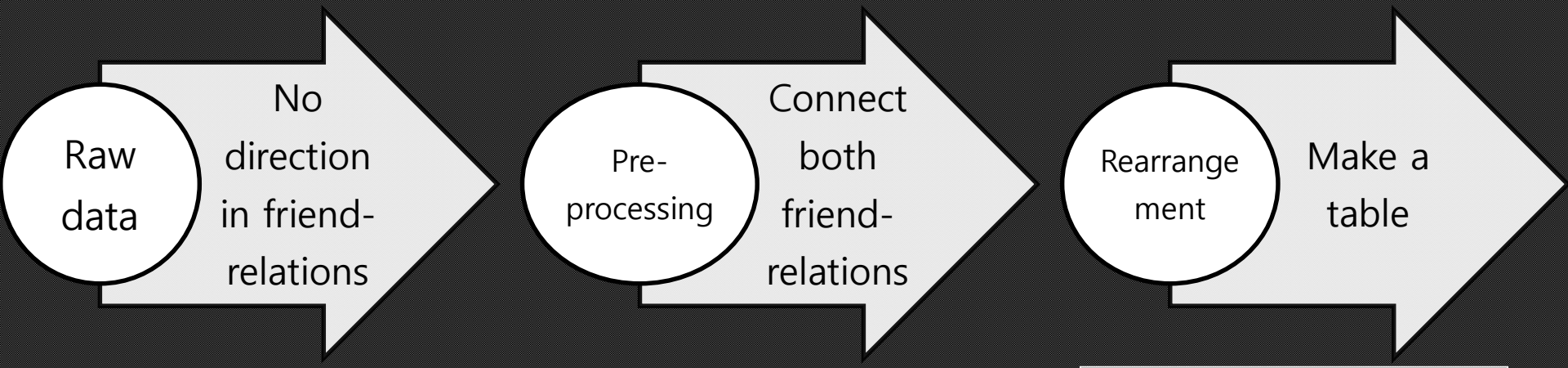
Team 2
2012040089 최윤식
2012048748 유동민
2013042884 서재우
2013042710 김민영
2015041103 이원호

CONTENTS

Method

Method – Inverted Lists

- Pre-processing



ex)

A	B
A	D
A	E
B	C
B	D
B	E
C	E
…	…

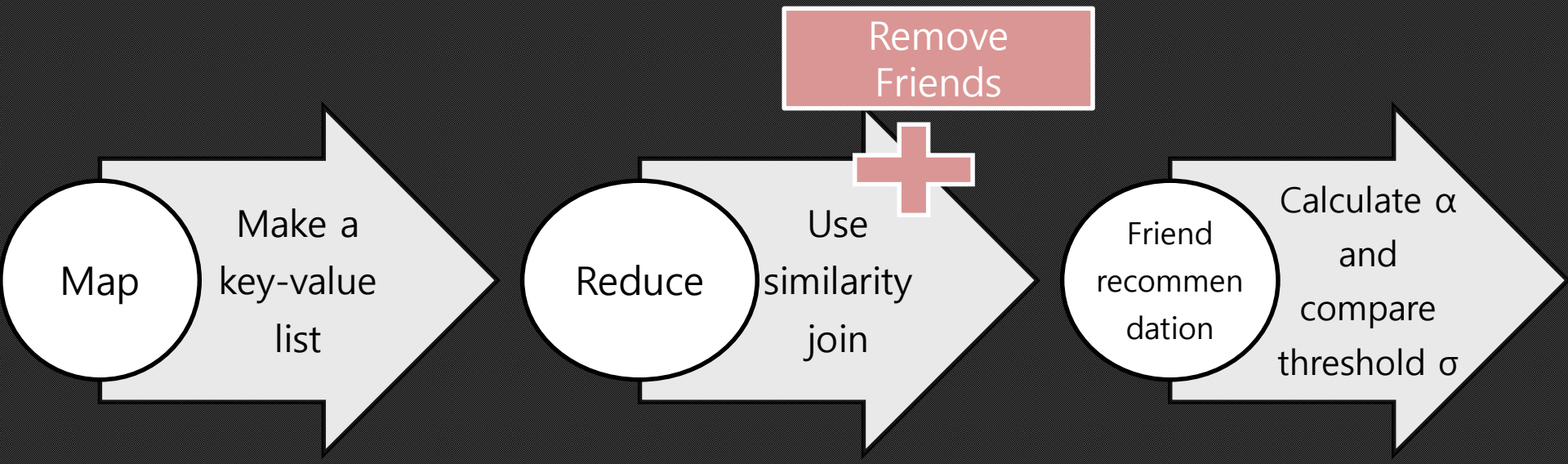
ex)
((A,B), (B,A)),
… …

ex)

A	B, D, E
B	A, C, D, E
C	B, E
D	A, B
E	A, B
…	…

Method – Inverted Lists

- MapReduce



Key	Value
(A,3) , (B,4)	1
(A,3) , (D,2)	1
(A,3) , (E,2)	1
…	…

Key	Overlap
(A,3) , (B,4)	3
(A,3) , (D,2)	2
(A,3) , (E,2)	5
…	…

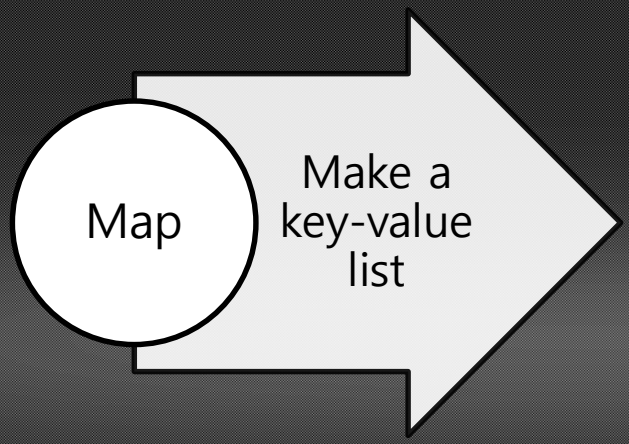
SubtractByKey

Similar pair
(A, C)
(D, E)
…
…

Method – Inverted Lists

➢ MapReduce

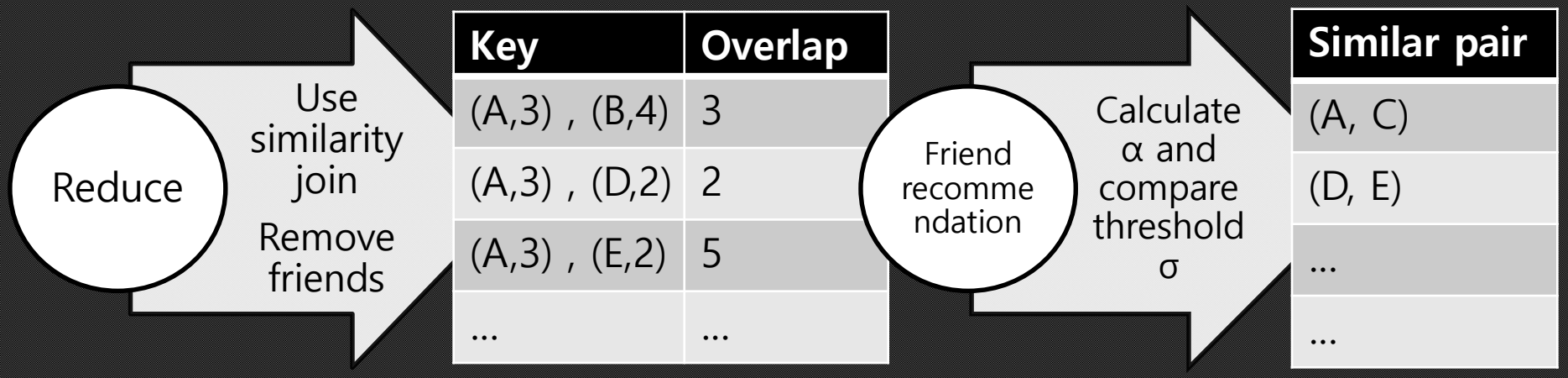
```
// Map
JavaPairRDD<Tuple2<Tuple2<Integer, Integer>, Tuple2<Integer, Integer>>, Integer> allCandidate = invertedTable.flatMapToPair(p -> {
    ArrayList<Tuple2<Tuple2<Tuple2<Integer, Integer>, Tuple2<Integer, Integer>>, Integer>> result = new ArrayList<>();
    int size = p._2().size();
    for (int i = 0; i < size - 1; ++i) {
        for (int j = i + 1; j < size; ++j) {
            Tuple2<Integer, Integer> key1 = new Tuple2<>(p._2().get(i)._1(), p._2().get(i)._2());
            Tuple2<Integer, Integer> key2 = new Tuple2<>(p._2().get(j)._1(), p._2().get(i)._2());
            result.add(new Tuple2<>(new Tuple2<>(key1, key2), 1));
        }
    }
    return result.iterator();
});
```



Key	Value
(A,3) , (B,4)	1
(A,3) , (D,2)	1
(A,3) , (E,2)	1
…	…

Method – Inverted Lists

➢ MapReduce




```
// Reduce
JavaPairRDD<Tuple2<Tuple2<Integer, Integer>, Tuple2<Integer, Integer>>, Integer> overlapCount = allCandidate.reduceByKey((a, b) -> a + b);
JavaRDD<Tuple2<Integer, Integer>> overlap = overlapCount.filter(g -> {
    int countA = g._1()._1()._2(), countB = g._1()._2()._2();
    double theta = THRESHOLD / (1 + THRESHOLD) * (countA + countB);
    return g._2() >= theta;
}).mapToPair(g -> g._1()._1()._1() > g._1()._2()._1() ?
        new Tuple2<>(g._1()._2()._1(), g._1()._1()._1()) : new Tuple2<>(g._1()._1()._1(), g._1()._2()._1())
    ).subtractByKey(pairs).map(p -> new Tuple2<>(p._1(), p._2()));
JavaRDD<Tuple2<Integer, Integer>> sortedOverlap = overlap.sortBy(new TupleComparator(), true, 2);

for (Tuple2<Integer, Integer> candidate : sortedOverlap.collect()) {
    System.out.println(candidate._1() + "₩t" + candidate._2());
}
```

THANK YOU