Pair RDD

- Spark gives you more transformations if the RDD type is pair RDD.
- Spark can shuffle elements based on the first element of the pair
- Called Key-Value pair RDD
  - First element in pair is called “key”
  - Second element in pair is called “value”

Create Pair RDD

```
pairs = sc.range(1, 100).map(lambda n: (n, 1))
```

```
val pairs = sc.range(1, 100).map(_ -> 1)
```

```
JavaRDD<Integer> numbers =
  sc.parallelize(Arrays.asList(1, 2, 3, 4, 5));

JavaPairRDD<Integer, Integer> pairs =
  rdd.mapToPair(n -> new Tuple2(n, 1))
```

- In Java we use Scala’s Tuple2

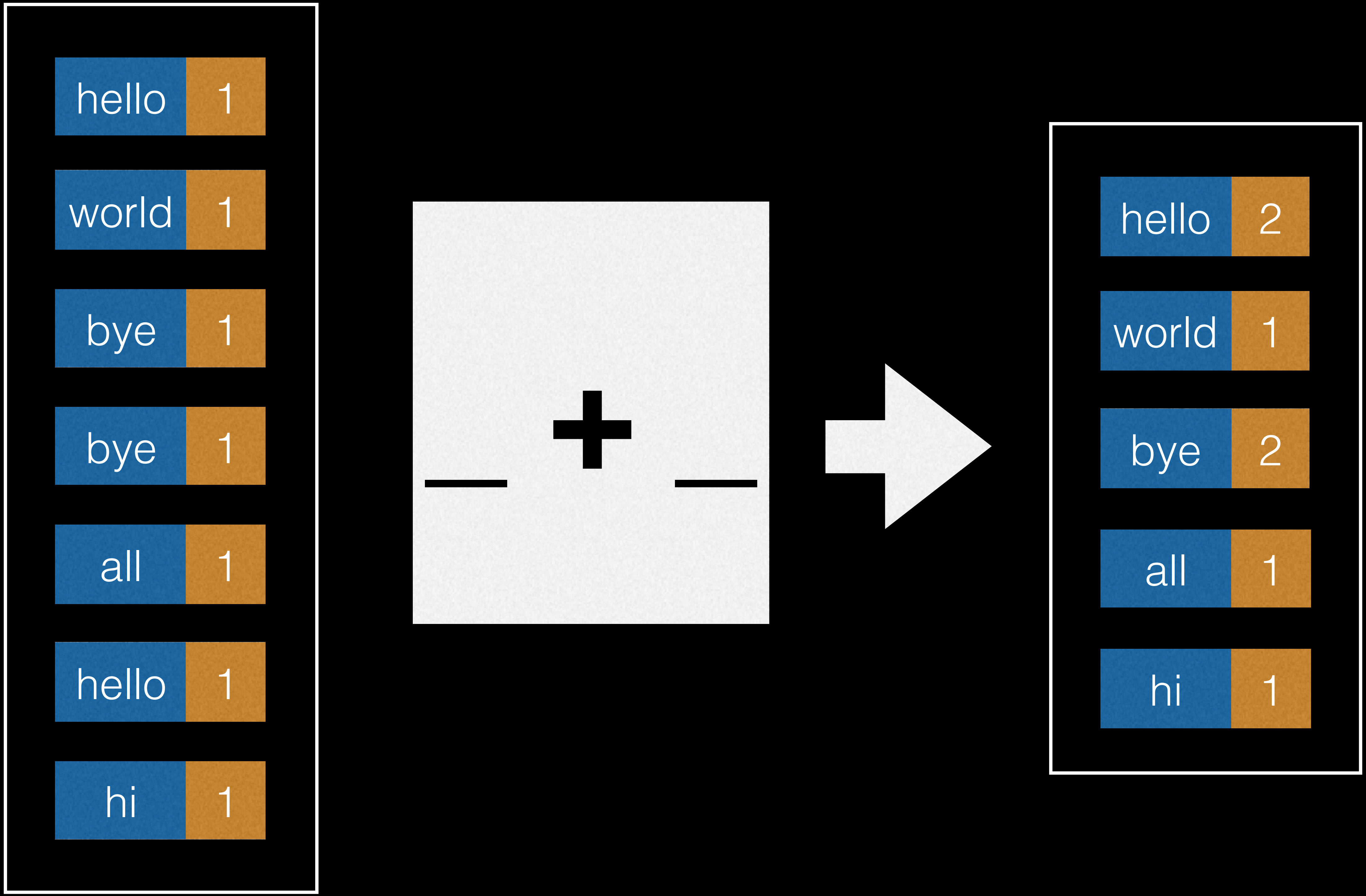
Transformations

- reduceByKey
- groupByKey
- mapValues
- keys
- values

reduceByKey

- Combines values that have same key
- Runs reduce on each group alone
- Returns RDD consisting of each key and the reduced value for that key.

reduceByKey

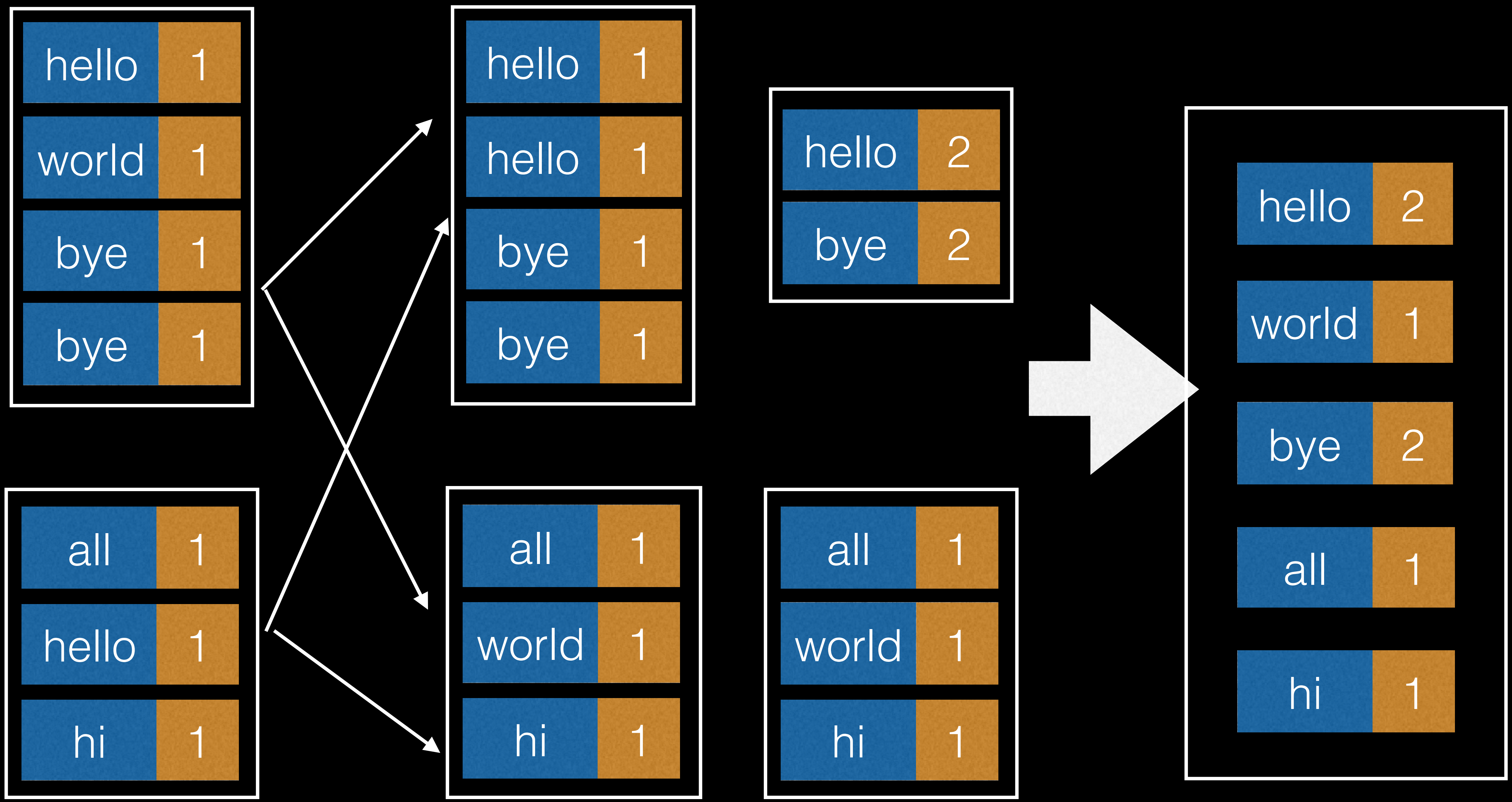


hello	1
world	1
bye	1
bye	1
all	1
hello	1
hi	1

_ + _

hello	2
world	1
bye	2
all	1
hi	1

reduceByKey



hello	1
world	1
bye	1
bye	1

all	1
hello	1
hi	1

hello	1
hello	1
bye	1
bye	1

all	1
world	1
hi	1

hello	2
bye	2

all	1
world	1
hi	1

hello	2
world	1
bye	2
all	1
hi	1

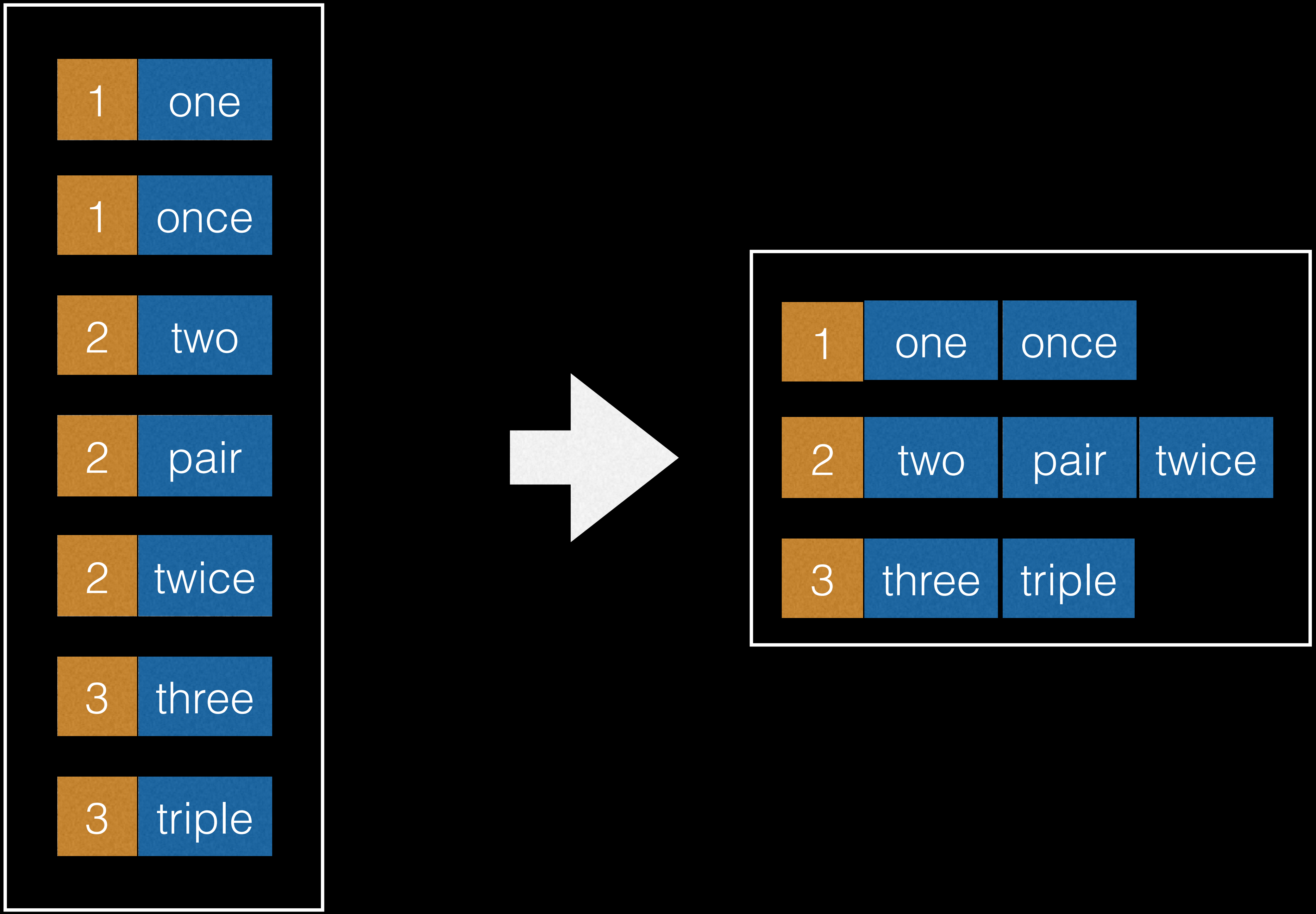
Word Count

- Let’s implement word count program
- 1. We need to read a text file
- 2. Generate RDD of words
- 3. Count occurrences of each word

groupByKey

- Groups values that have same key
- Returns RDD consisting of each key and list of values.

groupByKey



1	one
1	once
2	two
2	pair
2	twice
3	three
3	triple

1	one	once	
2	two	pair	twice
3	three	triple	

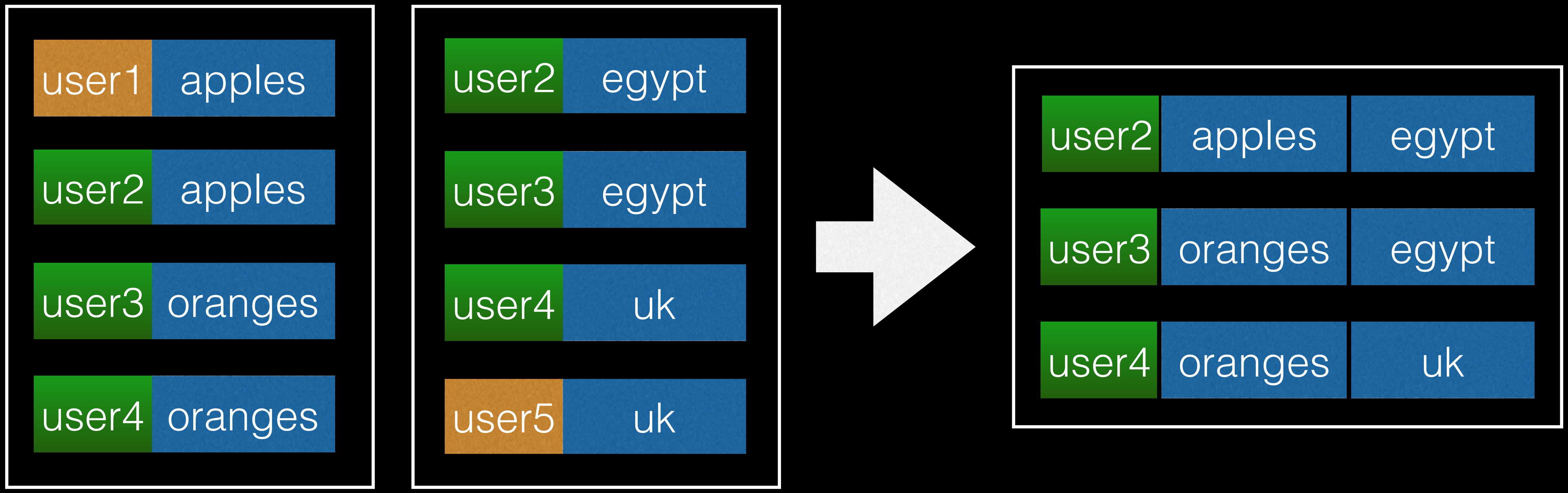
mapValues

- Applies a function to each value of RDD
- The key stays the same
- Better than “map”, as it allows Spark to know that key is the same (No need to reshuffle)

Transformations

- subtractByKey
- join
- rightOuterJoin
- leftOuterJoin
- cogroup

join

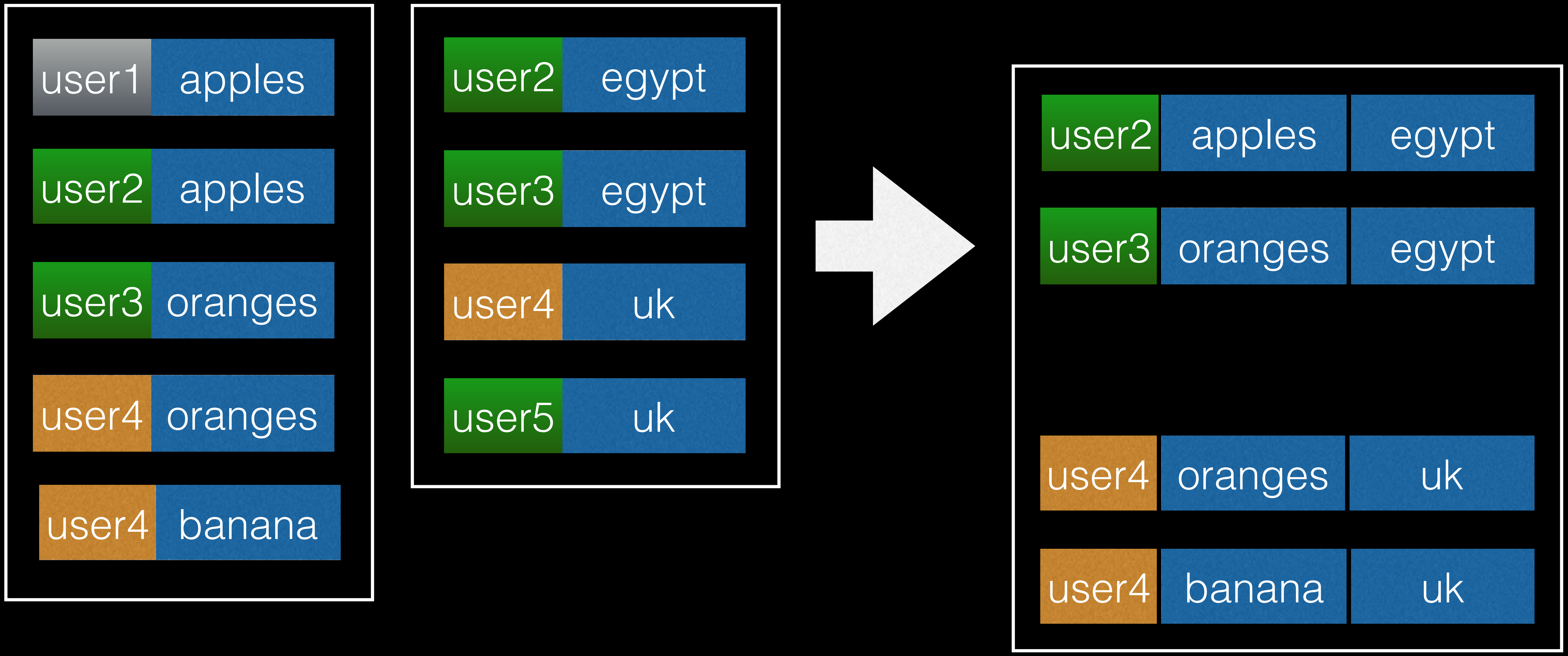


user1	apples
user2	apples
user3	oranges
user4	oranges

user2	egypt
user3	egypt
user4	uk
user5	uk

user2	apples	egypt
user3	oranges	egypt
user4	oranges	uk

join

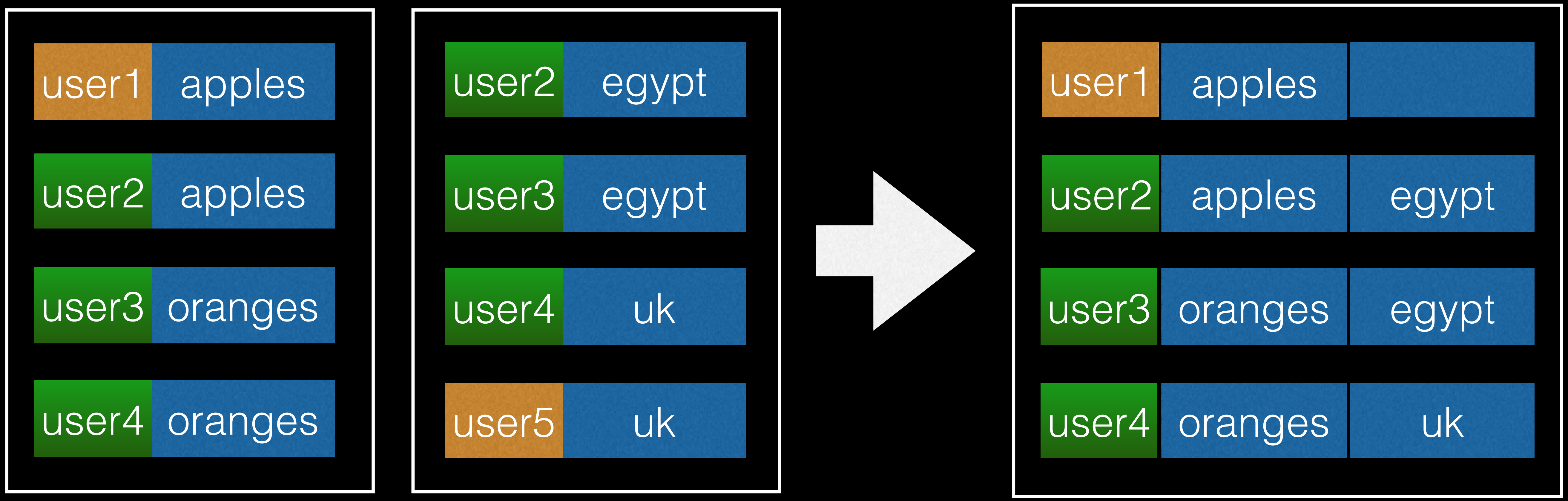


user1	apples
user2	apples
user3	oranges
user4	oranges
user4	banana

user2	egypt
user3	egypt
user4	uk
user5	uk

user2	apples	egypt
user3	oranges	egypt
user4	oranges	uk
user4	banana	uk

leftOuterJoin

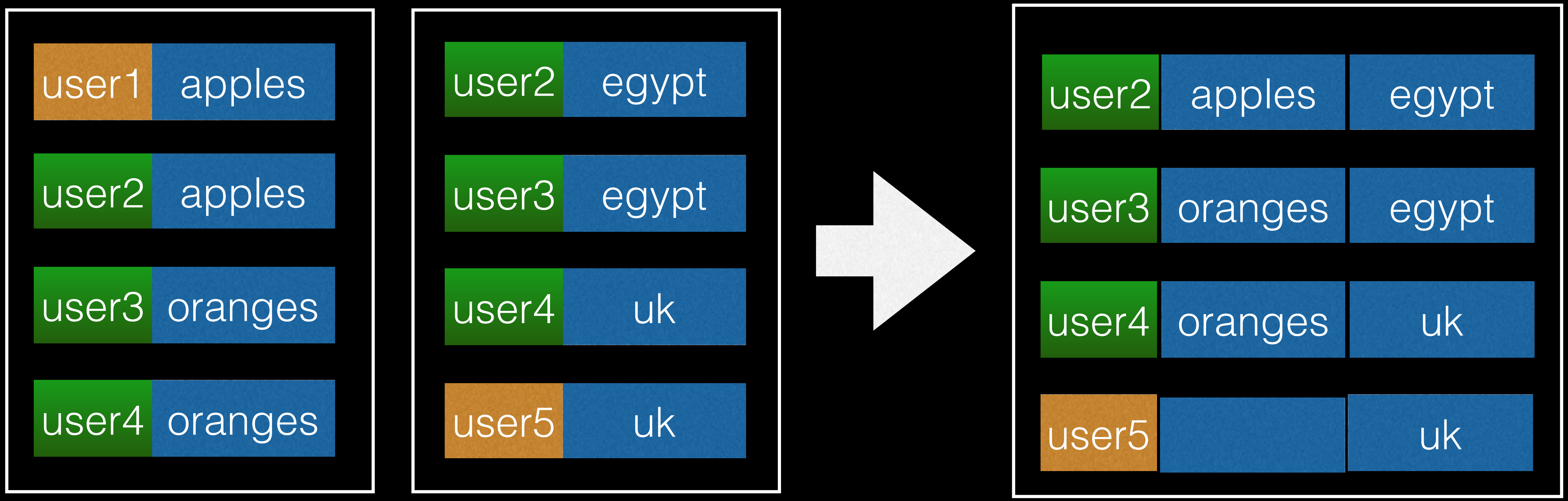


user1	apples
user2	apples
user3	oranges
user4	oranges

user2	egypt
user3	egypt
user4	uk
user5	uk

user1	apples	
user2	apples	egypt
user3	oranges	egypt
user4	oranges	uk

rightOuterJoin

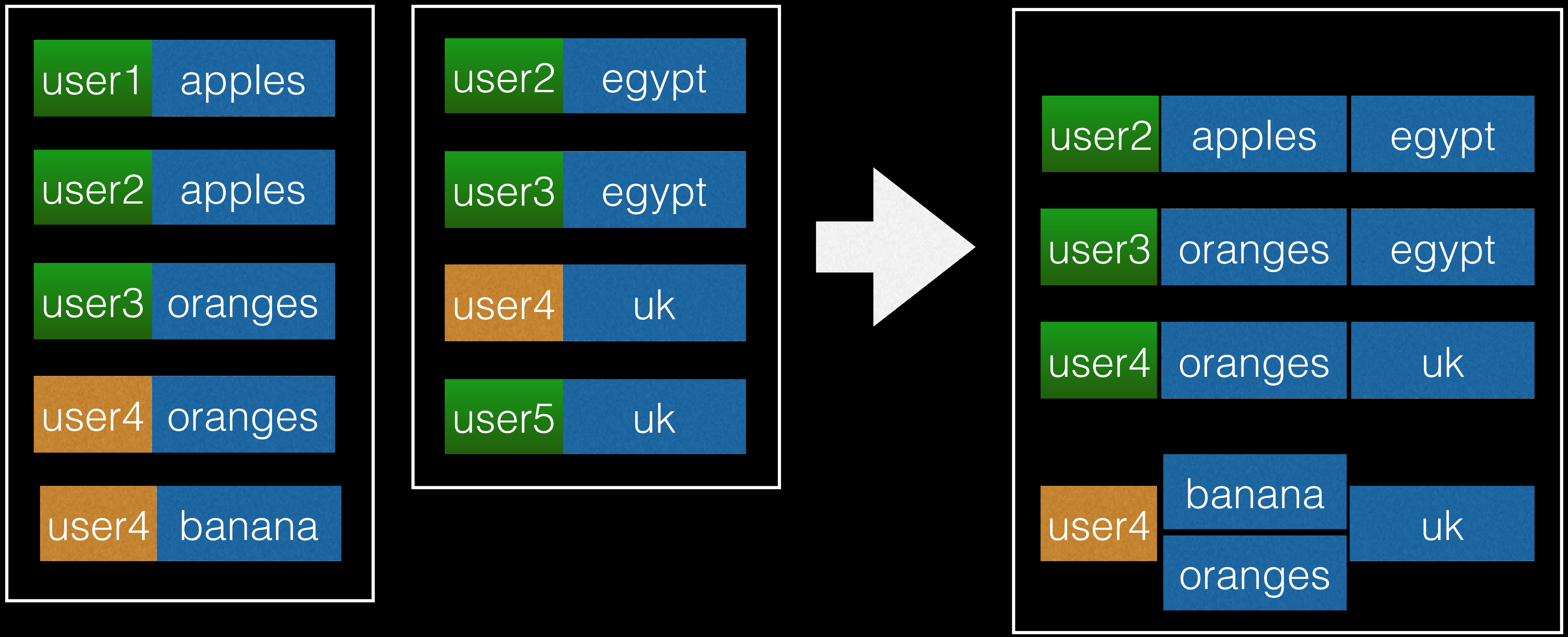


user1	apples
user2	apples
user3	oranges
user4	oranges

user2	egypt
user3	egypt
user4	uk
user5	uk

user2	apples	egypt
user3	oranges	egypt
user4	oranges	uk
user5		uk

cogroup



user1	apples
user2	apples
user3	oranges
user4	oranges
user4	banana

user2	egypt
user3	egypt
user4	uk
user5	uk

user2	apples	egypt
user3	oranges	egypt
user4	oranges	uk
user4	banana oranges	uk

Actions

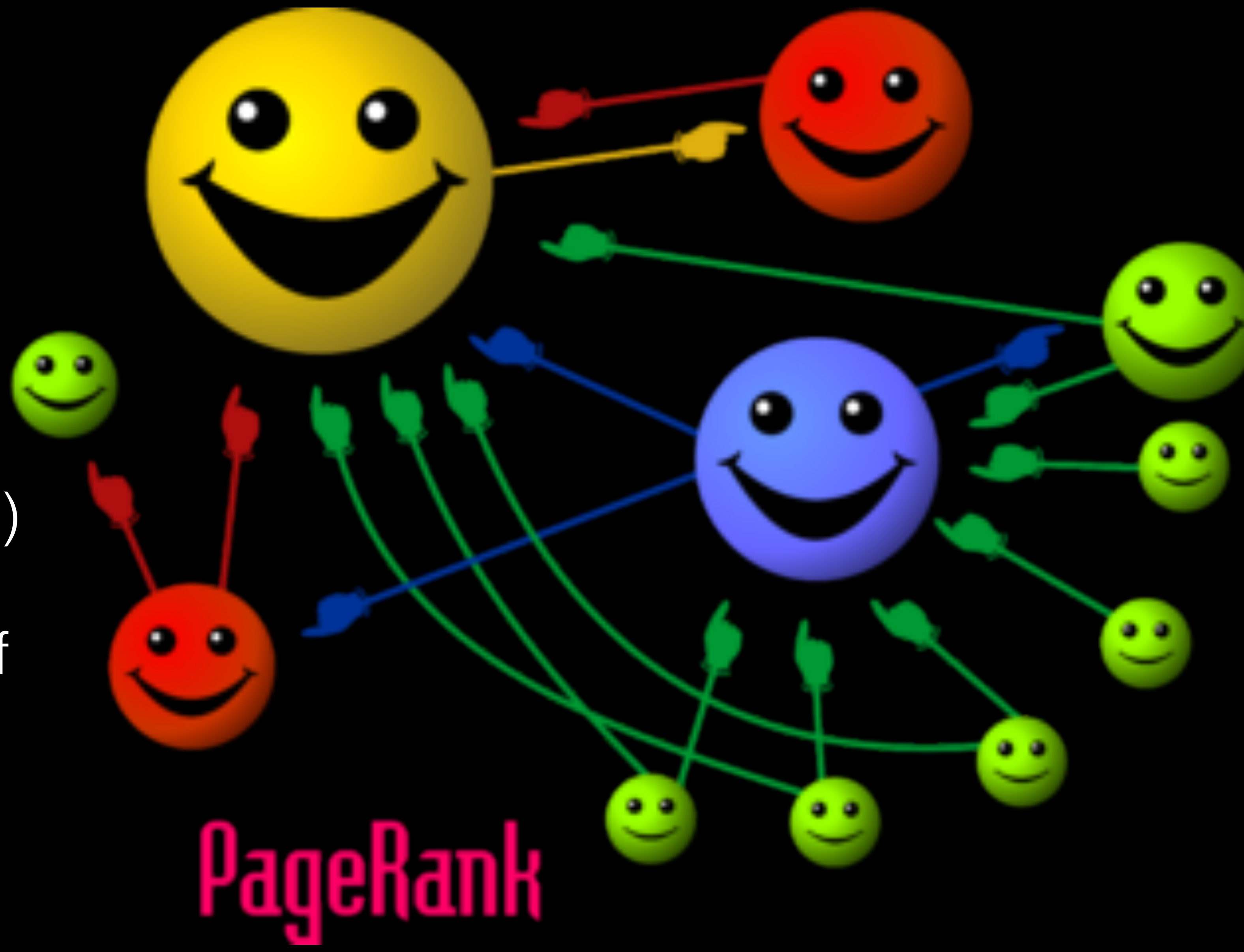
- countByKey
- collectAsMap
- lookup(key)

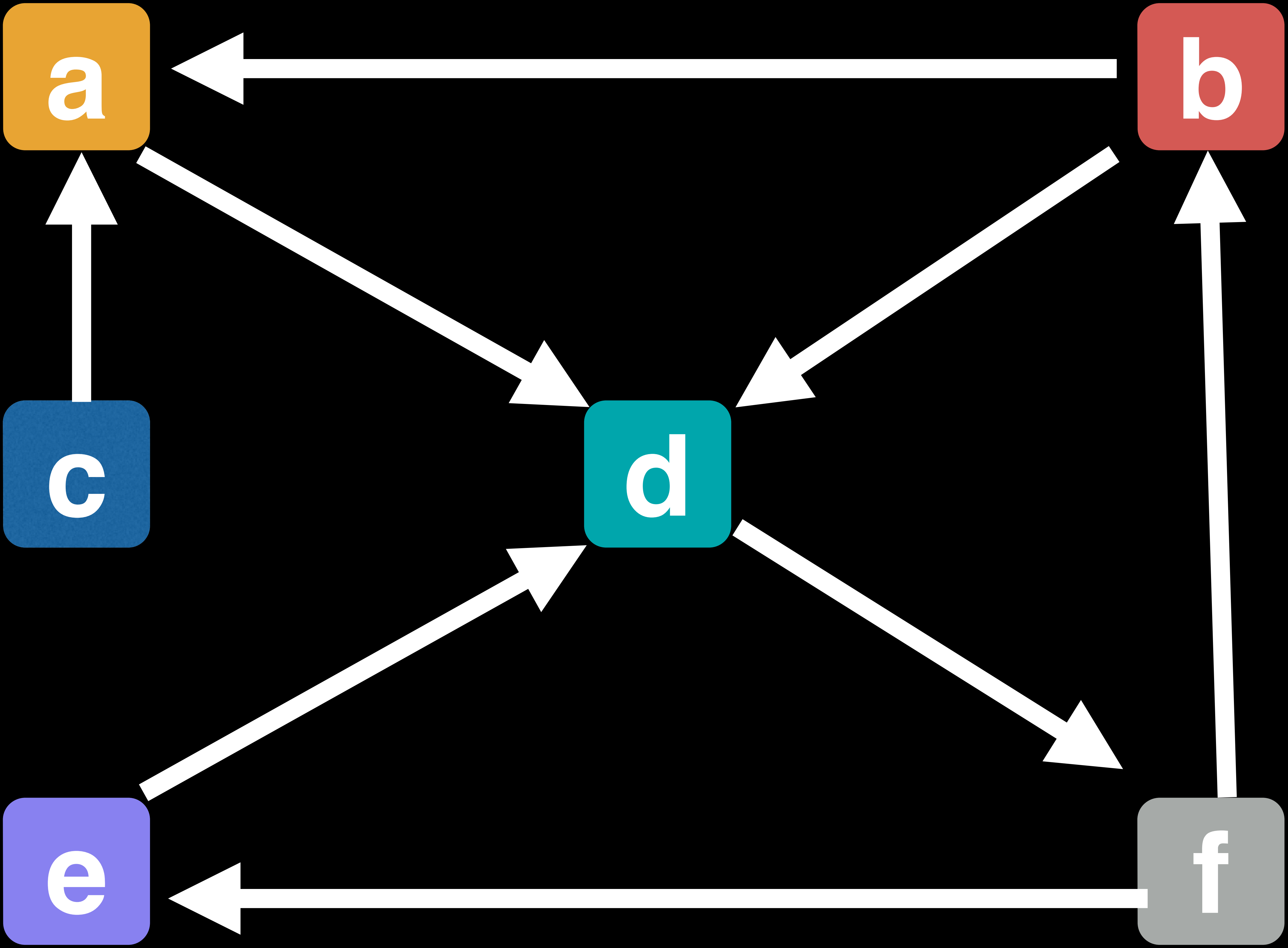
Page Rank

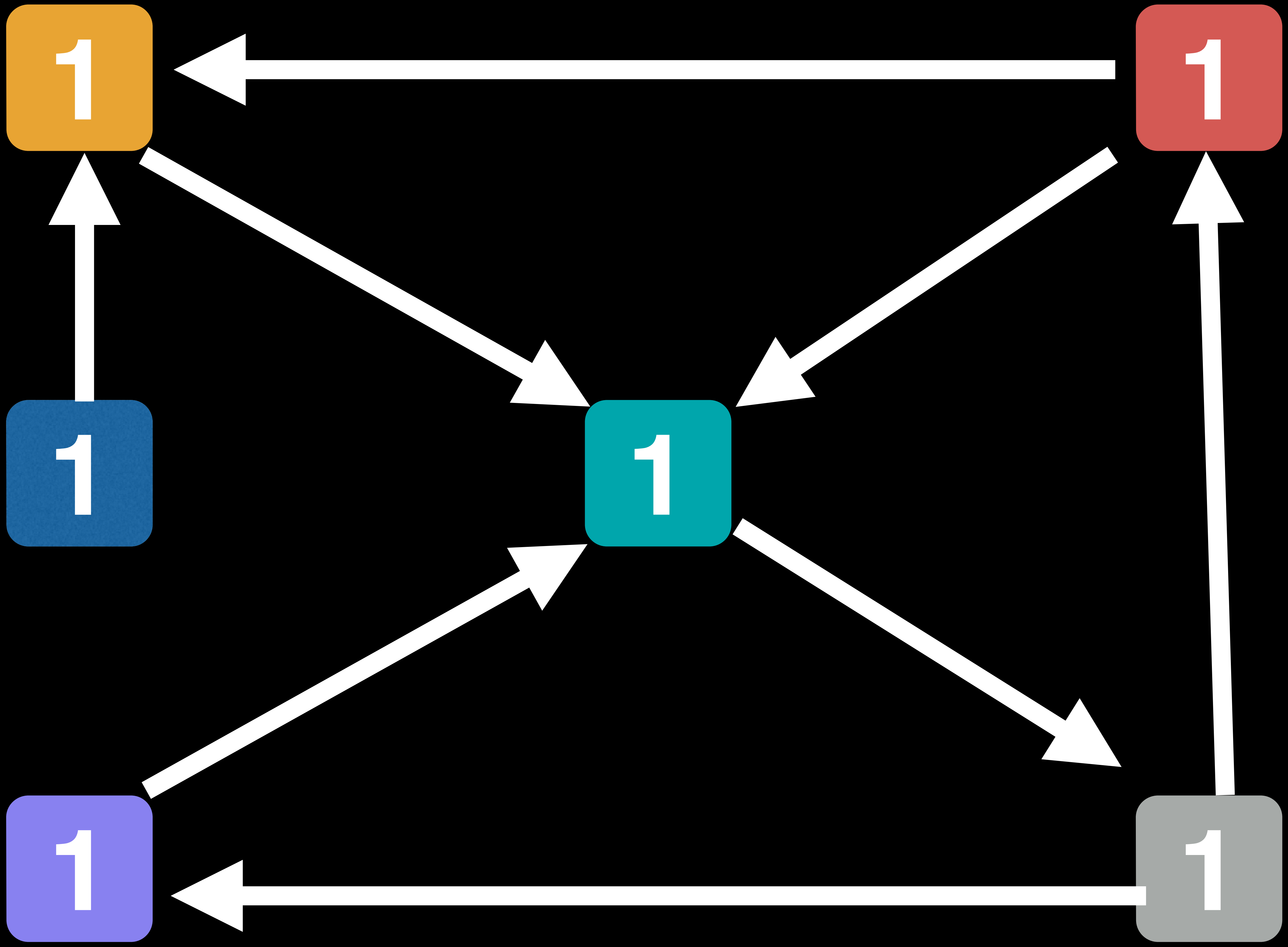
- Named after Larry Page

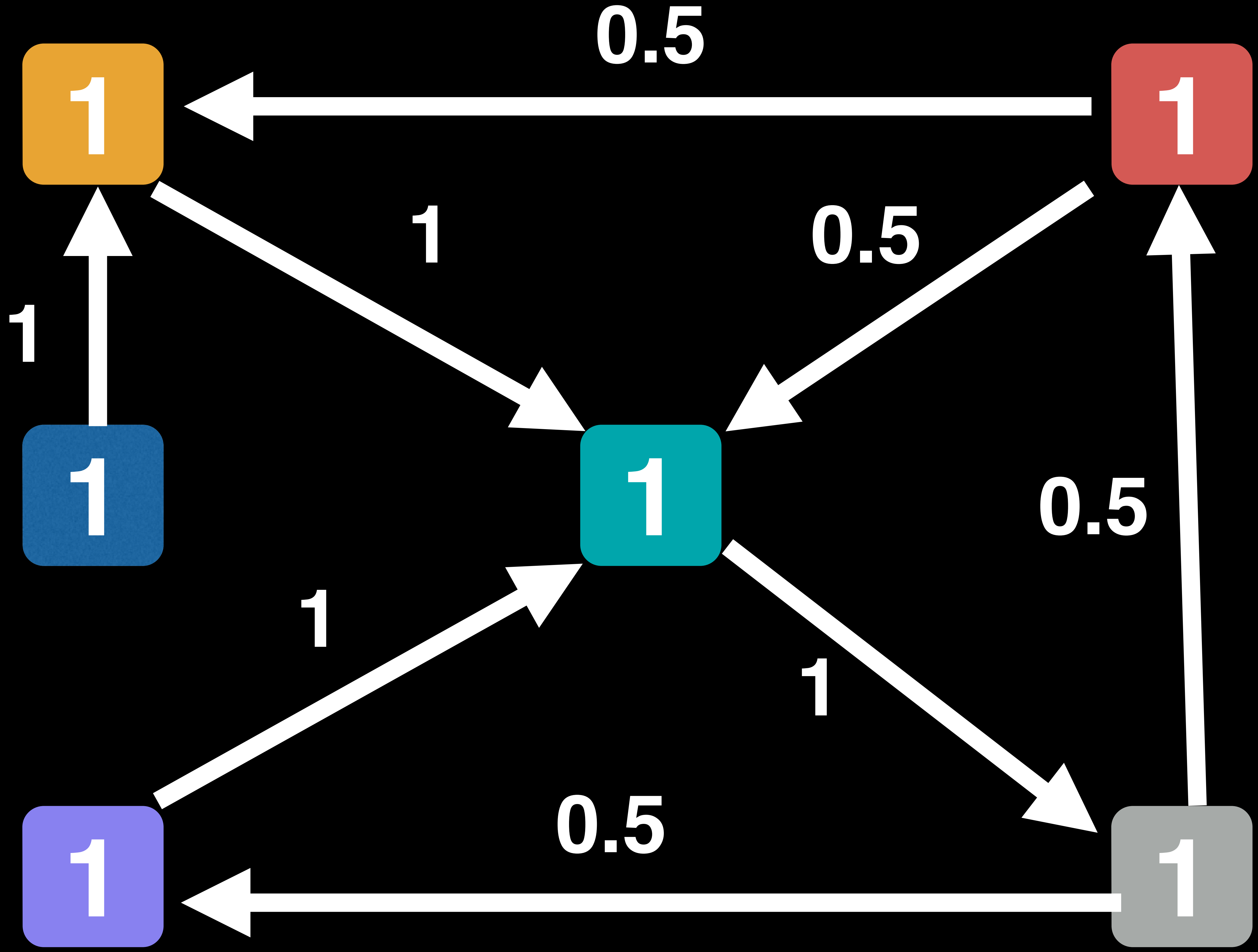
- assign a rank for each document based how many other documents links to it (and rank of those documents)

- Can also measure influence of users on social networks

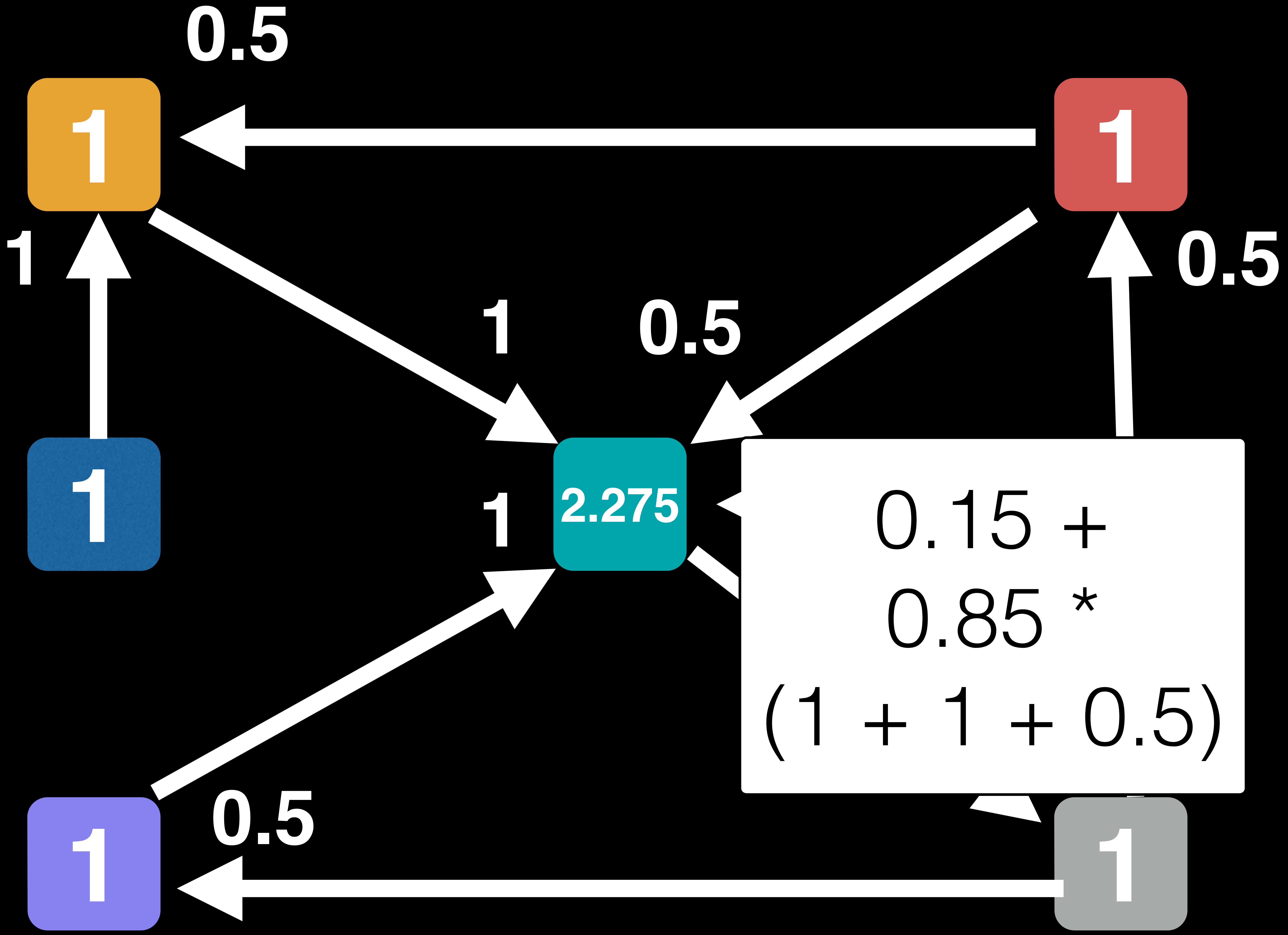



a
b
c
d
e
f

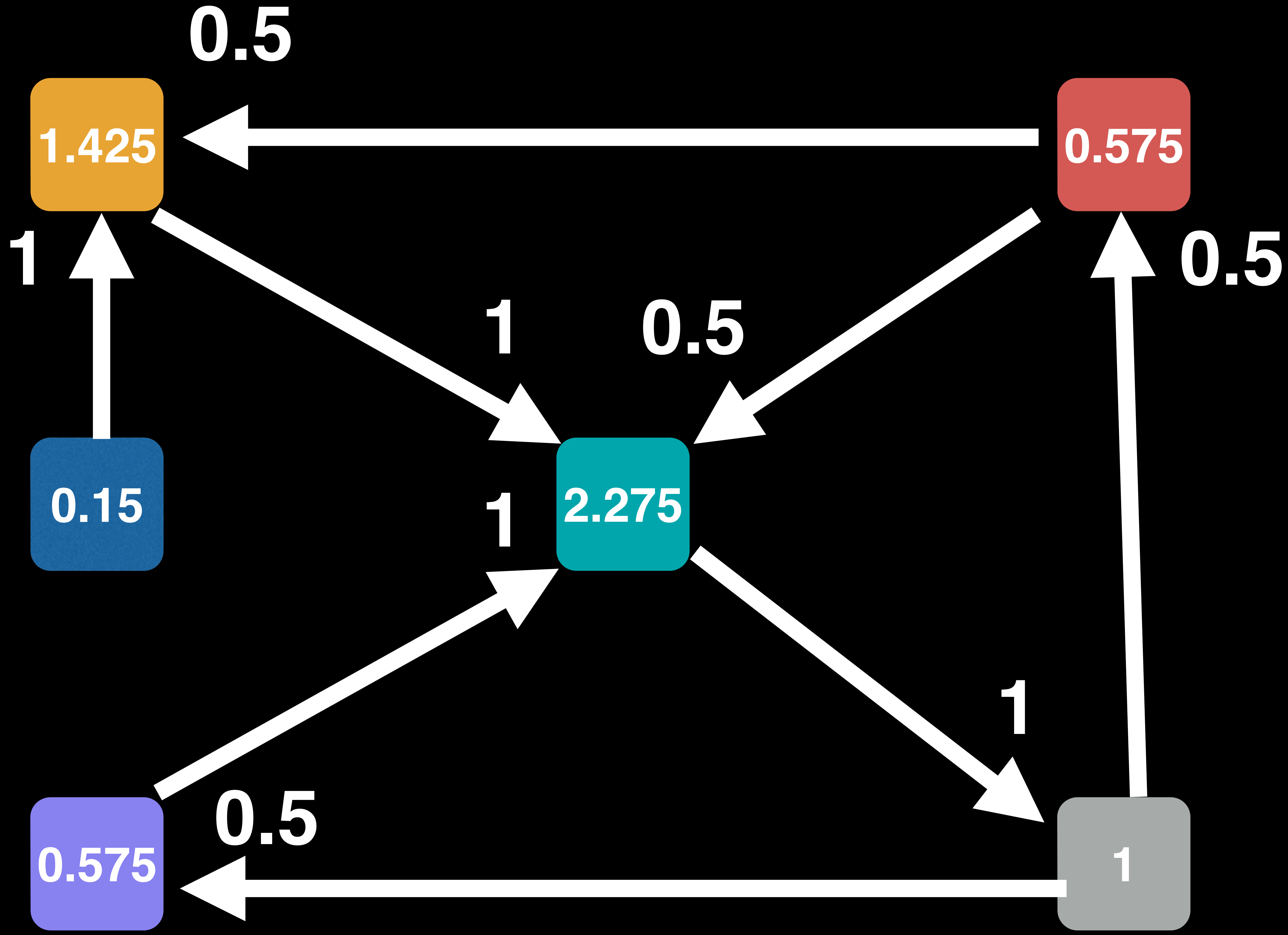

1
1
1
1
1
1

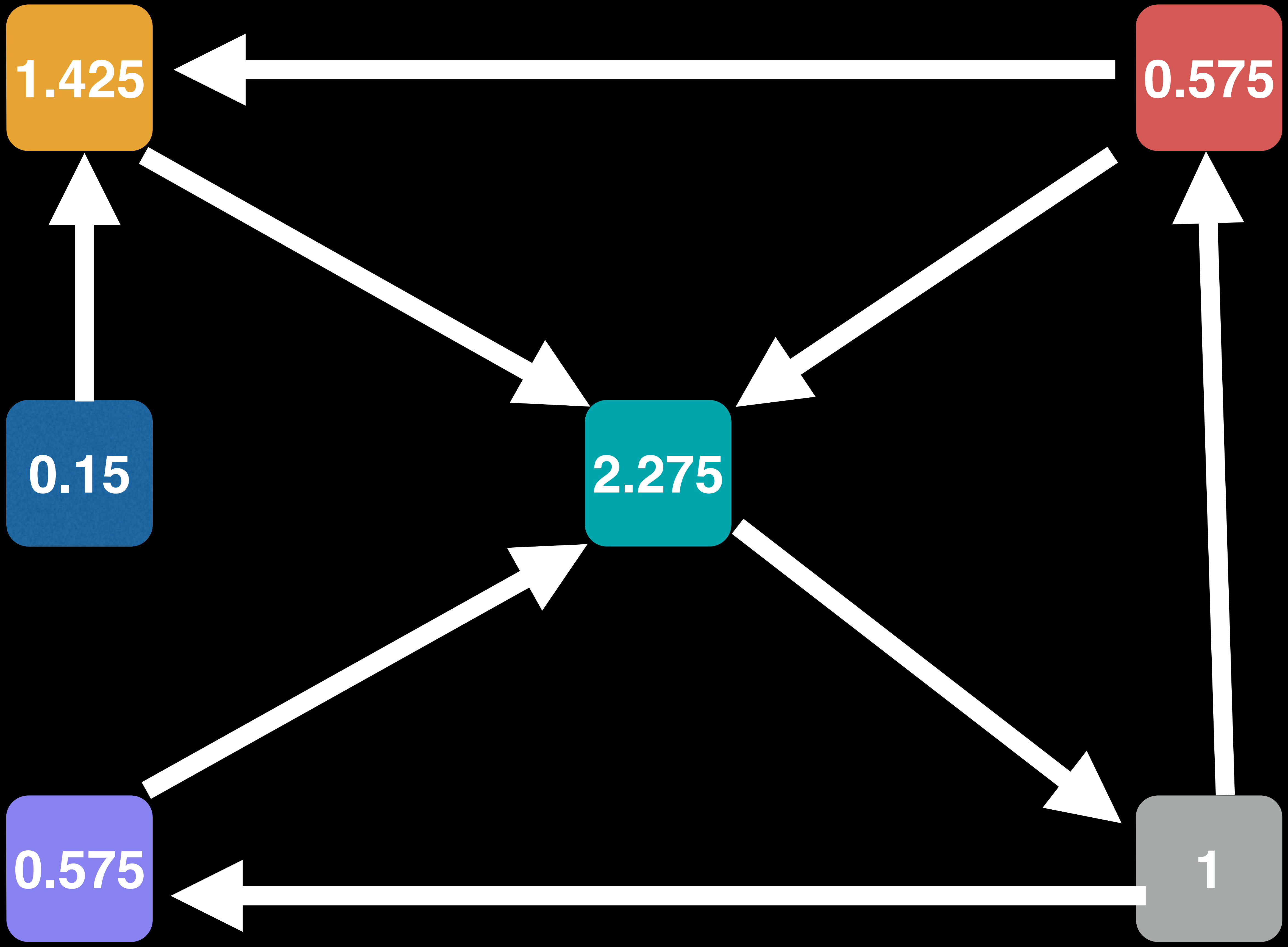


0.5
1
1
0.5
1
1
1
0.5
1
1
0.5
1
1



0.15 +
0.85 *
(1 + 1 + 0.5)


0.5
1.425
0.575
1
0.5
1
0.5
0.15
1
2.275
1
0.5
0.575
1


1.425
0.575
0.15
2.275
0.575
1