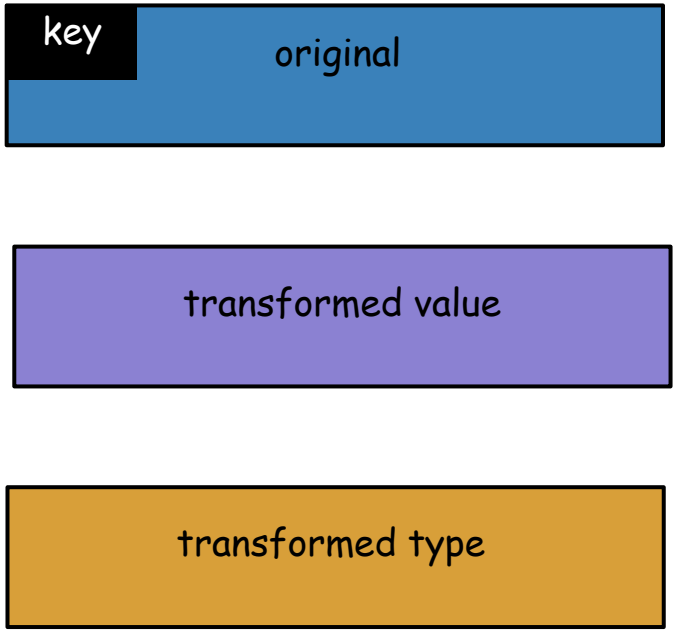
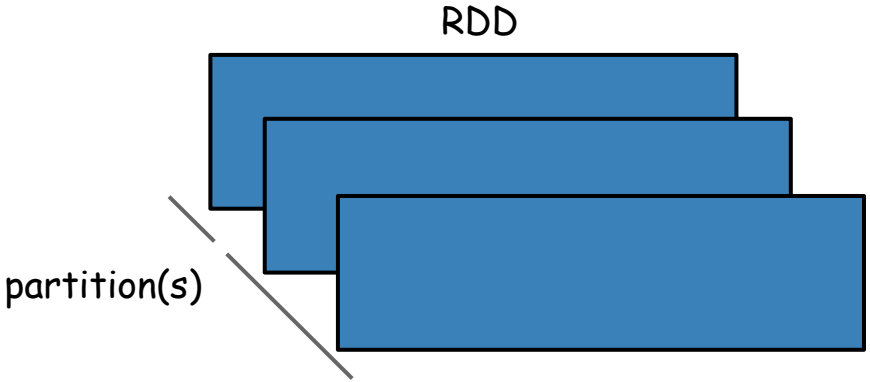
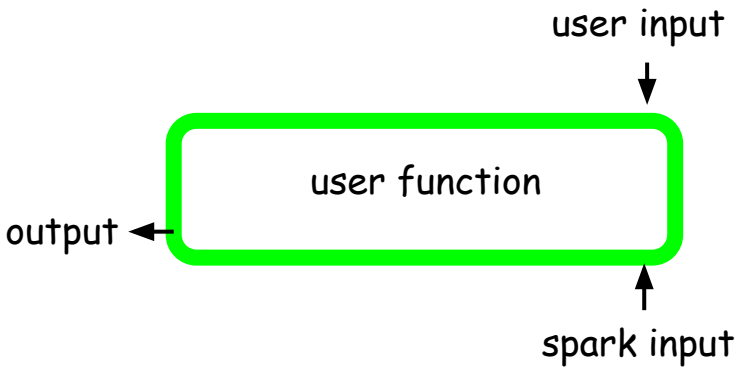
pySpark pictures

legend

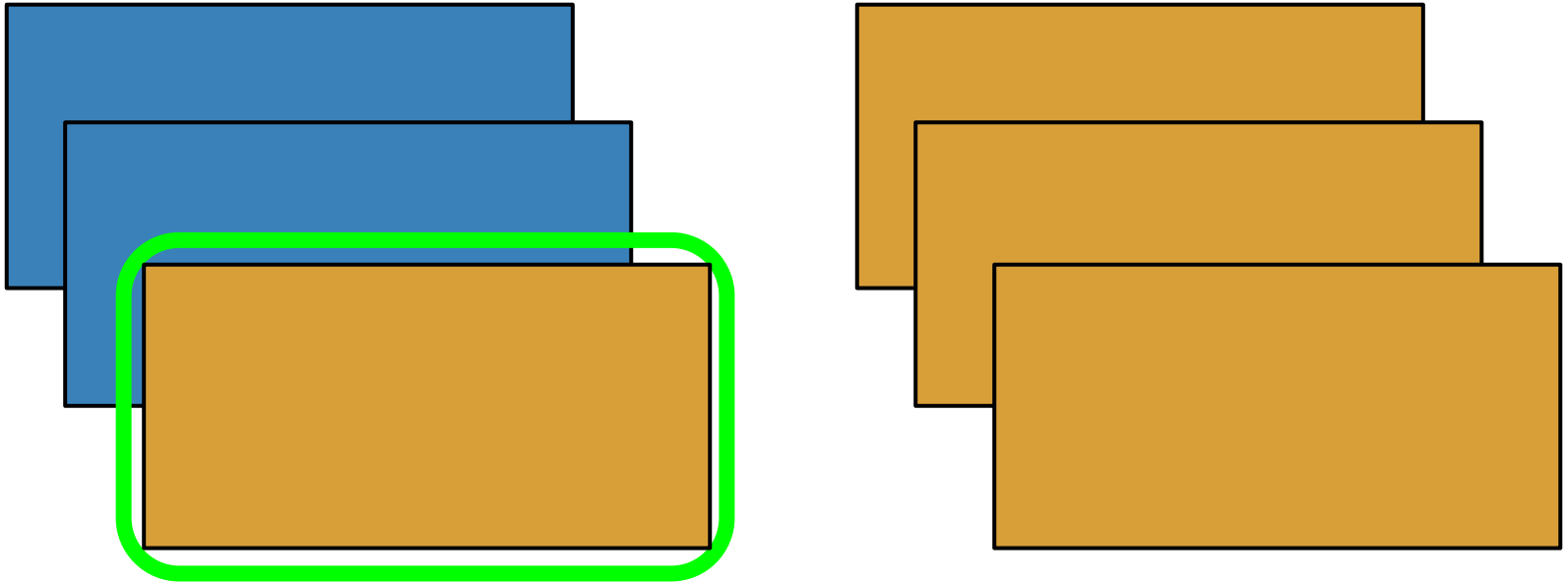
RDD Elements



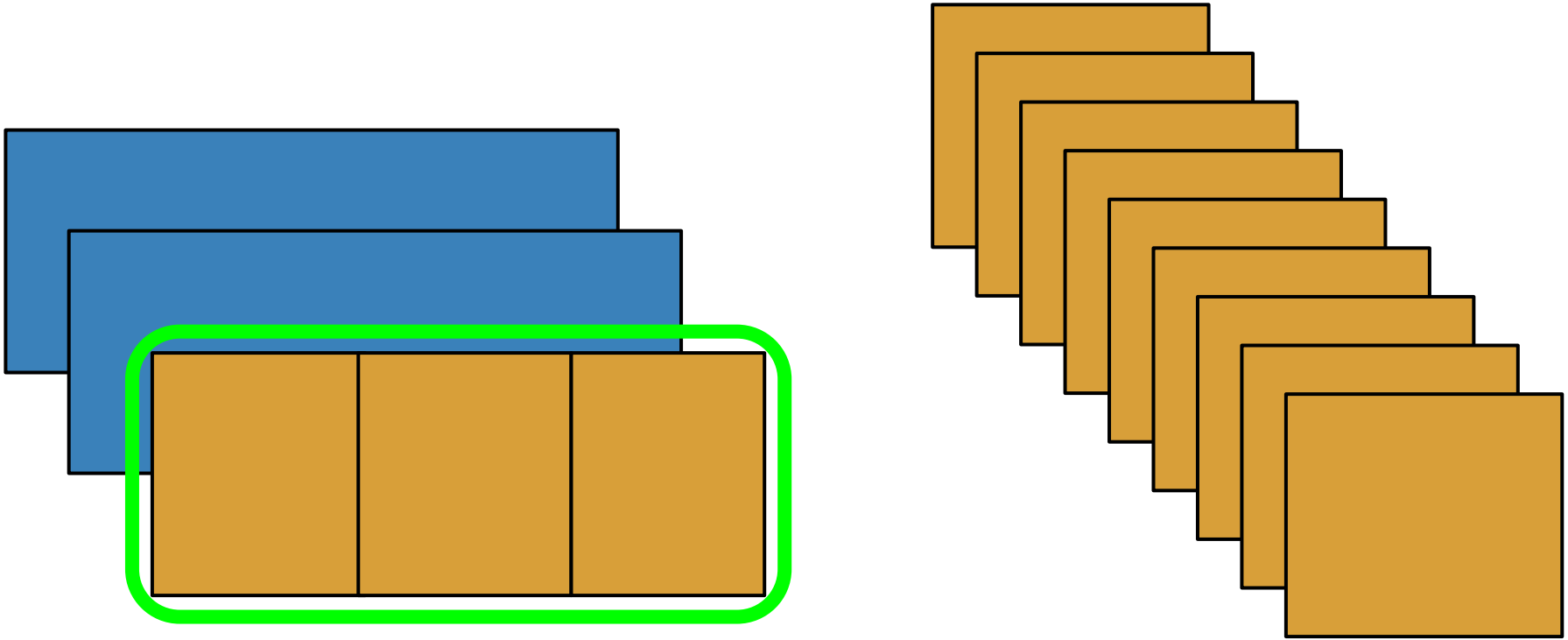




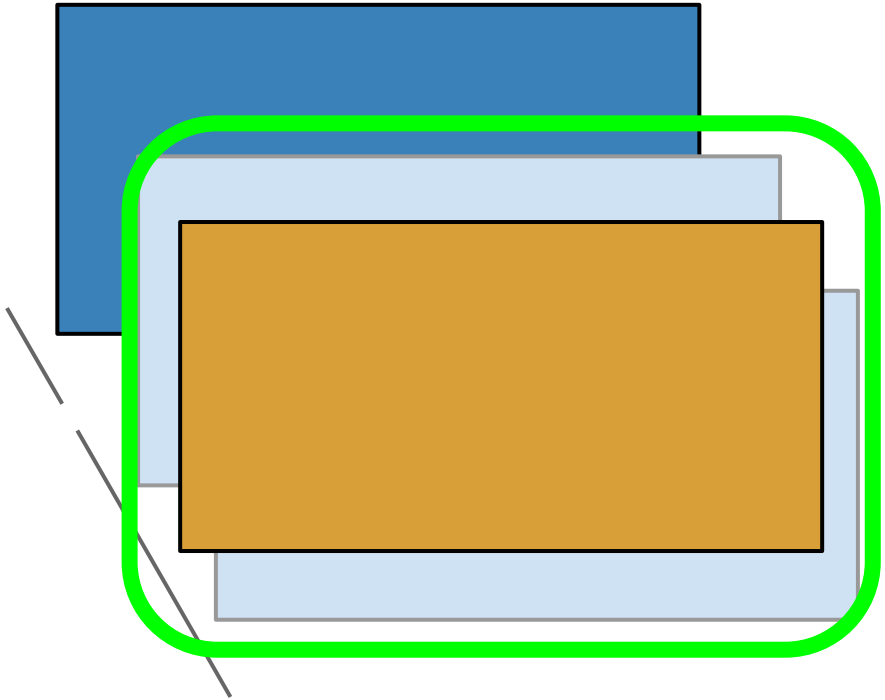
map

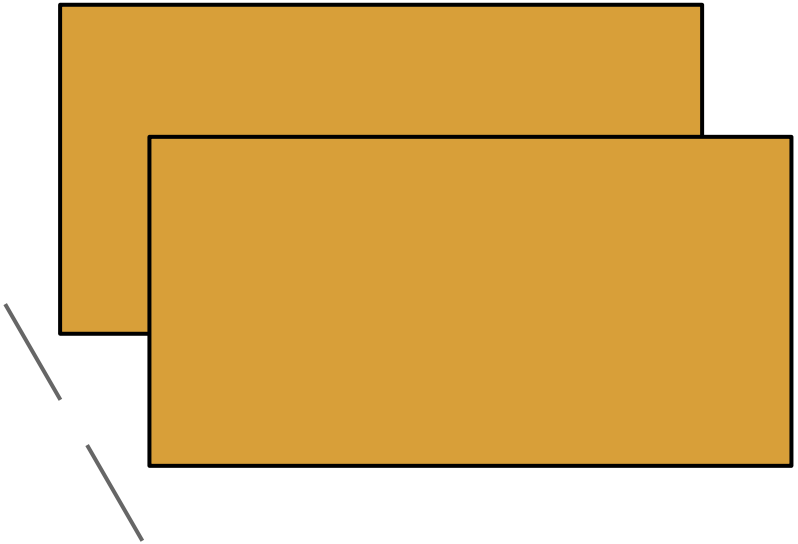


flatMap

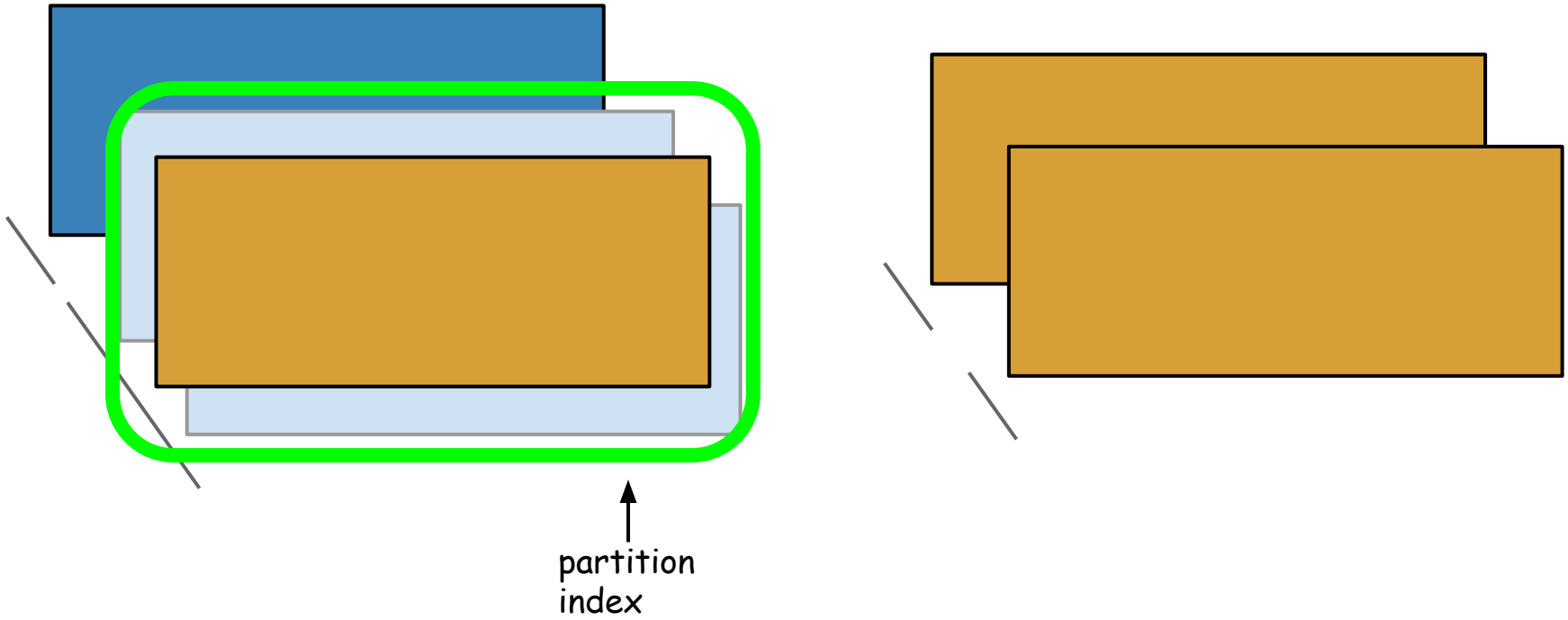


mapPartitions

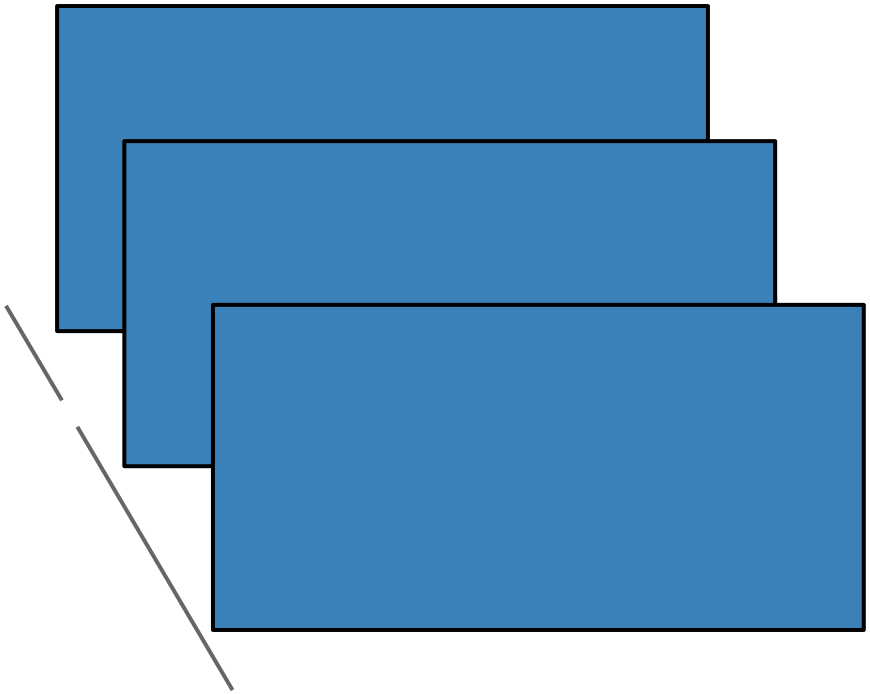




mapPartitionsWithIndex

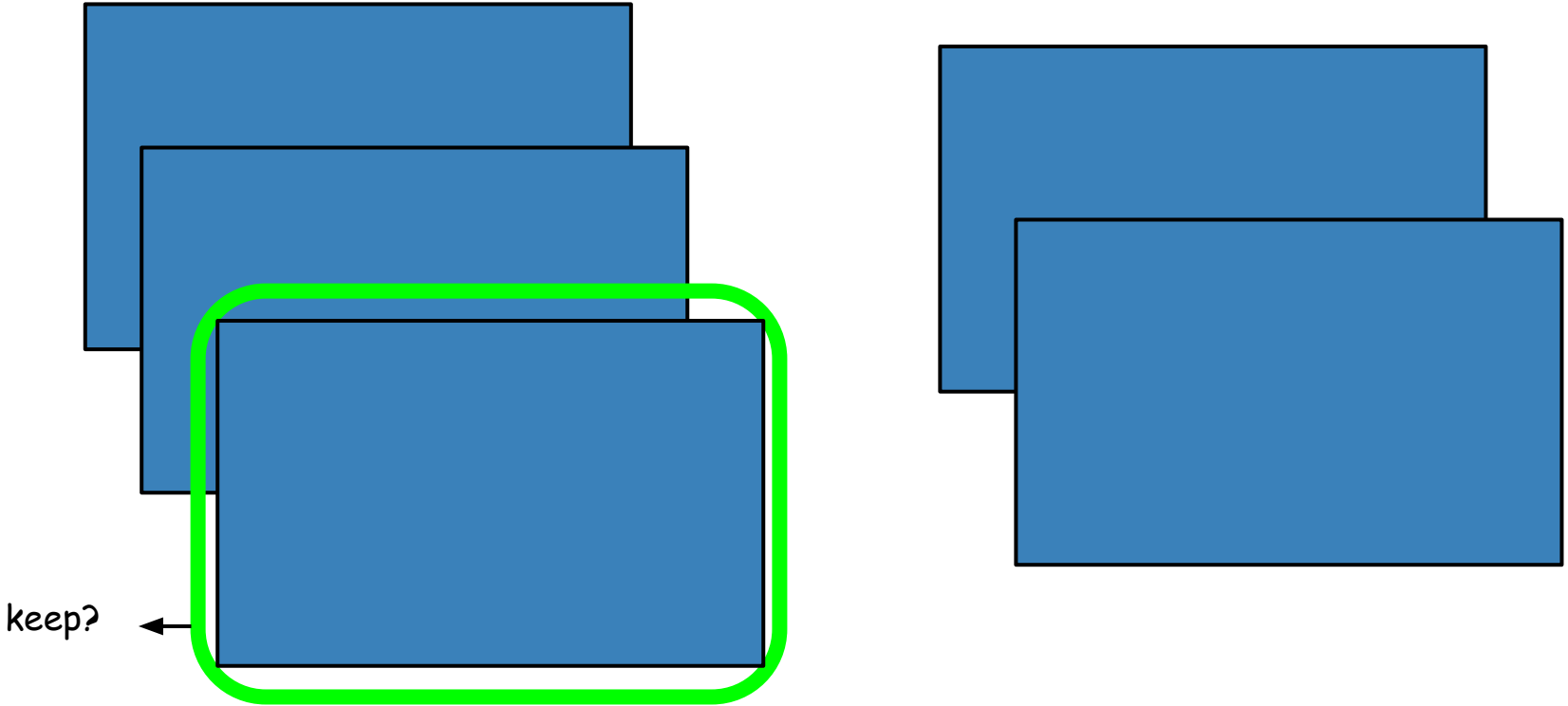


getNumPartitions

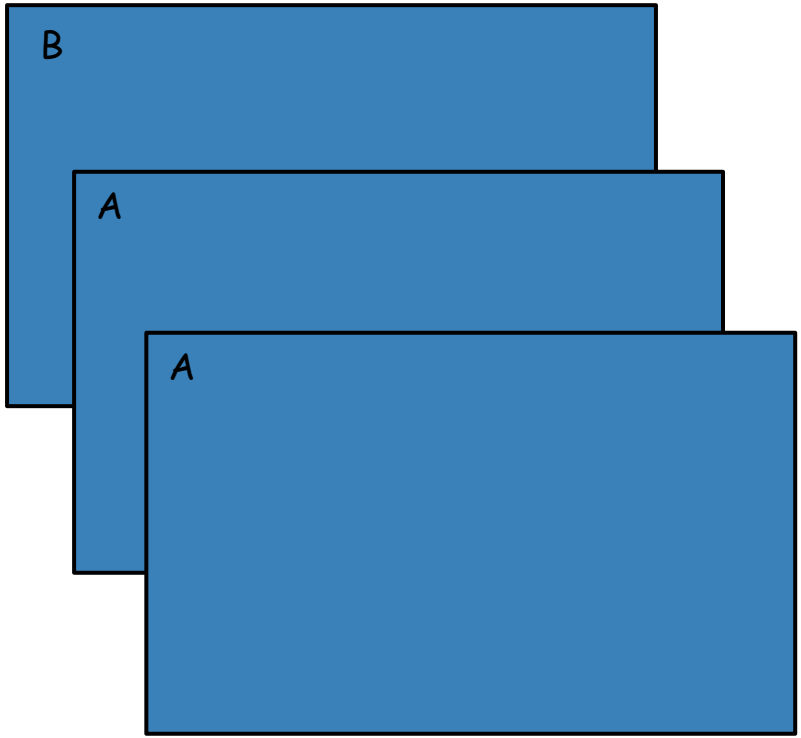


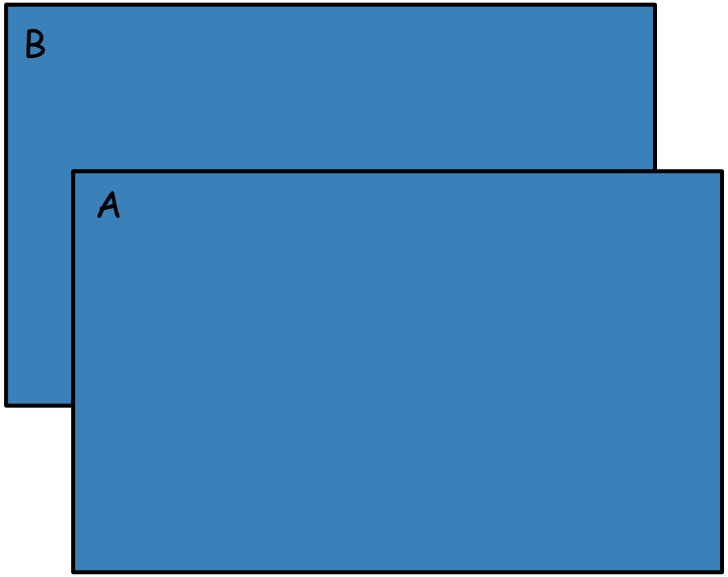
2

filter

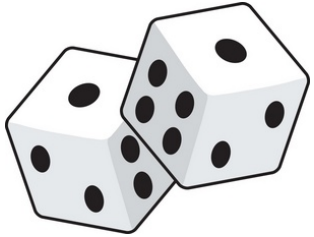


distinct



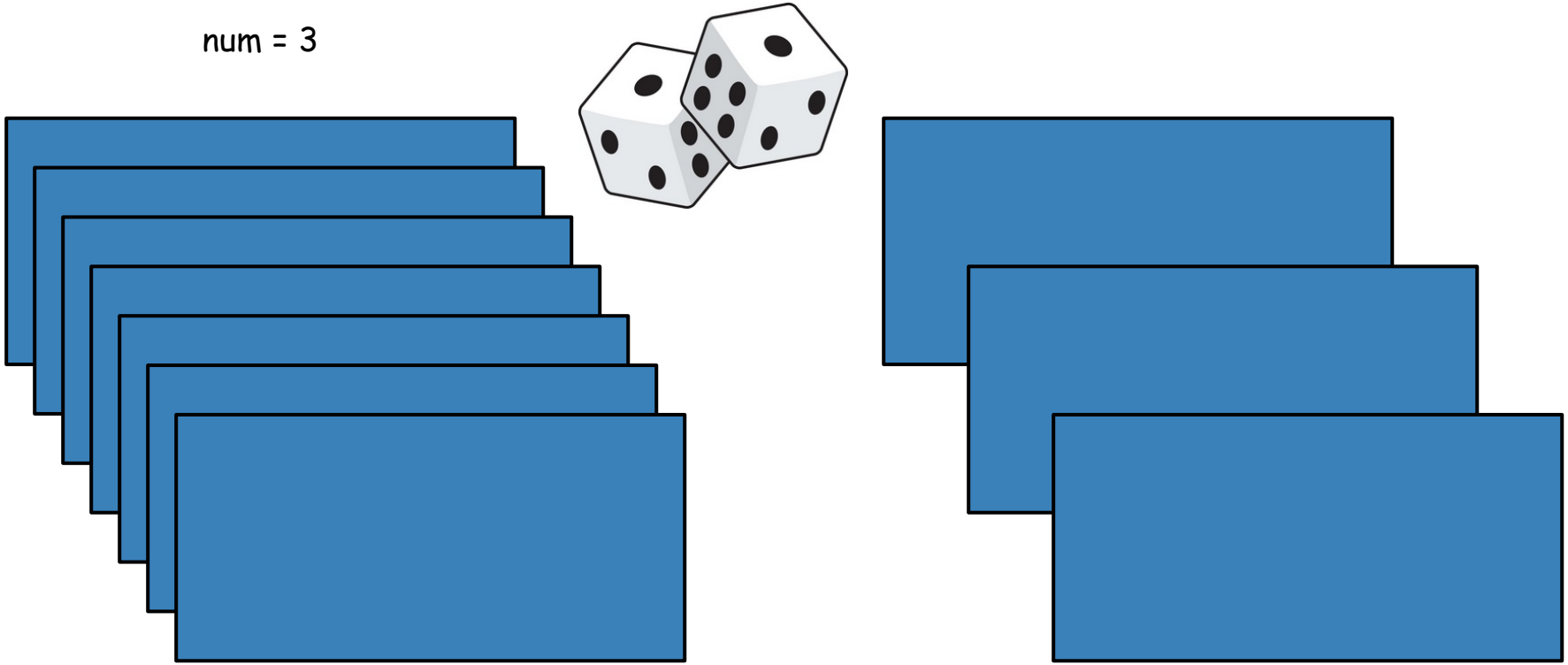


sample

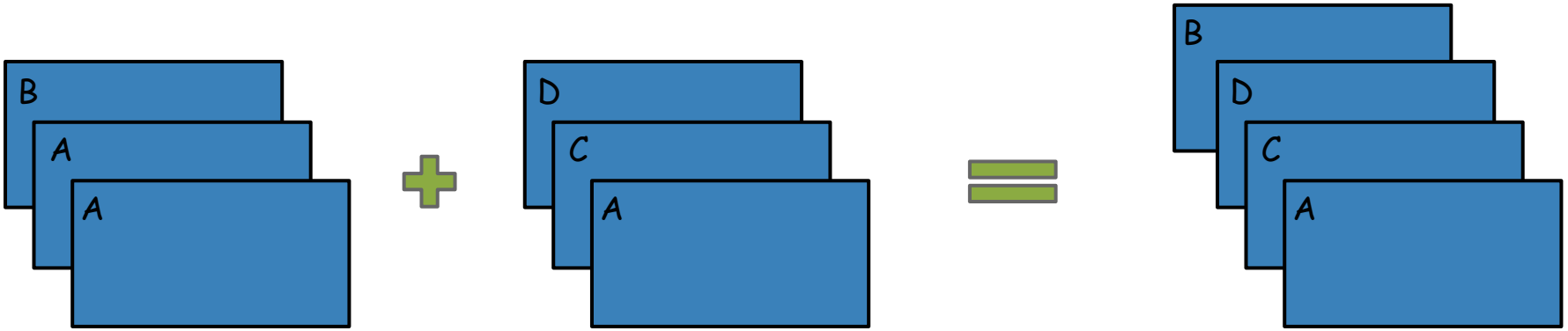


takeSample

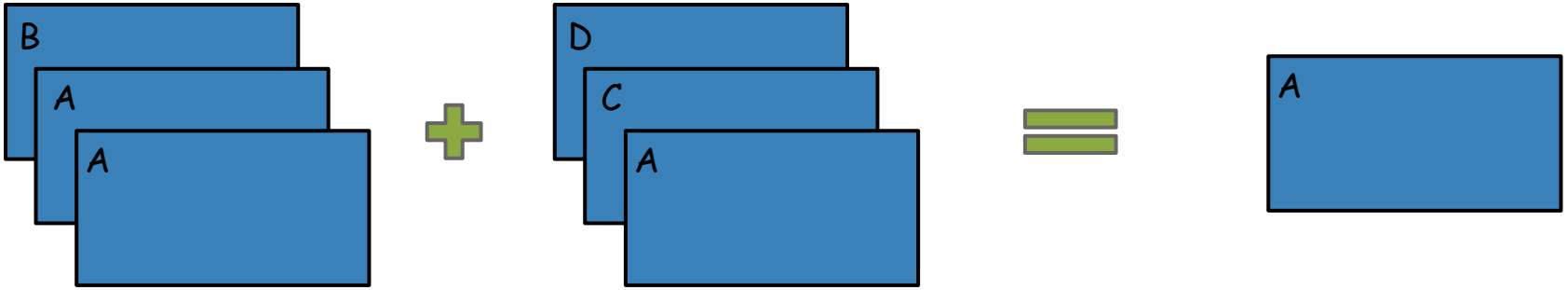
num = 3



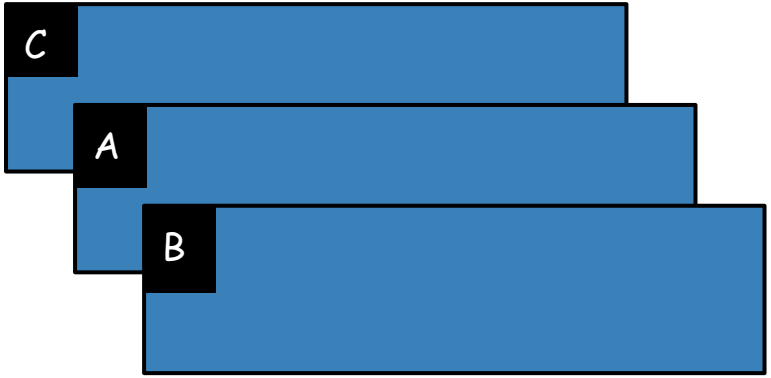
union (+)

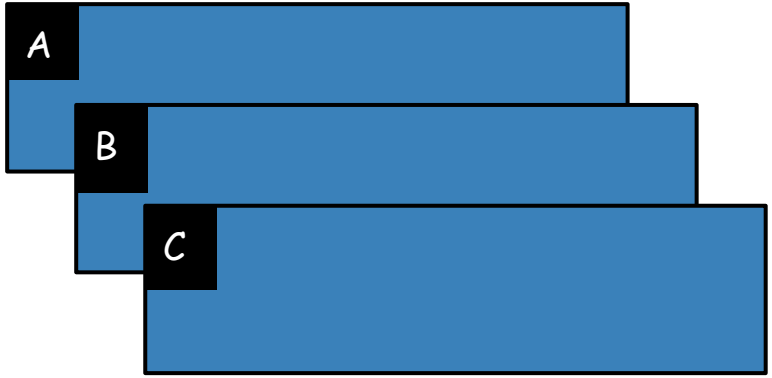


intersection



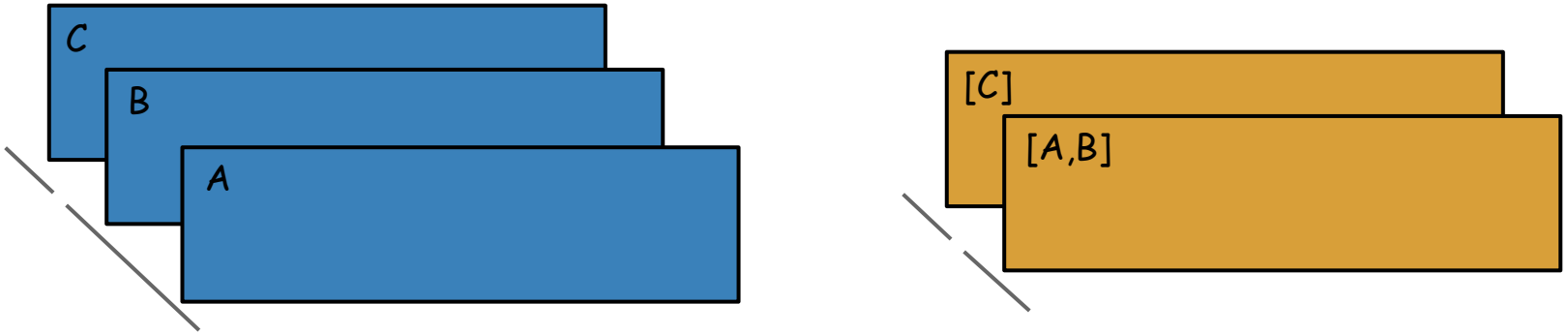
sortByKey



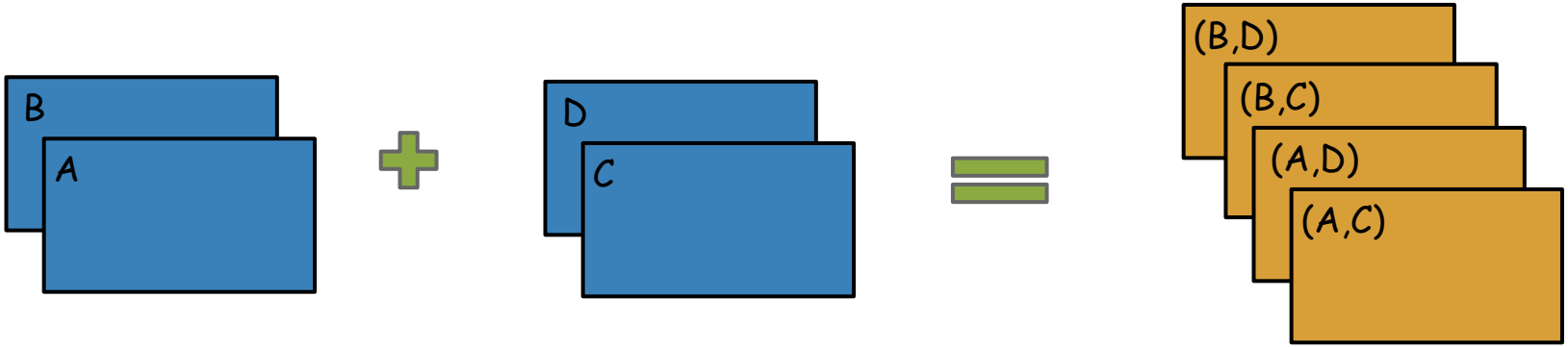


sortBy

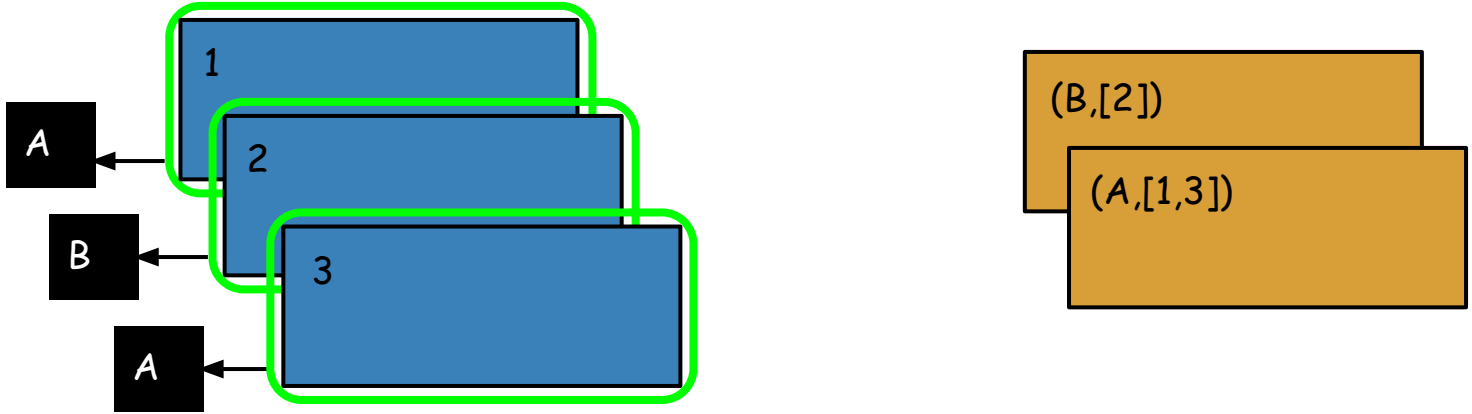
glom



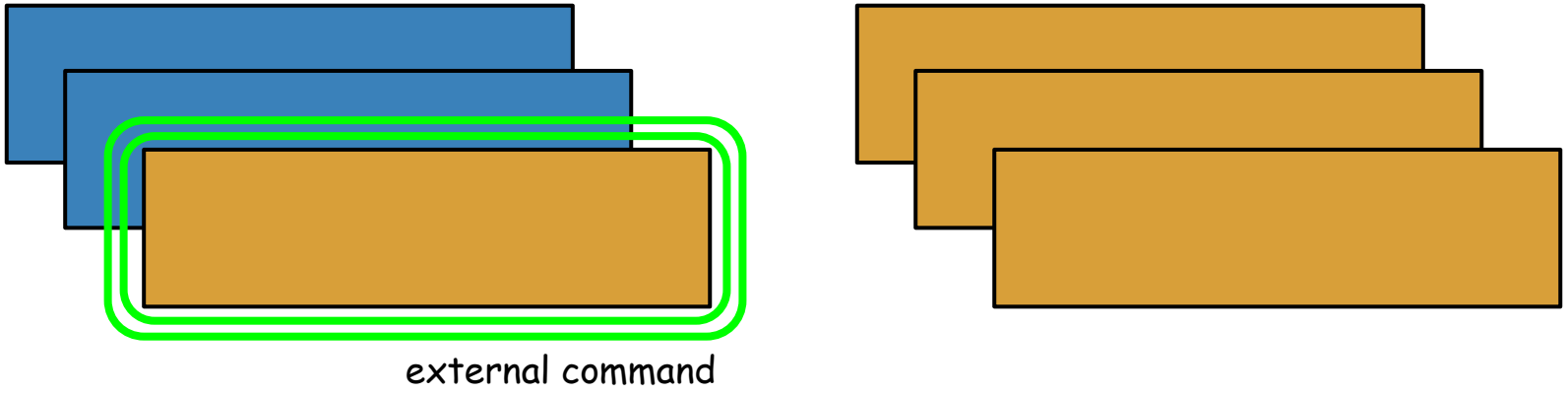
cartesian



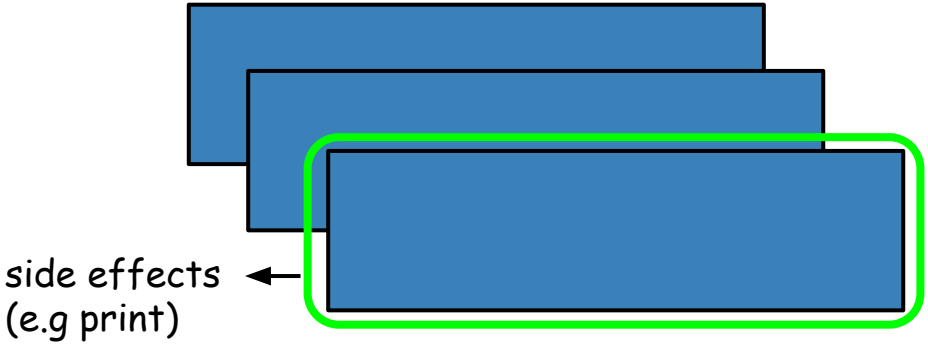
groupBy



pipe

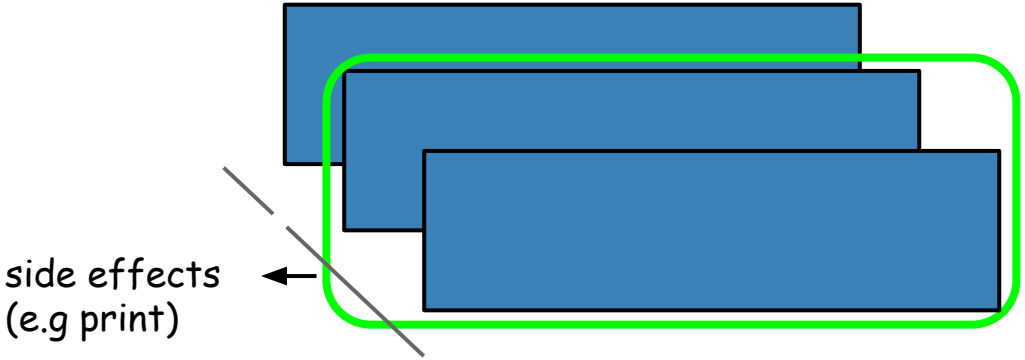


foreach



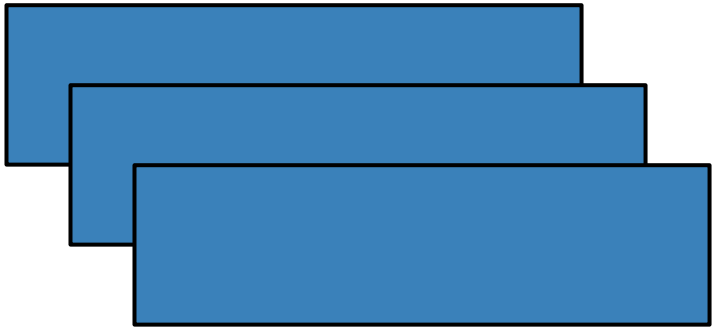
*no return value, original RDD unchanged

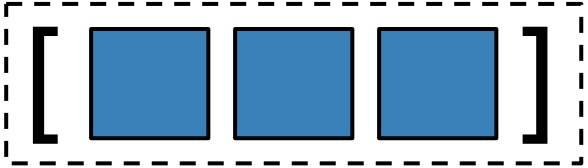
foreachPartition



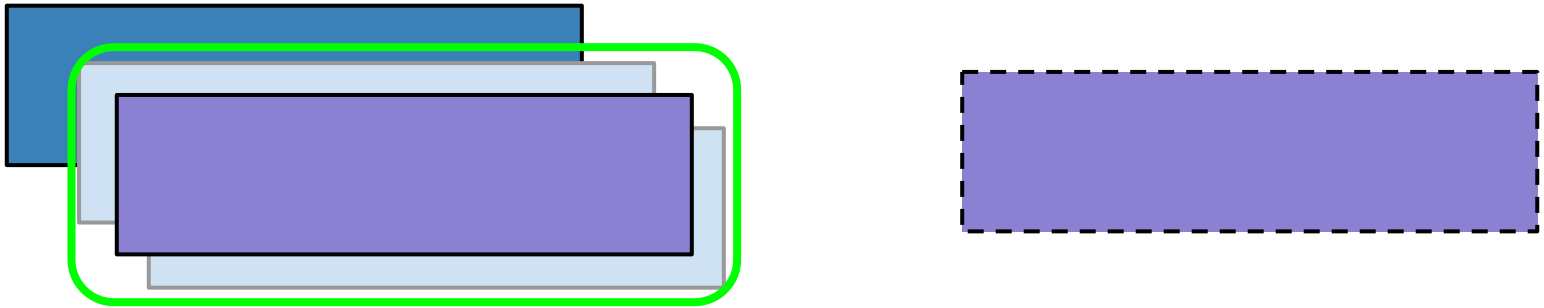
*no return value, original RDD unchanged

collect

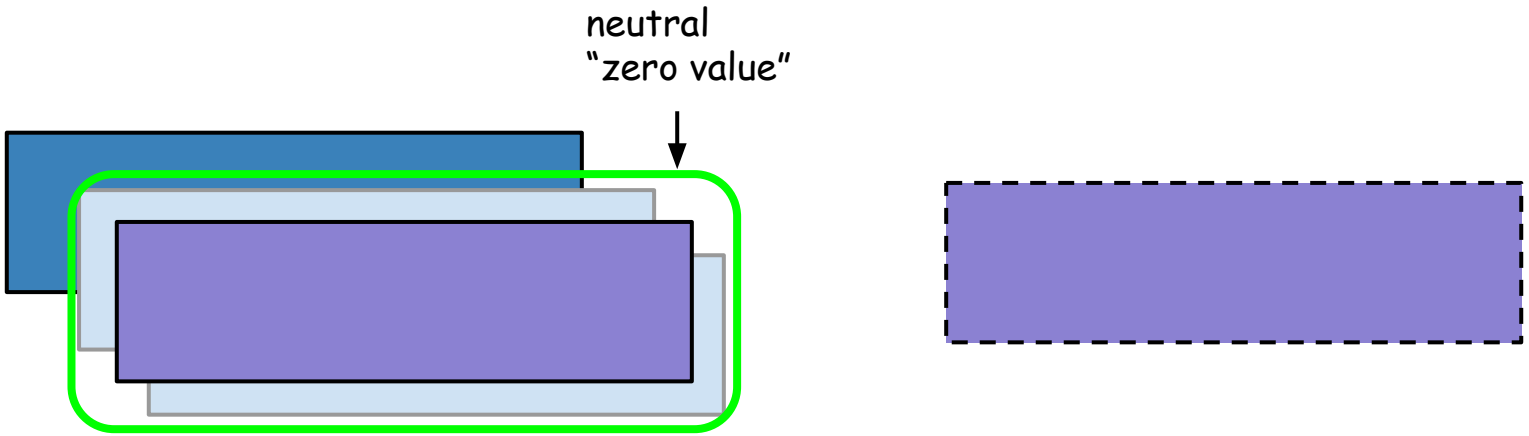




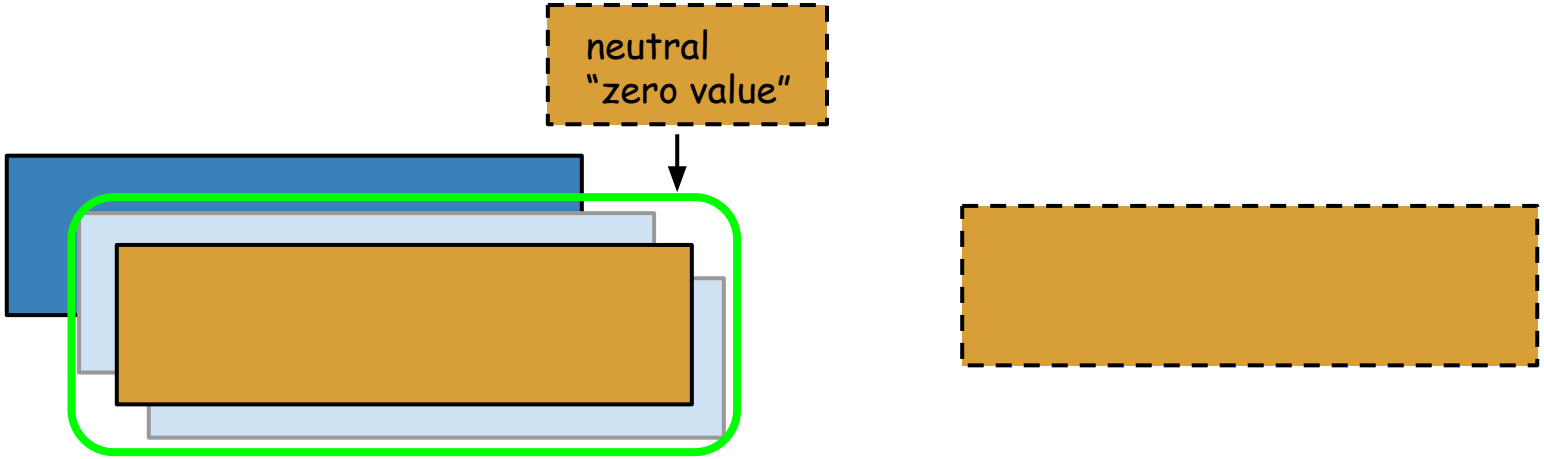
reduce



fold

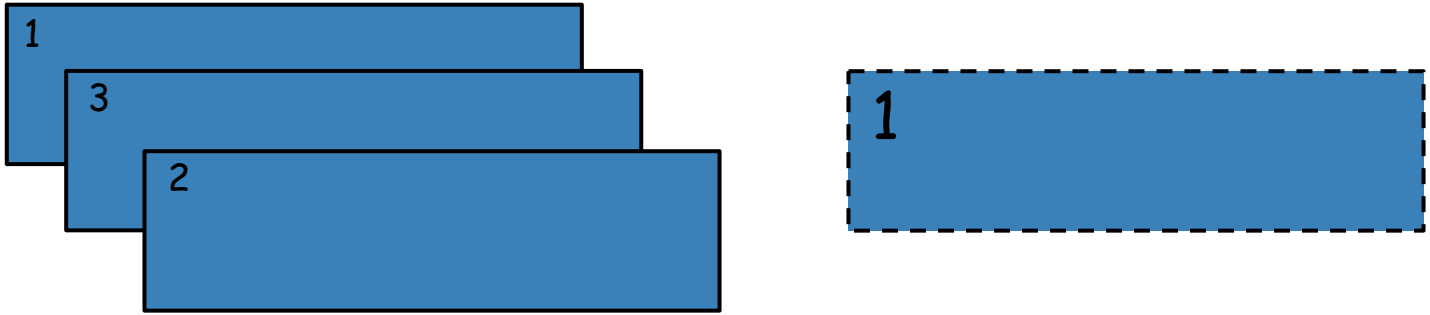


aggregate

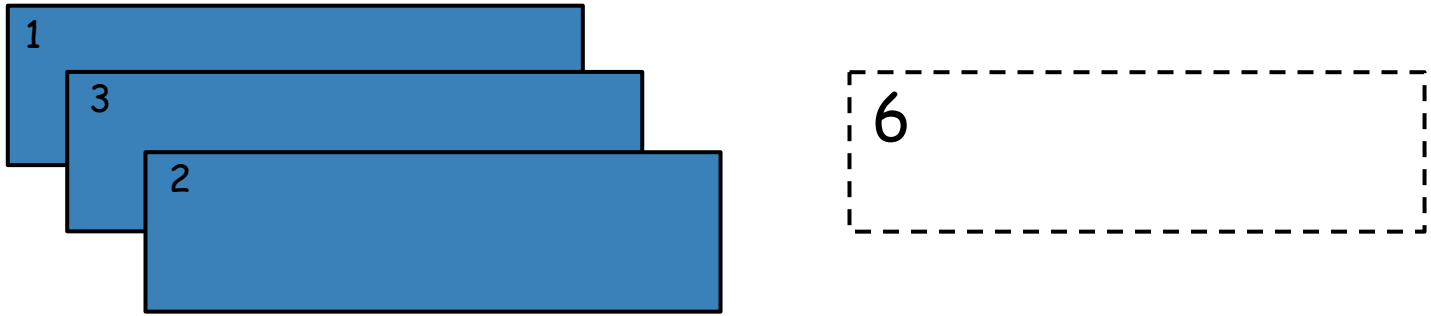


max

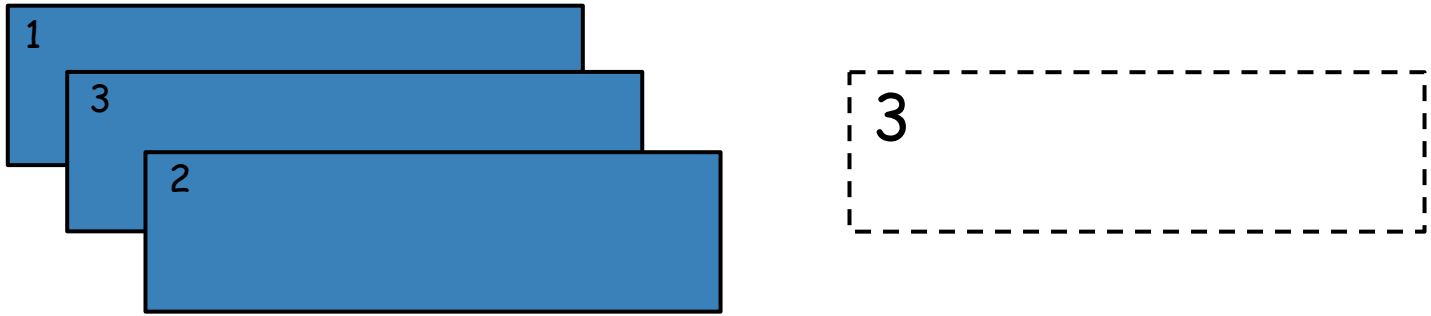
min



sum

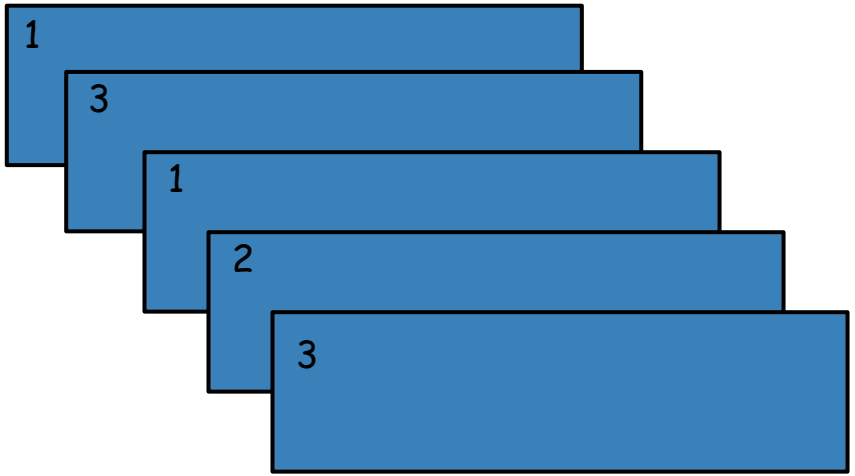


count



histogram

buckets = 2



((1,2,3),[2,3])

mean

2.0

variance

0.666...

stdev

0.816...

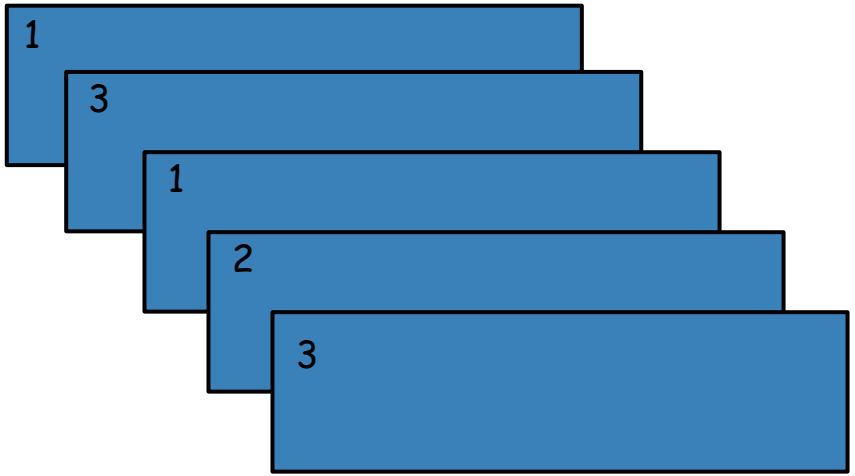
sampleStdev

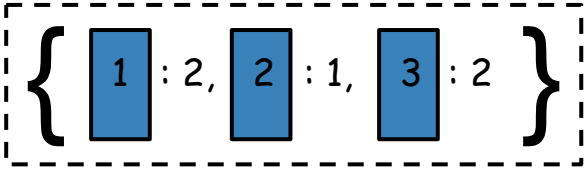
1.0

sampleVariance

1.0

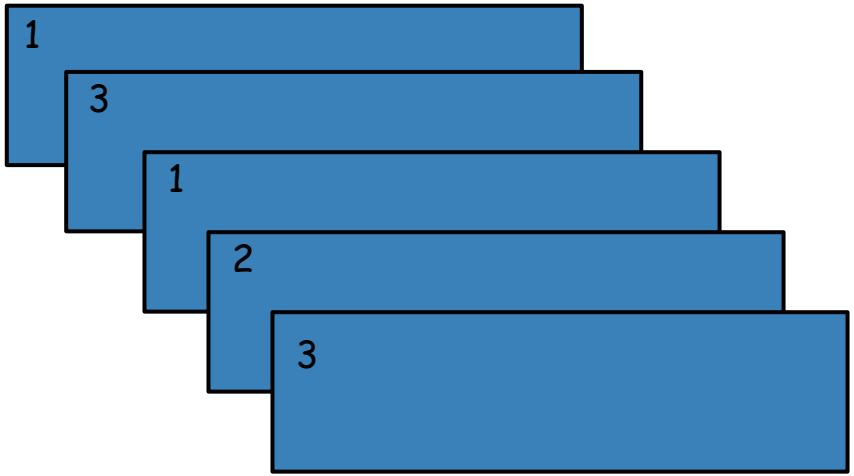
countByValue

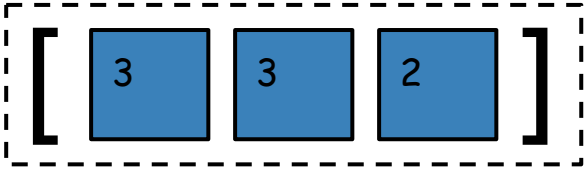




top

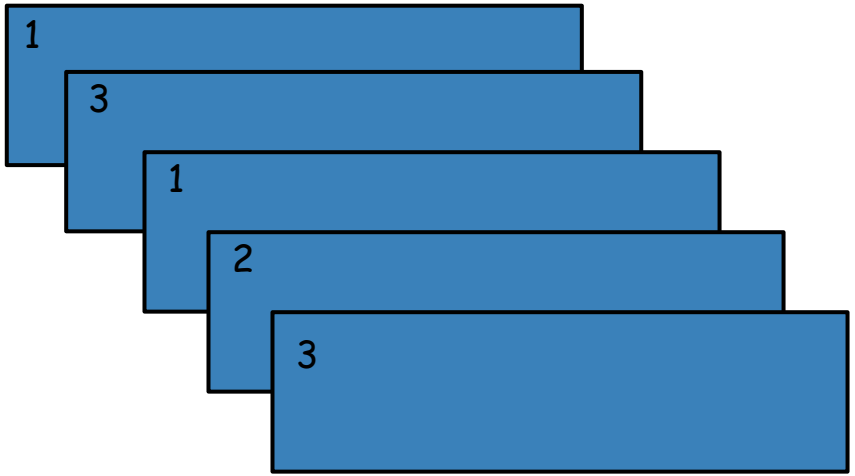
num = 3

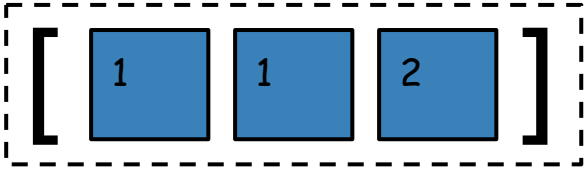




takeOrdered

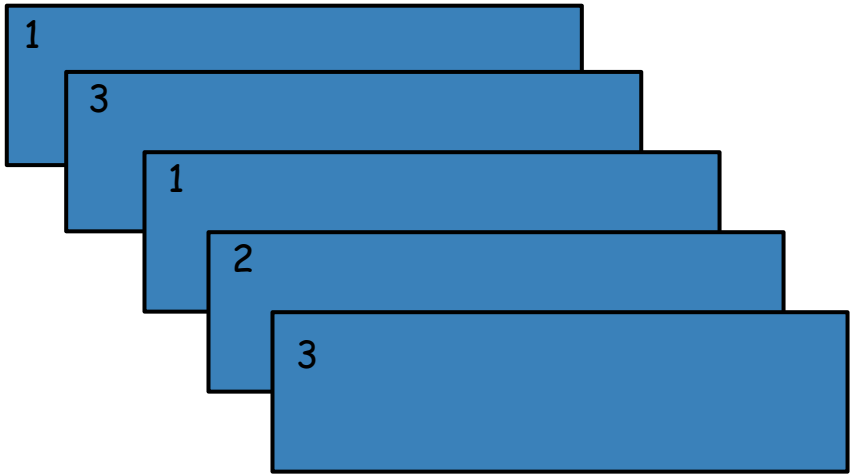
num = 3

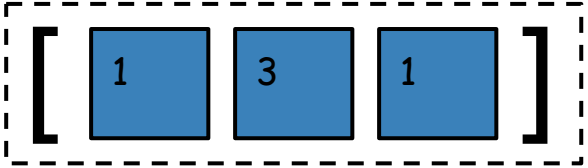




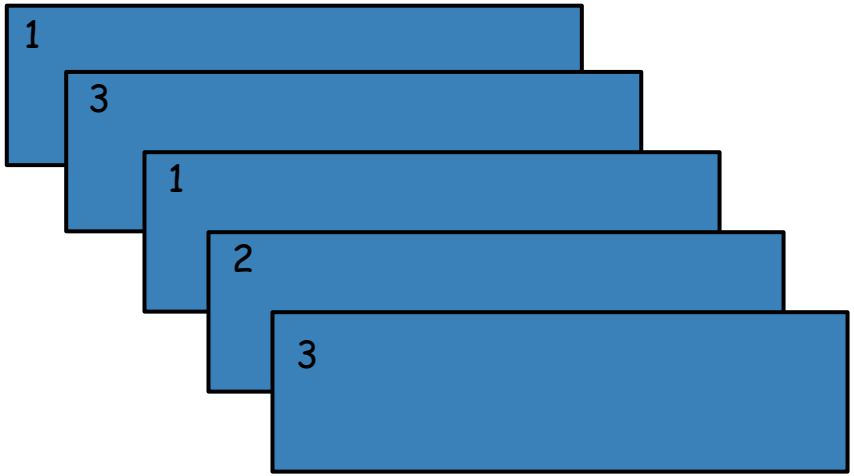
take

num = 3

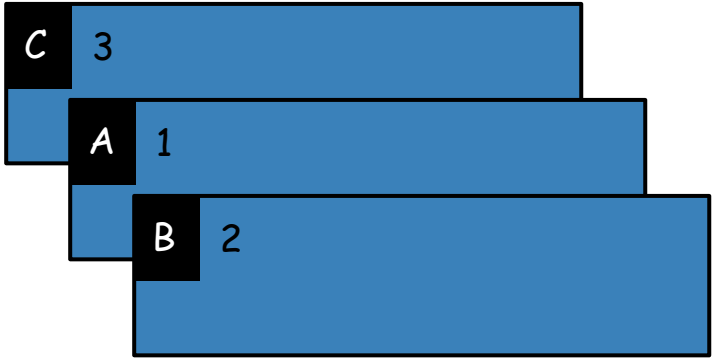


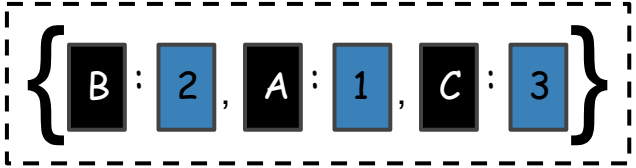


first

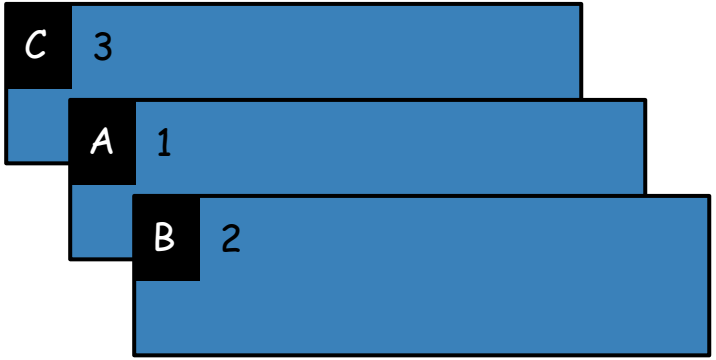


collectAsMap

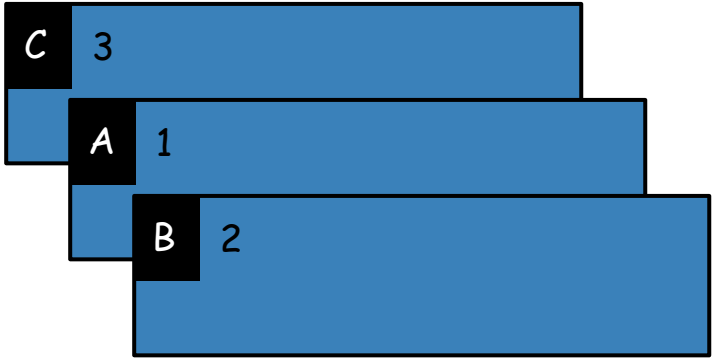




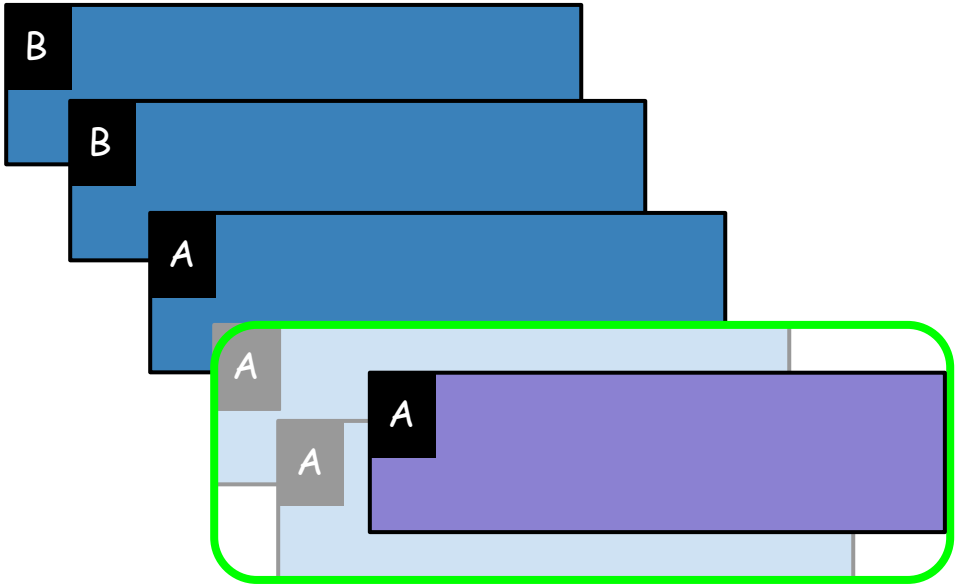
keys



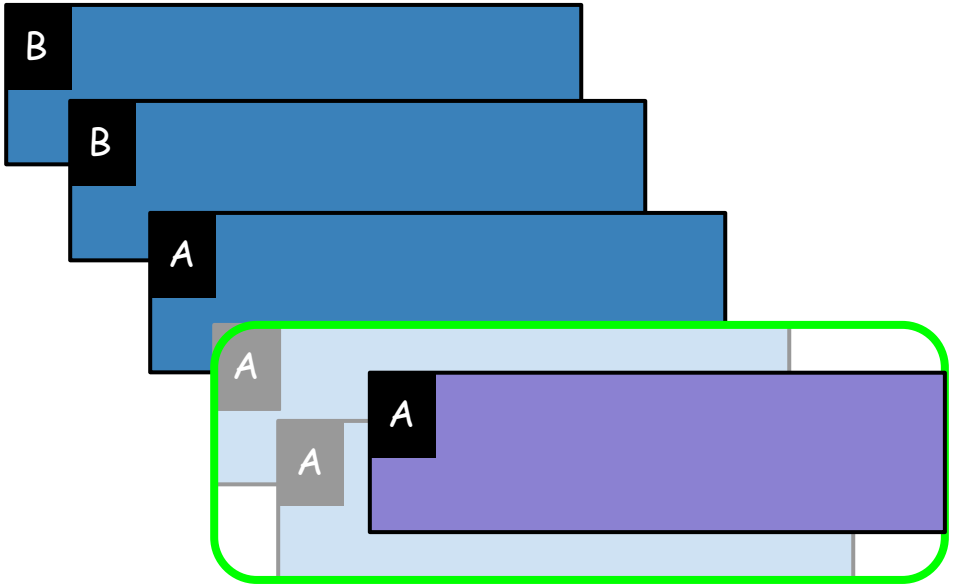
values

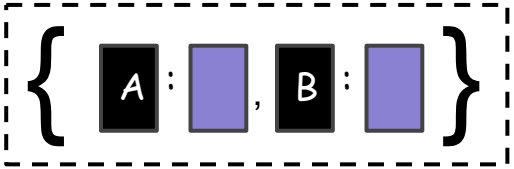


reduceByKey

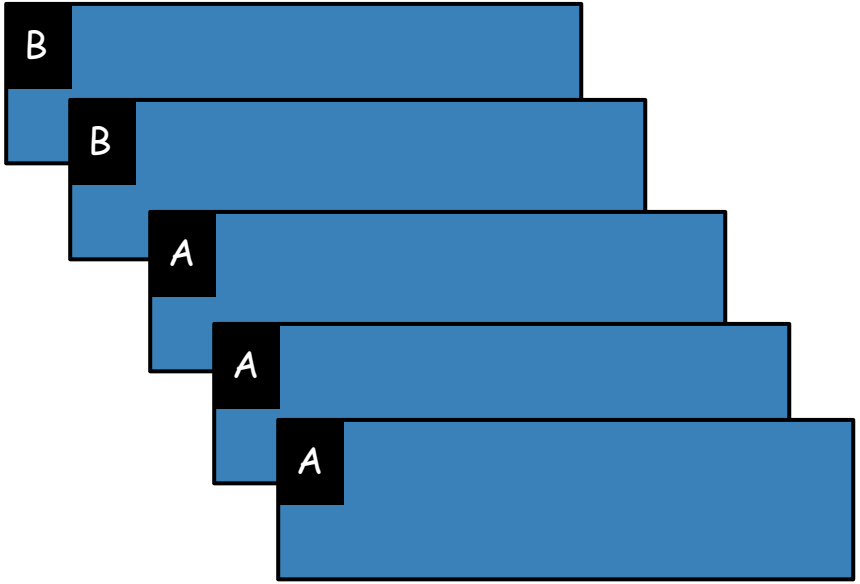


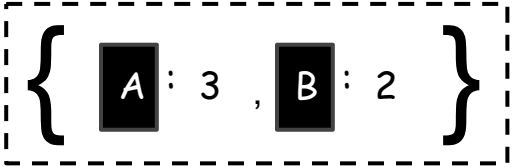
reduceByKeyLocally



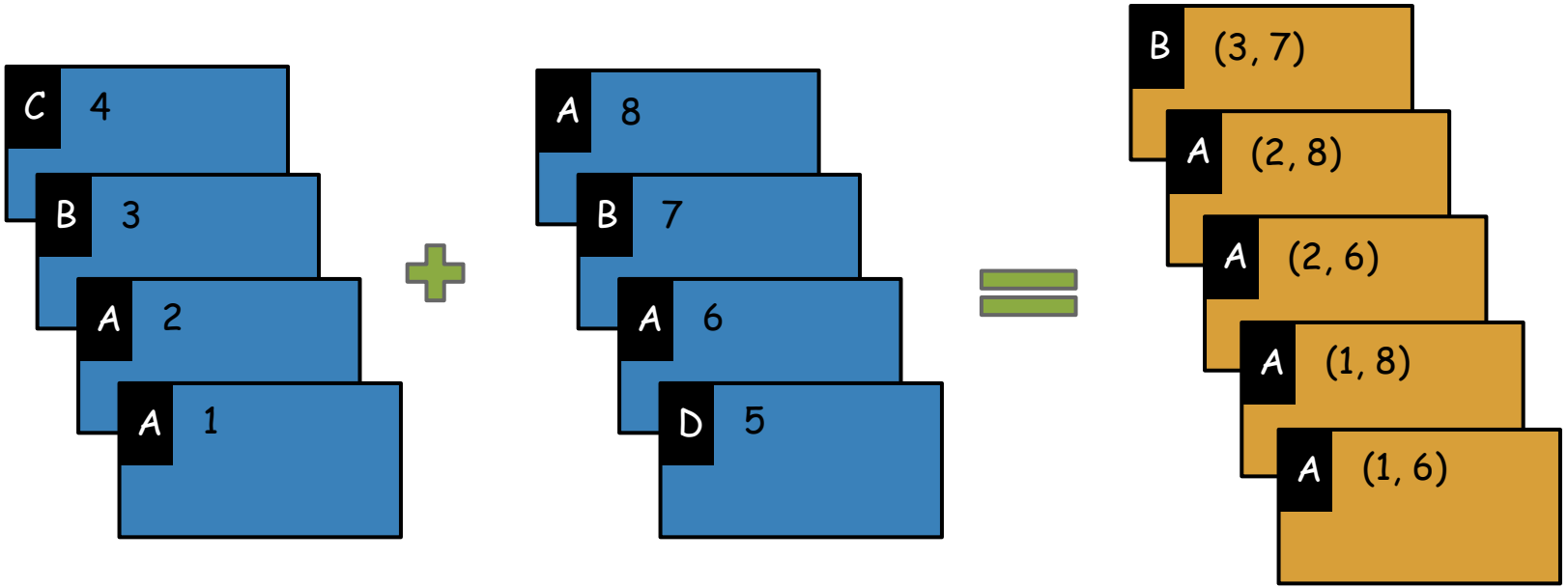


countByKey

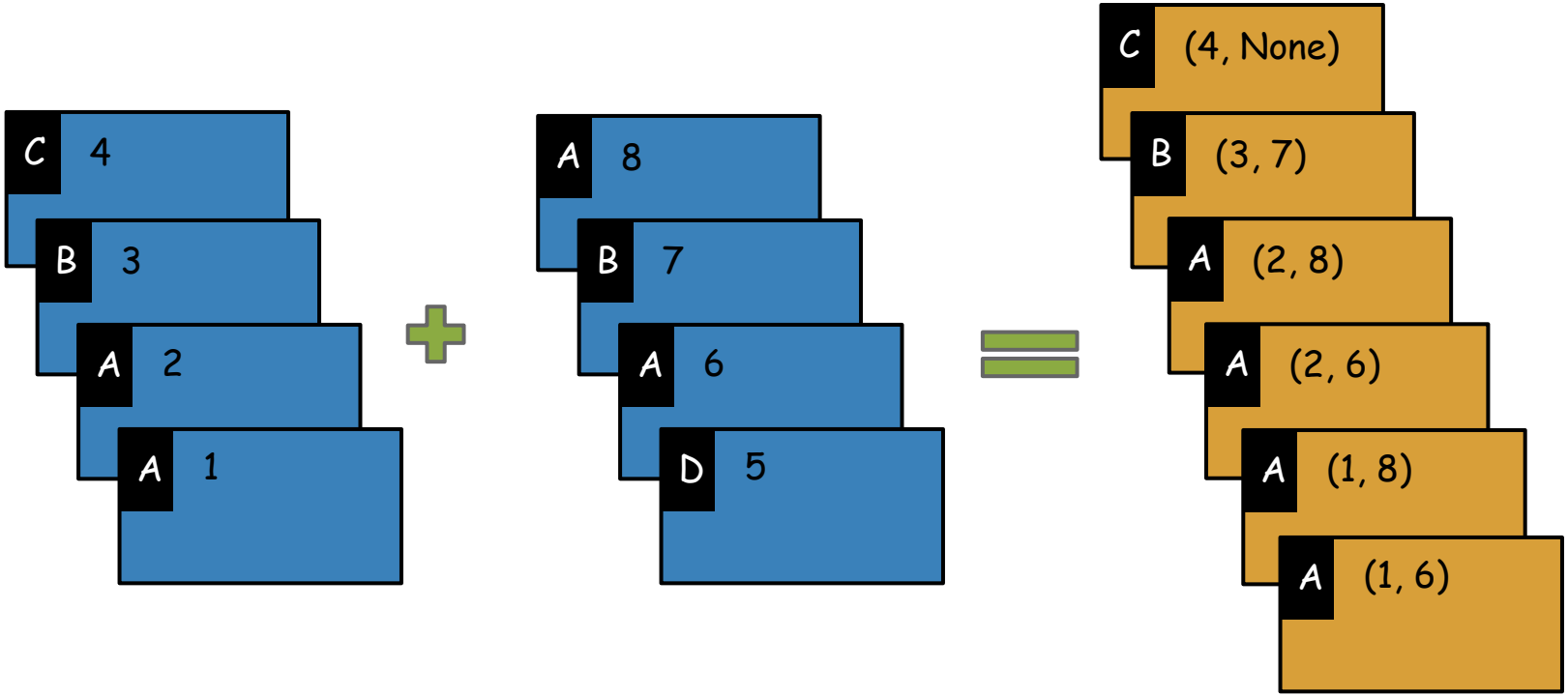




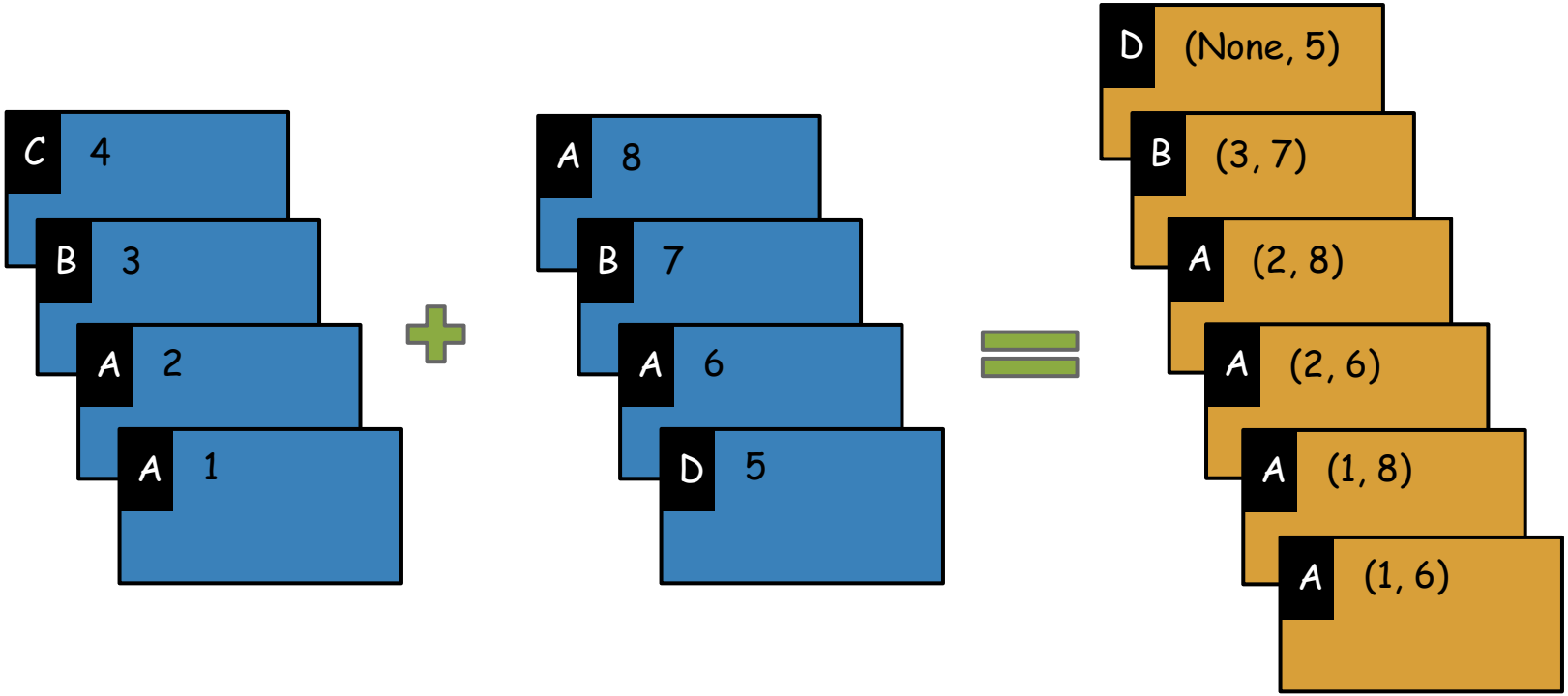
join



leftOuterJoin

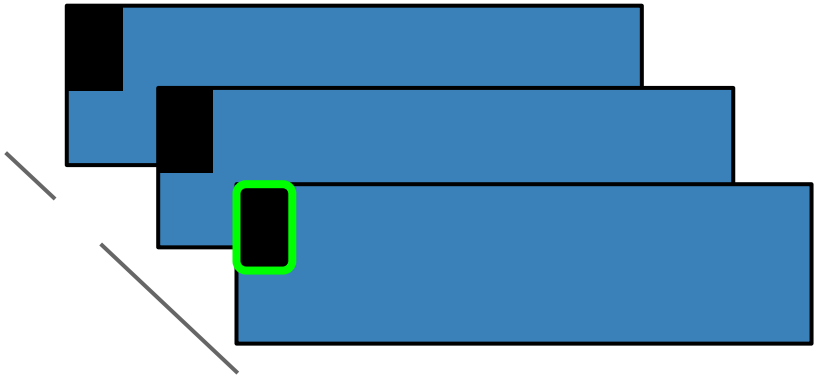


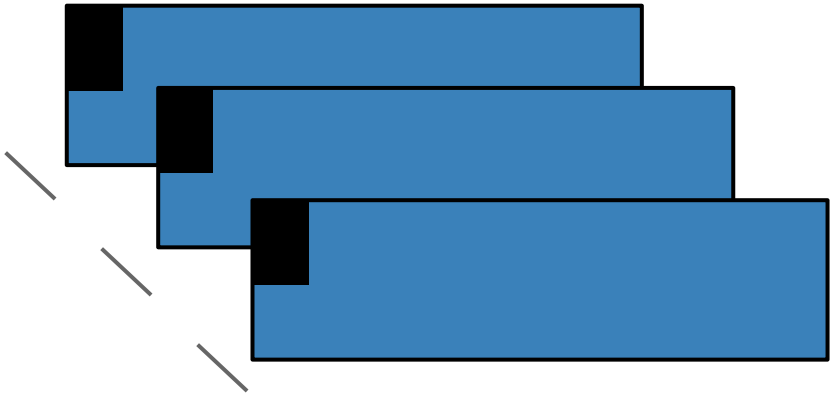
rightOuterJoin



partitionBy

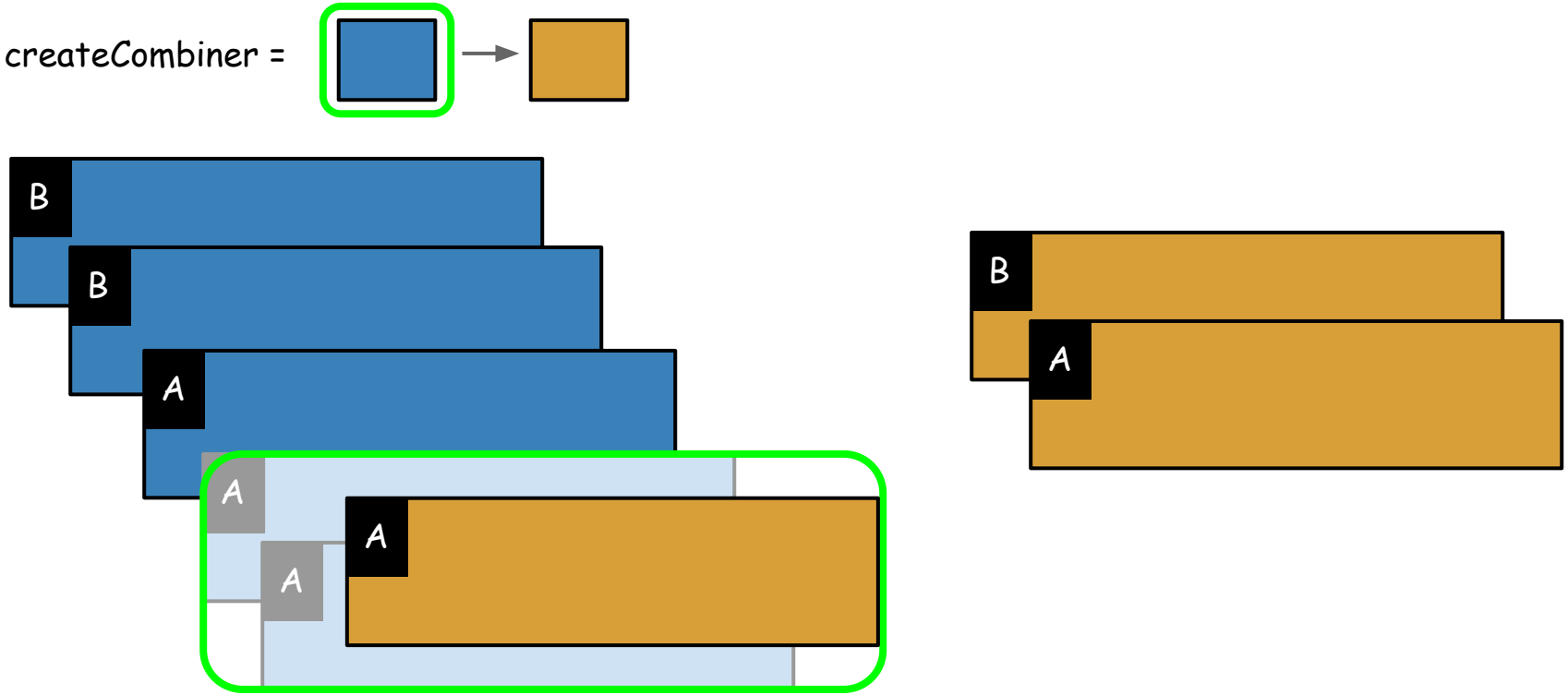
numPartitions = 3





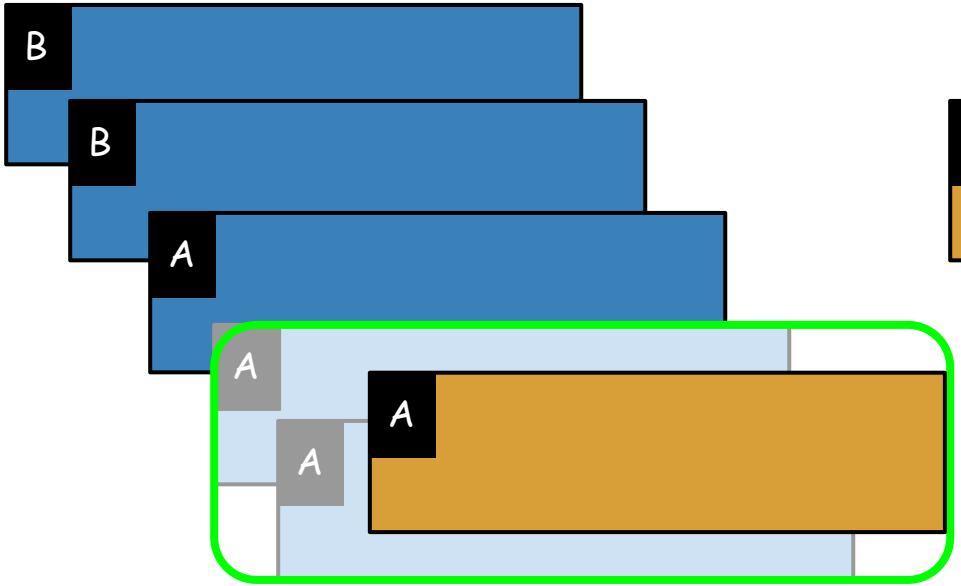
new partition ndex = f(key) % numPartitions

combineByKey

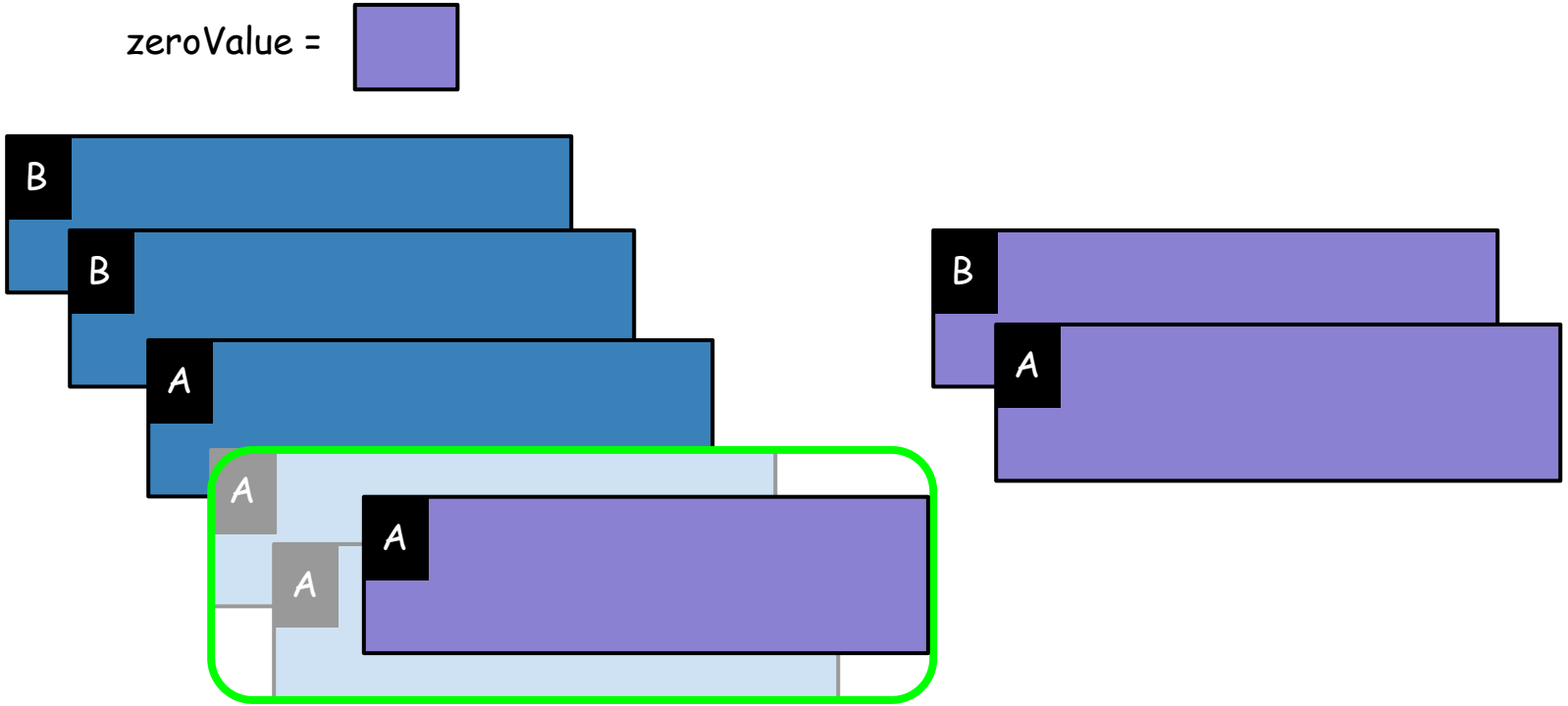


aggregateByKey

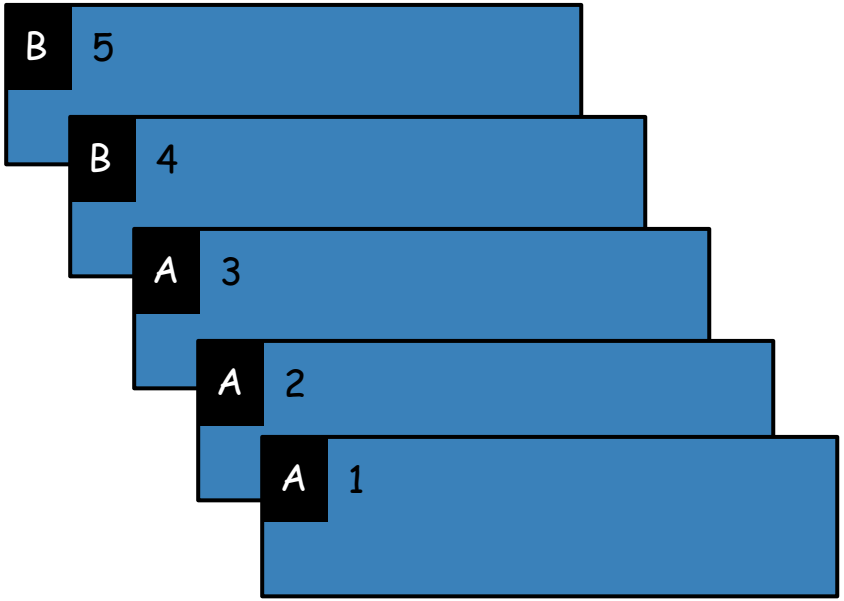
zeroValue =

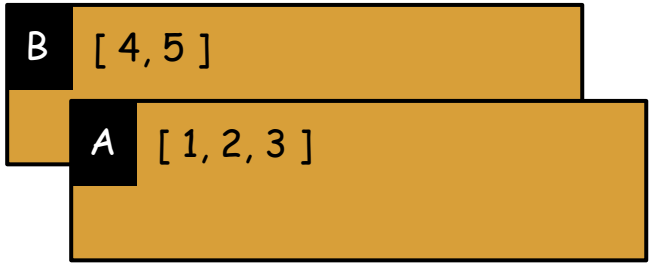


foldByKey

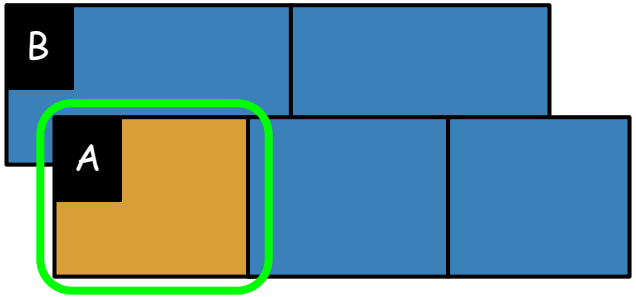


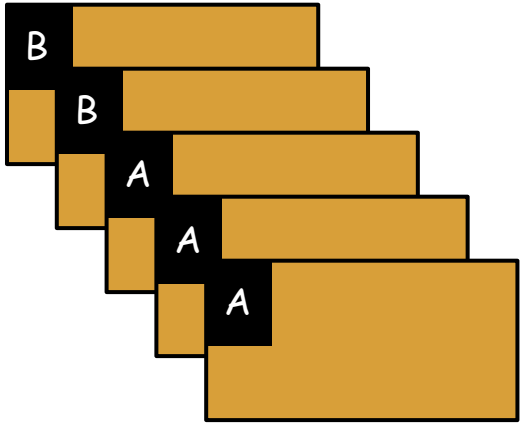
groupByKey



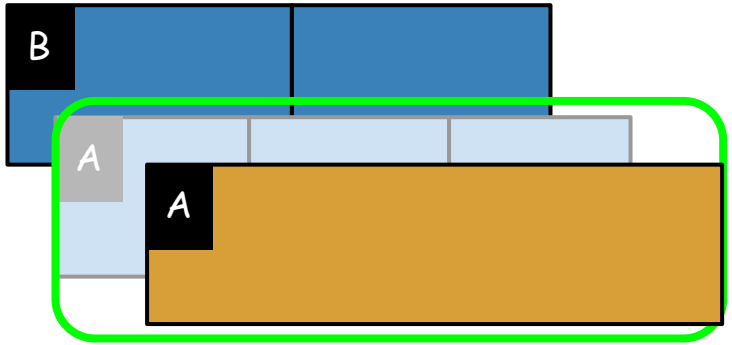


flatMapValues

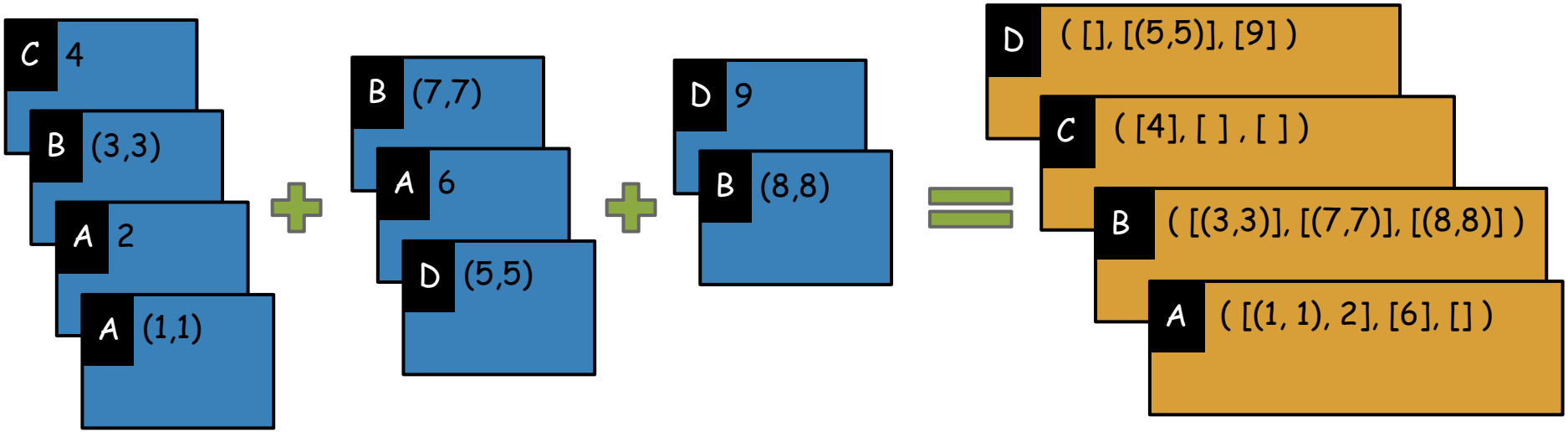




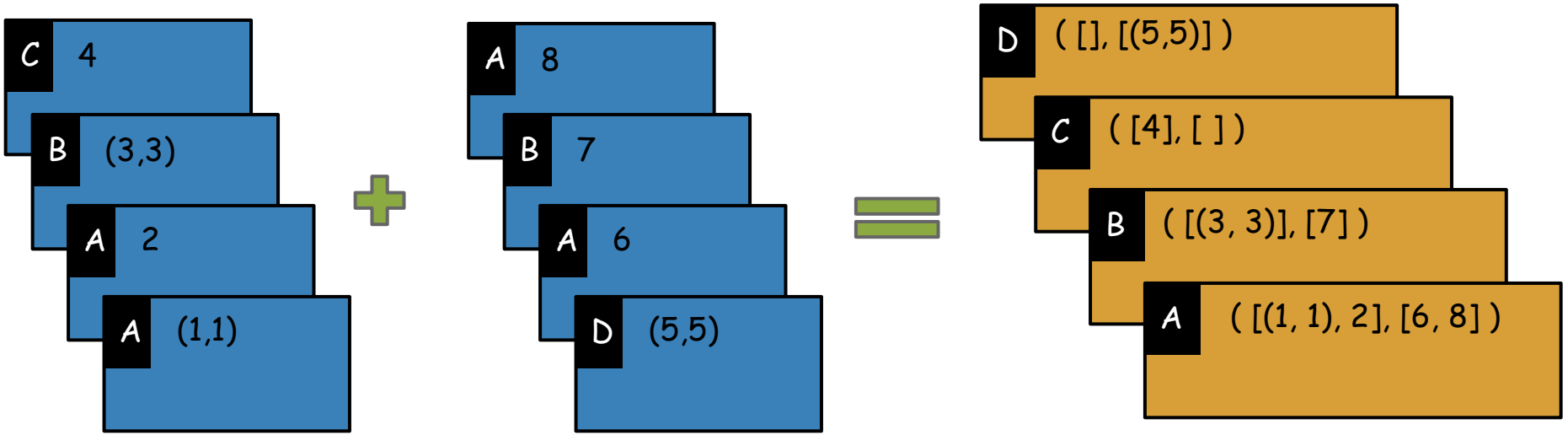
mapValues



groupWith

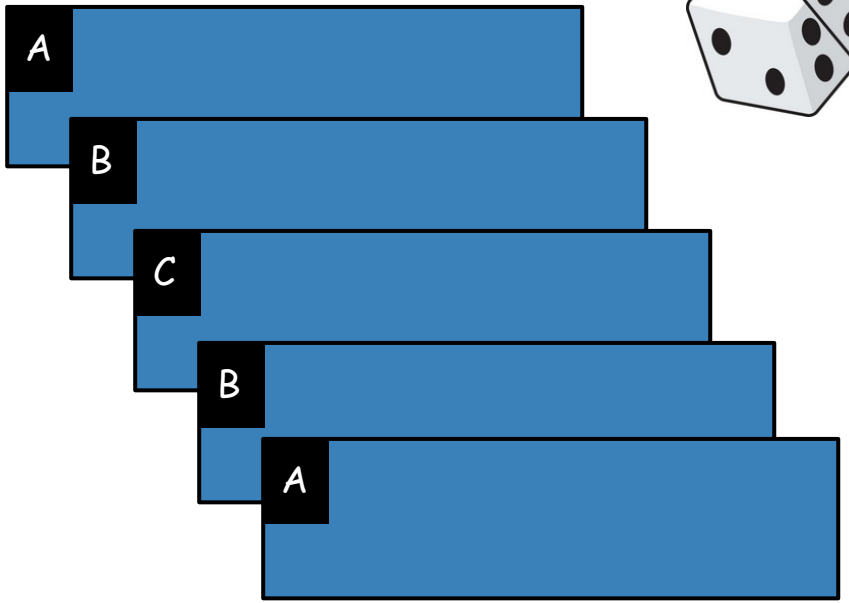


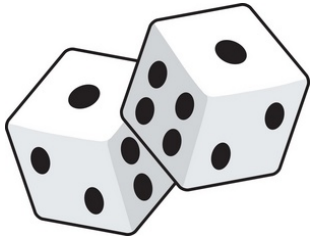
cogroup

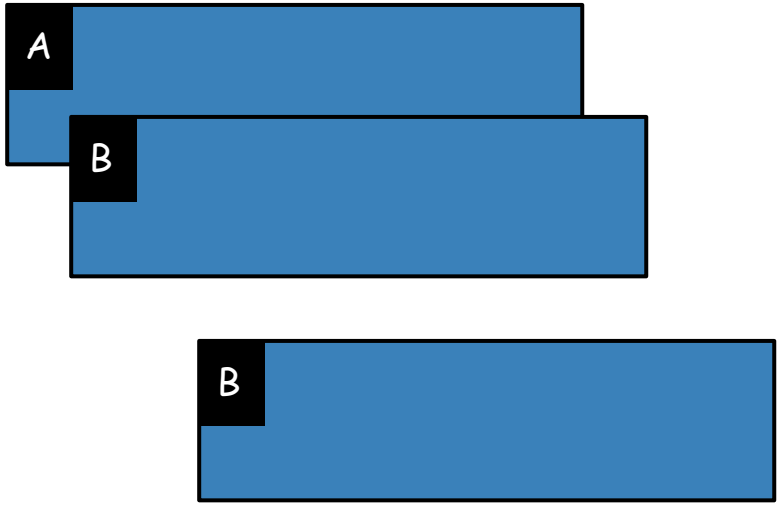


sampleByKey

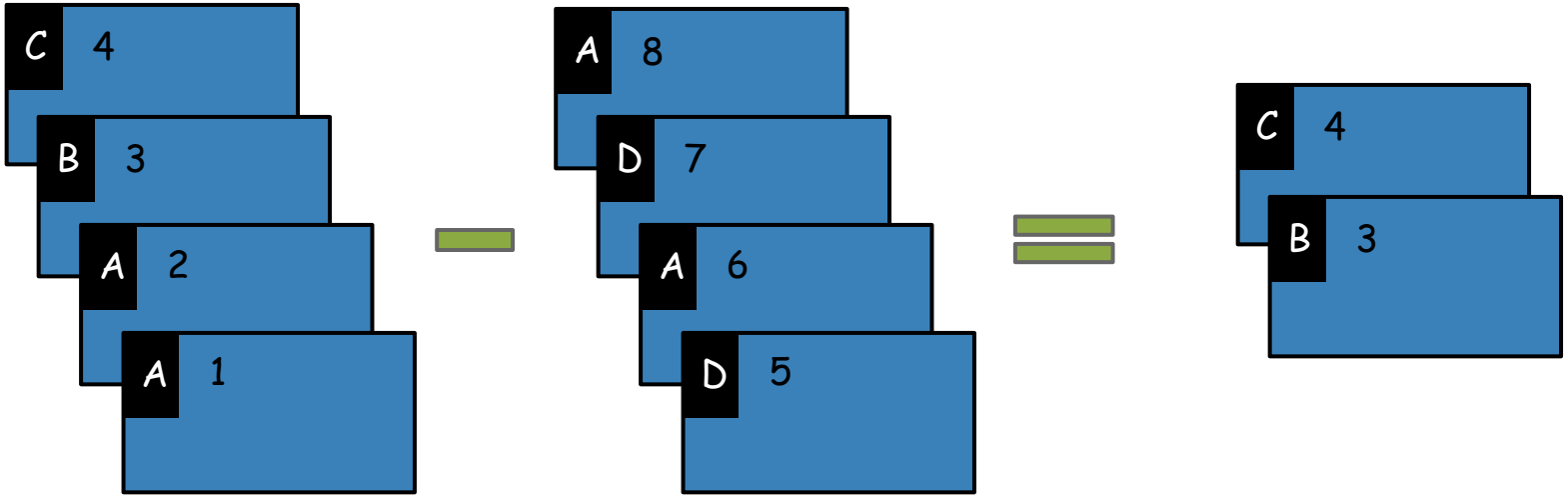
fractions = {'A': 0.5, 'B':1, 'C': 0.2}



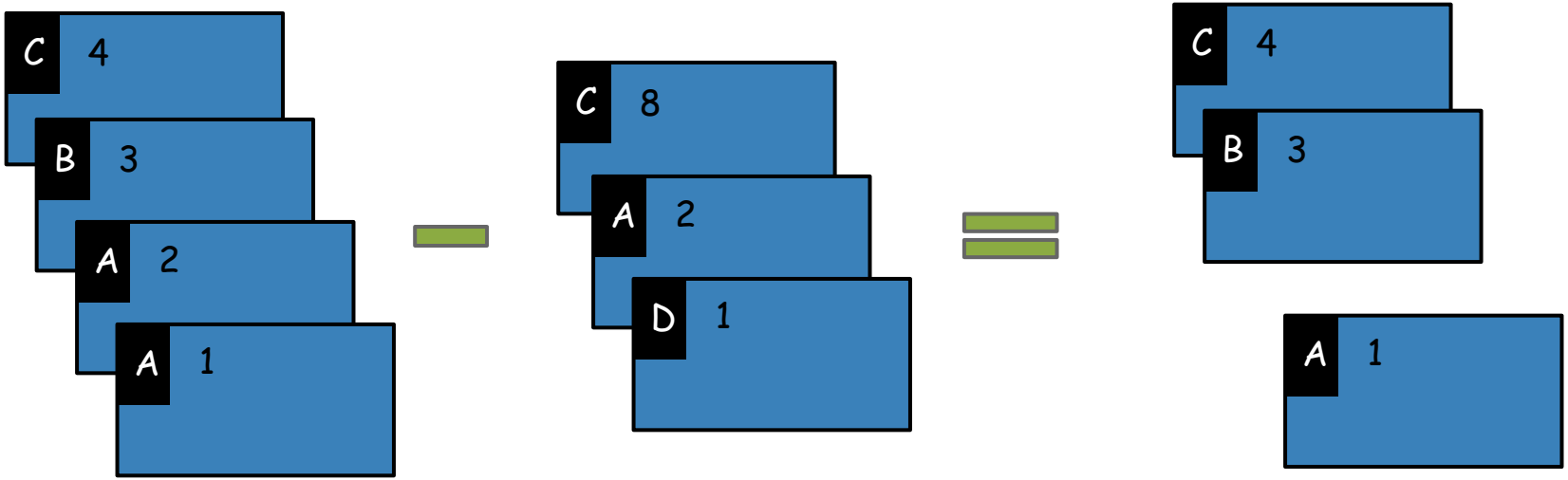




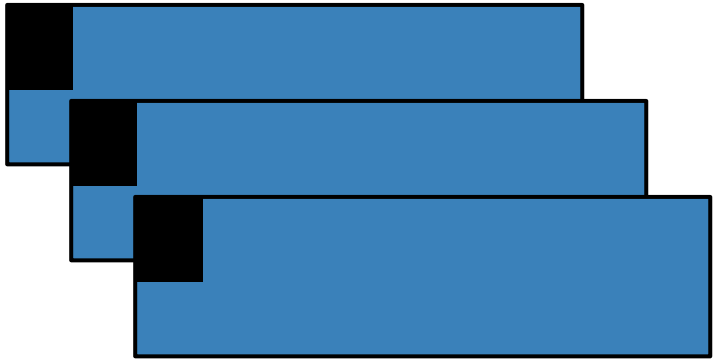
subtractByKey



subtract

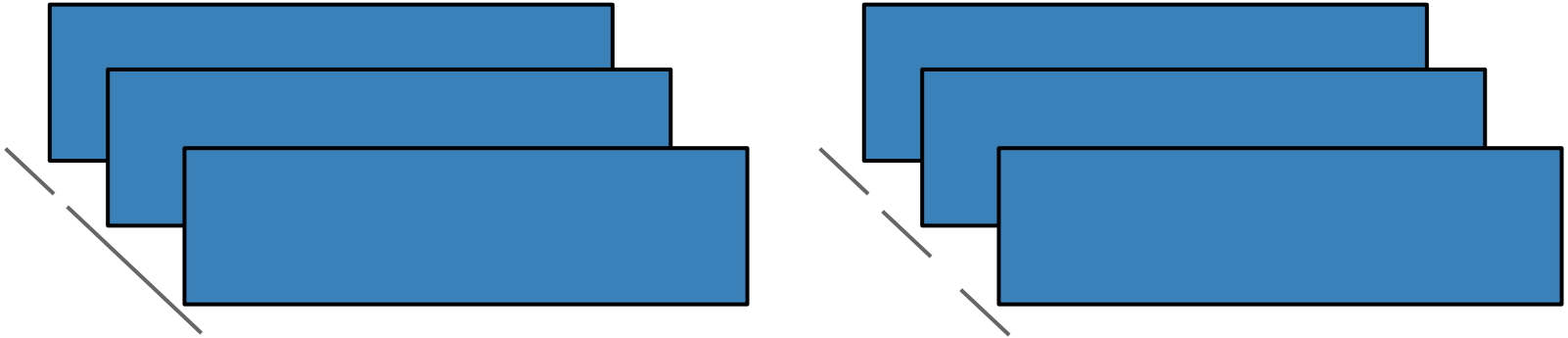


keyBy



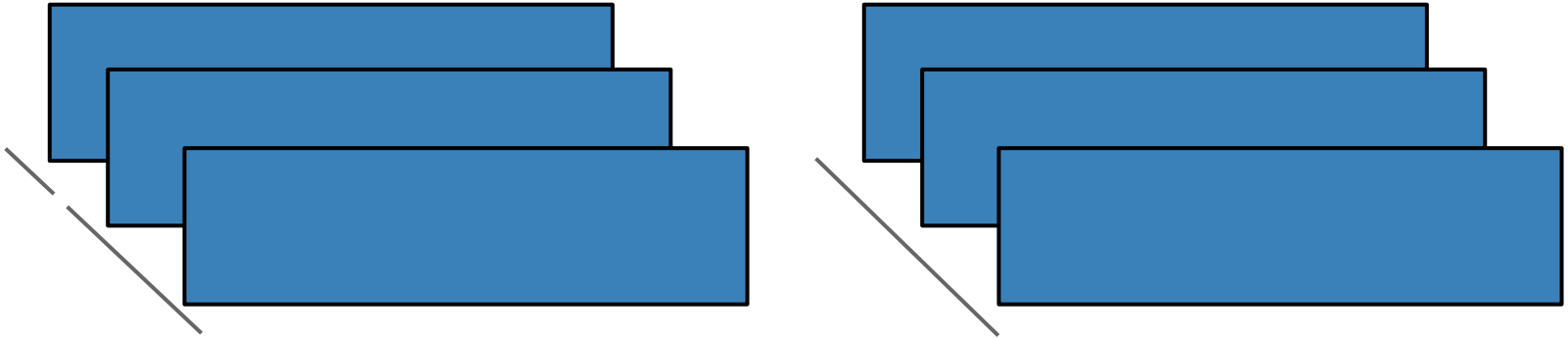
repartition

numPartitions = 3

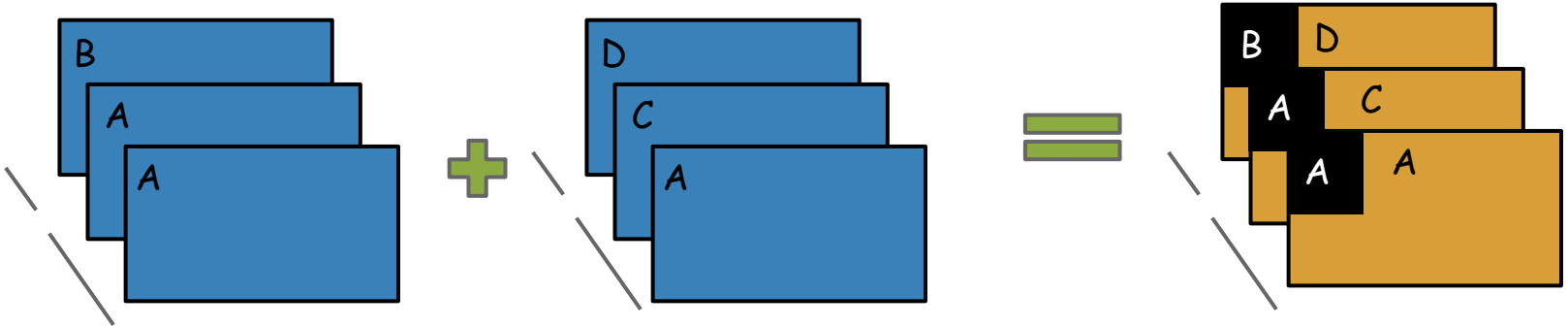


coalesce

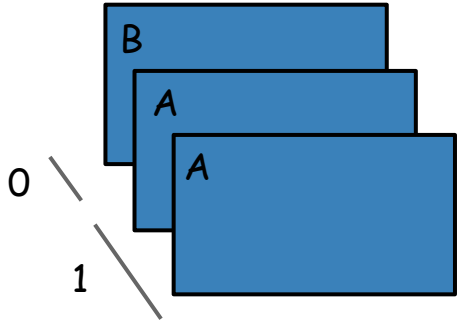
numPartitions = 1

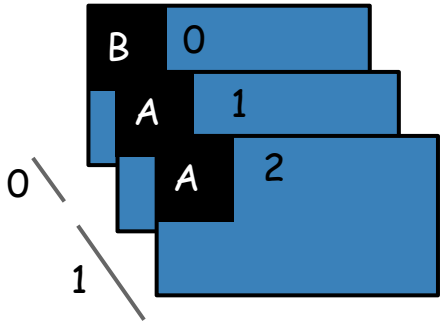


zip

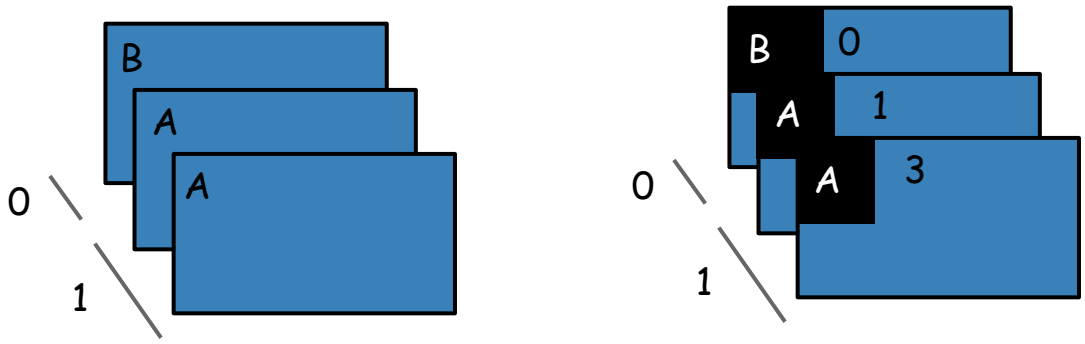


zipWithIndex





zipWithUniqueId



uniqueId = element index * #partitions + partition index