DATE: PAGE:

20/5/25 Lab-8.

Aggregate

```
scala> val numbers = sc.parallelize(1 to 20,4).
numbers: org.apache.spark.rdd.RDD[Int] =
ParallelcollectionRDD[18] at parallelize at <console>:24.

scala> numbers.glom().collect()
res22. Array[Array[Int]] = Array(Array(1,2,3,4,5),
Array(6,7,8,9,10), Array(11,12,13,14,15), Array(16,17,18,19,20))

scala> numbers.aggregate(0)(_+_,_+_).
res23: Int = 210
```

Fold operation.

```
Scala>.numbers.fold(0)((x,y) => x+y).

Scala> numbers.aggregate(1)(_+_, _+_).
res37: Int = 215

Scala> val numbers = sc.parallelize(1 to 4,2)
numbers: org.apache.spark.rdd.RDD[Int]:
ParallelcollectionRDD[20] at parallelize at <console>:24.

Scala> numbers.glom.collect
res39: Array[Array[Int]] = Array(Array(1,2), Array(3,4))

Scala> numbers.aggregate(2)(_+_, _*_).

Scala> valfunc = (x:Int) => x+x
```

DATE: PAGE:

```
Scala> Afunc(3)
Res0 : Int = 6
```

Lab9.
```
Scala> val sampleRDD = SC.parallelize(List(6,7,5,3,10,25,9,20,35))
sampleRDD: org.apache.spark.rdd.RDD[Int] = ParallelCollection
RDD[26] at parallelize at <console>:24

Sala> sampleRDD.collect
res855: Array[Int] = Array(6,7,5,3,10,25,9,70,38
```

Lab-10.

-> write a scala program to print numbers from 1 to 100 using for loop.

```
-> sudo apt install scala.
-> nano PrintNumbers.sala.
    object PrintNumbers {
        def main(args: Array[String]): Unit = {
            for(i <- 1 to 100) {
                println(i) }
            }
        }
-> scalac PrintNumbers.Scala
-> Scala PrintNumbes.
```

-> using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using spark

DATE:

PAGE:

```
-> mkdir -p wordcountersort/src/main/scala
   cd wordcountsort

-> nano build.sbt

     name := "wordcountsort"
     version := "0.1"
     scalaversion:="2.12.17"
     libraryDependencies ++=Seq(
         "org.apache.spark" %% "spark-core" % "3.3.1"
     )
   mainclass in compile:=some("wordcountsort")

-> nano src/main/scala/wordcountsort.scala

-> echo "hello spark hello spark hello world spark
         spark" > /home/bhoom/desktop/wc.txt

-> sbt run.

-> 3a  import org.apache.hadoop.io.Intwritable
   import org.apache.hadoop.io.text
   import org.apache.hadoop.mapreduce.reducer.
   import java.util.*,
   public class wordcountReducer{
         private final Intwritableresult= new Intwritable()
         public void reduce() throws Io exception{
           int sum=0.
           for (Intwritable val:values){
                 sum+= val.get();
           }
```

DATE: PAGE

```
result.set(sum)
context.write(key, result);
    }
}
```

[illegible]
20/5/25