# aggregate 1

```
seqOp  = (lambda x, y: x + y)  # elementwise sequential operation function
combOp = (lambda x, y: x + y)  # tuplewise operation function

listRDD = spark.sparkContext.parallelize([1, 2, 3, 4, 5, 3, 2])
agg1 = listRDD.aggregate(0, seqOp, combOp)
print(agg1, type(agg1))        # output 20
```

seqOp

```
Current x     x     y
        0     0+1   1
        1
Current x     x     y
        0     0+2   2
        2
Current x     x     y
        0     0+3   3
        3
Current x     x     y
        0     0+4   4
        4
Current x     x     y
        0     0+5   5
        5
Current x     x     y
        0     0+3   3
        3
Current x     x     y
        0     0+2   2
        2
```

combOp

```
Current x     x     y
        0     0+1   1
        1     1+2   2
        3     3+3   3
        6     6+4   4
       10     10+5  5
       15     15+3  3
       18     18+2  2
       20
```

# aggregate 2

```
seqOp2 = (lambda x, y: (x[0] + y, x[1] + 1))         # elementwise sequential
operation function
combOp2 = (lambda x, y: (x[0] + y[0], x[1] + y[1]))  # tuplewise operation
function

listRDD = spark.sparkContext.parallelize([1, 2, 3, 4, 5, 3, 2])
agg2 = listRDD.aggregate((0, 0), seqOp2, combOp2)
print(agg2)                                          # output (20, 7)

[(1, 2), (3, 4), (5, 3), 2]
```

seqOp2

```
y = (1, 2)
Current x     x[0]    x[1]    y
   (0, 0)     0+1     0+1     1
   (1, 1)     1+2     1+1     2
   (3, 2)

y = (3, 4)
Current x     x[0]    x[1]    y
   (0, 0)     0+3     0+1     3
   (3, 1)     3+4     1+1     4
   (7, 2)

y = (5, 3)
Current x     x[0]    x[1]    y
   (0, 0)     0+5     0+1     5
   (5, 1)     5+3     1+1     3
   (8, 2)

y = 2
Current x     x[0]    x[1]    y
   (0, 0)     0+2     0+1     2
   (2, 1)
```

combOp2

```
Current x     x[0]    x[1]    y[0]
y[1]
   (0, 0)     0+3     0+2     3       2
   (3, 2)     3+7     2+2     7       3
   (10, 4)    10+8    4+2     8       2
   (18, 6)    18+2    6+1     2       1
   (20, 7)
```