Scalable Data Mining
Practice Questions
Spark

1. Which among below is/are NOT a characteristic of Spark?
   (a) In-memory computation
   (b) Fault tolerance
   (c) Cost efficient
   (d) All of them

2. Which of the following actions does not return any value?
   (a) collect()
   (b) count()
   (c) foreach()
   (d) reduce()

3. Which of the following statement/s is/are true about lineage graph?
   (a) It consists of actual data.
   (b) It consists of dependencies between RDDs.
   (c) It helps to reconstruct lost paritions.
   (d) It gets created at the call of action operation.

4. RDD representing a HDFS file will have only one partition for all the blocks of the file. State *True* or *False*.

5. Which feature of Spark is missing from Hadoop Map Reduce?
   (a) Big data framework
   (b) Processes structured and unstructured data
   (c) Iterative computation
   (d) Fault tolerance

6. Consider the following snippet:

```
var a = List[String]()
_______1________
```

Fill in the blanks such that it will add a string ”SDM” to the list ’a’?

(a) a.add(”SDM”)
(b) a = ”SDM”
(c) a ::= ”SDM”

7. Which of the following will get executed correctly?

(a)

```
val f = _ + _
f(5,6)
```

(b)

```
val f = (_:Int) + (_:Int)
f(5,6)
```

(c)

```
val f = _ + 1
f(5)
```

8. Consider the two statements below for lineage dependency:

(i) Join with inputs co-partitioned induces wide dependency.
(ii) groupByKey is a transformation which induces wide dependency.

Which of them is the correct statement for the above?

(a) (i) is True, (ii) is False.
(b) (i) is False, (ii) is True.
(c) Both of them are true.
(d) Both are false.

9. Consider the following code snippet below:

```
val Rdd1 = sc.parallelize(List(9, 2))
val Rdd2 = numbersRdd.map { x: Int => x * x }
val Rdd3 = squaresRdd.filter { x: Int => x % 2 == 0 }
val v = Rdd3.collect()
```

What does this return?

(a) List(2)
(b) Array(4)
(c) Array(81, 4)
(d) Compilation Error

10. Consider the following code snippet below:

```
val r00 = sc.parallelize(0 to 9)
val r01 = sc.parallelize(0 to 90 by 10)
val r10 = r00.cartesian(r01)
val r11 = r00.map(n => (n, n))
val r12 = r00.zip(r01)
val r13 = r01.keyBy(_ / 20)
val r20 = Seq(r11, r12, r13).foldLeft(r10)(_ union _)
r20.collect()
```

What is the number of stages in the DAG of the result?

(a) 2

(b) 3

(c) 1

(d) 4