CS60021: Scalable Data Mining 2019
Sample Questions: Hadoop and Spark

1. Which is the correct statement about MapReduce?
   (a) It is an open source data warehouse system to query and analyze large data stored in hadoop files.
   (b) It provides resource management.
   (c) It is a data processing layer of Hadoop.
   (d) All of the above.
2. What happens if number of reducers are set to 0?
   (a) Map only job takes place.
   (b) Reduce only job takes place.
   (c) Reducer output is the final output.
   (d) None of the above.
3. Where does the intermediate output of the mapper go as input to?
   (a) Shuffle
   (b) Mapper
   (c) Reducer
   (d) All of the above
4. Which of the modules controls the partitioning of the keys of map output? On what does the number of partitions depend?
   (a) Combiner , mapper tasks
   (b) Reducer, reduce tasks
   (c) RecordReader , number of records
   (d) None of the above
5. How do we disable the reduce step?
   (a) set conf.setNumreduceTasks = 0
   (b) set job.setNumreduceTasks() = 0
   (c) set job.setNumreduceTasks(0)
   (d) All of the above

6. Which of the methods are invoked by Hadoop framework for splitting of files?

   (a) get.InputSplit()
   (b) getSplit()
   (c) getInputSplit()
   (d) getInputSplit(int)

7. How many mapper outputs are provided as input to a single combiner?

   (a) All of them
   (b) Outputs of one mapper
   (c) As many reducer inputs
   (d) Not fixed

8. What is the correct sequence of data flow in MapReduce?

   (a) Combiner -> Reducer -> Mapper
   (b) Mapper -> Reducer -> Combiner
   (c) Mapper -> Combiner -> Reducer
   (d) Reducer -> Combiner -> Mapper

9. Which of the classes is responsible for conversion of inputs to key-value pairs?

   (a) InputSplit
   (b) FileInputFormat
   (c) Mapper
   (d) RecordReader

10. Where does Mapper store its intermediate output?

    (a) In-Memory
    (b) Local Disk
    (c) HDFS
    (d) All of the above

11. Consider the following code snippet:

```
public class WordCount
{
public static class Map extends Mapper<LongWritable,Text,Text,IntWritable> {
private Text token = new Text()
public final static IntWritable mapvalue = new IntWritable(1)
public void map(LongWritable key, Text content,Context con) throws IOException,
InterruptedException{
String line = content.toString();
StringTokenizer tokenize = new StringTokenizer(line);
```

```
while (tokenize.hasMoreTokens()) {
token.set(____________ (1));
__________________ (2);
}
}
}
```

Fill in the blanks:

(a) (1) tokenize.nextToken() (2) con.write(token, new IntWritable(1))

(b) (1) tokenize.nextToken() (2) con.write(line, new IntWritable(1))

(c) (1) tokenize.nextToken() (2) con.write(token, mapvalue)

(d) Both (a) and (c)

12. Considering the code in question 11 , what is the datatype of key and value pair provided as input to the Mapper?

(a) Integer, String

(b) Integer, Text

(c) LongWritable, Text

(d) Text, IntWritable

13. Consider the code snippet below:

```
public static class Reduce extends Reducer<Text,IntWritable,Text,IntWritable> {
private IntWritable count = new IntWritable()
public void reduce(Text key, Iterable<IntWritable> values,Context context)
throws IOException,InterruptedException {

int summ=0;
for(IntWritable value: values)
{
____________ (1) ;
}
context.write(_____________ (2));
}
}
```

Fill in the blanks:

(a) (1) summ = summ + value.get() (2) key , count.set(new IntWritable(summ))

(b) (1) summ = summ + value.get() (2) key , count.set(summ)

(c) (1) summ = summ + values.get() (2) key , summ

(d) Both (a) and (b)

14. Which of the following is true about data locality?

(a) Moving computation to data instead of data to computation.

(b) Moving data to computation instead of computation to data.

(c) Both of them.

(d) None of them.

15. A client connects with the namenode for accessing a file. What does the namenode respond with?

(a) Block ID of the file.

(b) Size of the file.

(c) Block ID and hostname of all data nodes having that file.

(d) Block ID and hostname of any one of the data nodes having that file.

16. Which of the features overcomes Single Point of Failure in HDFS?

(a) Erasure Coding

(b) HDFS Namenode High Availability

(c) HDFS Federation

(d) All of the above

17. Which of the following modules does client first connect to while reading/writing from/to HDFS?

(a) Secondary NameNode

(b) DataNode

(c) NameNode

(d) None of the above

18. Which of them is true about metadata?

(a) FsImage is one of the metadata files.

(b) Metadata displays the structure of HDFS directories/files

(c) Metadata contain information like number of blocks and their locations.

(d) All of the above

19. On the failure of Active NameNode, which of them takes up its responsibility?

(a) Standby NameNode

(b) Backup NameNode

(c) Secondary NameNode

(d) None of the above

20. Which of them is true about Rack Awareness Algorithm?

(a) Reduces fault tolerance and improves data high availability and reliability.

(b) Reduces latency and improves network bandwidth.

(c) Increases latency and reduces performance of the cluster.

(d) Both (a) and (b)

21. Which one of them is true in terms of data processing?

(a) Hadoop performs both batch and stream processing.

(b) Spark performs both batch and stream processing.

(c) Hadoop performs batch processing only.

(d) Both (b) and (c)

22. Is Hadoop MapReduce good for iterative algorithms? Select with considerable reason.

(a) Yes, because it is built on the core of fine-grained, lightweight, composable operations.

(b) Yes, because it handles event stream processing thus doing in-memory computations with in-built graph structure.

(c) No, because it is not aware of the whole pipeline of Map-Reduce steps thus being unable to cache intermediate data in-memory.

(d) No, reason is not stated here.

23. State the correct statement for RDD.

(a) RDD is a database.

(b) RDD is a distributed data structure.

(c) RDD is an immutable collection of objects.

(d) Both (b) and (c)

24. In which of the cases do we keep the data in-memory?

(a) Iterative algorithms

(b) Interactive tools

(c) Both (a) and (b)

(d) None of the above

25. Consider the following code snippet below:

```
val Rdd1 = sc.parallelize(List(("a", 1), ("b",2), ("c",3)))
val Rdd2 = sc.parallelize(List(("b", "box"), ("c","cat"), ("d","dog")))
val Rdd3 = Rdd1.join(Rdd2)
Rdd3.collect().foreach(println)
```

What will be the output?

(a) (a,(1,None)) (d,(None,dog))

(b) (b,(box,2))
(c,(cat,3))

(c) (b,(2,box))
(c,(3,cat))

(d) (a,(1,None)) (b,(2,box)) (c,(3,cat)) (d,(None,dog))

26. Consider the code snippet given below:

```
val Rdd1 = sc.parallelize(List(("a", 1), ("b",2), ("c",3)))
val Rdd2 = sc.parallelize(List(("b", "box"), ("c","cat"), ("d","dog")))
val Rdd3 =
Rdd1.leftOuterJoin(Rdd2)
Rdd3.collect().foreach(println)
```

What will be the output?

(a) (a,(1,None)) (b,(2,box)) (c,(3,cat)) (d,(None,dog))

(b) (a,(1,None))
(b,(2,Some(box)))
(c,(3,Some(cat)))

(c) (a,(1,Some(None)))
(b,(2,Some(box)))
(c,(3,Some(cat)))
(d,(None, Some(dog))

(d) (a,(1,None))
(b,(2,box))
(c,(3,cat))
(d,(None, dog))

27. Does this code compile successfully? If yes, what does it print?

```
def func(a :Int) :Int = 2*a
val welcome = func
println(welcome(3))
```

(a) It compiles but generates no output

(b) It does not compile and throws an error.

(c) It compiles and prints "Hello!"

(d) None of the above

28. What is the output for the following code:

```
1. val arr=Array(1,2,3)
2. arr.update(1,7)
```

(a) Compilation error at line 2

(b) val cannot be reassigned.

(c) arr holds (1,7,3)

(d) arr holds (7,2,3)

29. Let us consider the following pseudo-code where RDDs A and D have some common key:

```
B = A.partitionBy(new HashPartitioner(10))
C = D.partitionBy(new HashPartitioner(10))
F = B.join(C)
```

Which of the statements is true?

(a) B and A will use the same paritioner.
(b) C and D will use the same partitioner.
(c) B and F will use the same paritioner.
(d) None of the above.

30. Analyze the code snippet below:

```
val words = sc.parallelize(Seq("We","are","enrolled","into","SDS","course"))

val wordp = words.map(w => (w.charAt(0), w))

wordp.foreach(println)
```

What kind of operation is performed here?

(a) Transformation
(b) Action
(c) No operation

31. Consider the Scala code snippet below:

```
val strings = List("1", "2", "foo", "3", "bar")
strings.flatMap(toInt)
```

What does this return?

(a) List(1, 2, "foo", 3, "bar")
(b) List(List(1), List(2), List("foo"), List(3), List("bar"))
(c) List(1, 2, 3)
(d) List(1), List(2), List("foo"), List(3), List("bar")

32. Consider the code snippet below:

```
val list = List(1,2,3,4,5)
def g(v:Int) = List(v-1, v, v+1)
list.map(x => g(x))
```

What does this return?

(a) List(0, 1, 2, 1, 2, 3, 2, 3, 4, 3, 4, 5, 4, 5, 6)

(b) List(List(0, 1, 2), List(1, 2, 3), List(2, 3, 4), List(3, 4, 5), List(4, 5, 6))

(c) List(0, 1, 2), List(1, 2, 3), List(2, 3, 4), List(3, 4, 5), List(4, 5, 6)

(d) Compilation Error

33. When do you use map-side join?

(a) For joining two very large tables, which don't fit in the memory in the mapper

(b) For joining two very large tables, which don't fit in the memory in the reducer

(c) For joining one very large table which doesn't fit in the memory with another small table which fits in the mapper memory

(d) For joining two small tables which fit in the mapper memory

34. Consider a map-reduce program which takes a collection of documents ids and topics as input in the following format:

`<article id> \t <space separated list of topics>`

and computes lists of documents on each topic.

Complete the following mapper code:

```
Mapper Code:

for line in sys.stdin:
    articleid,content = line.split("\t")
    topics = content.split()
    for topic in topics:
        ------------------- (1)
```

(a) print(articleid, topic, sep='\t')

(b) print(topic, line, sep='\t')

(c) print(topic,articleid,sep='\t')

(d) print(topic, line, sep='\t')

35. Following the question above, complete the reducer code:

```
Reducer Code:

articlelist=[]
prevtopic = NULL
for line in sys.stdin:
         topic, articleid = line.split("\t")

         ----------------------- (2)
```

```
    articlelist.append(articleid)

  else:
    if prevtopic != NULL:
      ________________ (3)
    articlelist=[]
    prevtopic=topic
    articlelist.append(articleid)
```

(a) (2) if topic == prevtopic:
    (3) print(prevtopic, articlelist)

(b) (2) if articleid not in articlelist:
    (3) print articlelist

(c) (2) if len(articlelist) != 0:
    (3) print articlelist

(d) (2) if topic != prevtopic:
    (3) print(articlelist, prevtopic)

36. Mark the correct statement for NameNode in the context of HDFS.

(a) The master node storing actual data

(b) The slave node storing actual data

(c) The master node storing meta data

(d) The slave node storing meta data

37. Which of the following is true:

(i) Total size of all RDDs in a spark program is smaller than the total RAM in the cluster

(ii) Once an RDD on which an action has been called, is materialized, the RDDs on which it depends can be de-materialized (memory unallocated)

(a) Both (i) and (ii) are true

(b) (i) is true and (ii) is false

(c) (ii) is true and (i) is false

(d) Both (i) and (ii) are false

38. Suppose we have four mappers with outputs being as follows:

Mapper 1: (a,1) (b,2)

Mapper 2: (c,3) (c,6)

Mapper 3: (a,5) (c,2)

Mapper 4: (b,7) (c,8)

What will be the key value pairs that will be fed as input to the reducer (i) with Combiner (ii) without Combiner ?

(a) (i) (a,1) (a,5) (b,2) (b,7)(c,3) (c,6) (c,2) (c,8)
(ii) (a,1) (b,2) (c,3) (c,6) (a,5) (c,2) (b,7) (c,8)

(b) (i) (a,[1,5]) (b,[2,7]) (c,[3,6,2,8])
(ii) (a,1) (a,5) (b,2) (b,7)(c,3) (c,6) (c,2) (c,8)

(c) (i) (a,[1,5]) (b,[2,7]) (c,[2,9,8]
(ii) (a,[1,5]) (b,[2,7]) (c,[3,6,2,8])

(d) (i) (a,6) (b,9) (c,18)
(ii) (a,[1,5]) (b,[2,7]) (c,[3,6,2,8])

39. Consider the two statements below:

(i) Lineage Graph is the result of transformations on RDD.

(ii) Lineage Graph is the result of actions on RDD.

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.

40. Consider the code snippet given below:

```
val Rdd1 = sc.parallelize(List(("a", 1), ("b",2), ("c",3)))
val Rdd2 = sc.parallelize(List(("b", "box"), ("c","cat"), ("d","dog")))
val Rdd3 =
Rdd1.leftOuterJoin(Rdd2)
Rdd3.collect().foreach(println)
```

What will be the output?

(a) (a,(1,None)) (b,(2,box)) (c,(3,cat)) (d,(None,dog))

(b) (a,(1,None))
(b,(2,Some(box)))
(c,(3,Some(cat)))

(c) (a,(1,Some(None)))
(b,(2,Some(box)))
(c,(3,Some(cat)))
(d,(None, Some(dog))

(d) (a,(1,None))
(b,(2,box))
(c,(3,cat))
(d,(None, dog))

41. Under which of the conditions does K-means fail to give good results?

(a) Problems with outlier instances.

(b) Problems with non-convex cluster shapes.

(c) Problems with round cluster shapes

(d) Both (a) and (b)

42. Which one of them is true in terms of data processing?

(a) Hadoop performs both batch and stream processing.

(b) Spark performs both batch and stream processing.

(c) Hadoop performs batch processing only.

(d) Both (b) and (c)