**CSCE 4013/5013 Cloud Computing and Security**

**Quiz #2 (20 points)**

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem #1 (10 pts)**

1. Yes / No Do you participate in this quiz?

**Problem #2 (10 pts)**

1. (3 pts) A programmer is writing a MapReduce program to deal with a set of input files. The sizes of the files are listed in the following table. The size of a split is 128 MB. How many mappers are going to be launched to process these input files? (A)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File 1 | 126 MB |  | File 3 | 54 MB |
| File 2 | 129 MB |  | File 4 | 300 MB |

1. 7
2. 8
3. 5
4. 4

2. (3 pts) For the same set of input files to be processed by a MapReduce program, how many reducers will be launched to process the intermediate data? (D)

1. 7
2. 8
3. 5
4. Cannot be determined from the problem statement

3. (4 pts) For the classic word count program, the map method and the reduce method are shown as follows. There is no in-method combining or in-mapper combining. The reducer class is used as the combiner class.

|  |  |
| --- | --- |
| public void map(Object key, Text value, Context context) throws IOException, InterruptedException {  StringTokenizer itr = new StringTokenizer(value.toString());  while (itr.hasMoreTokens()) {  word.set(itr.nextToken());  context.write(word, one);  }  } | public void reduce(Text key, Iterable<IntWritable>  values, Context context) throws IOException,  InterruptedException {  int sum = 0;  for (IntWritable val : values) {  sum += val.get();  }  result.set(sum);  context.write(key, result);  } |

We only process one single file. The content of the file is as follows.

Alice goes to wonderland to collect flowers

Alice goes to wonderland to have fun

How many intermediate pairs of (k2, v2) are there after the local aggregation and before the shuffle and sort phase? (D)

1. 14
2. 8
3. 10
4. Cannot be determined