Section 8

MapReduce and Spark

Assume for these problems that you have a relation:

SampleDNA(sid, symptomatic, sequence, located)

Sequences are of, at most, length 2000.
All sequences only contain the nucleotides A, T, G, and C.

SampleDNA			
sid	symptomatic	sequence	located
49396937	T	ATTTCGATGCGCGTAAA...	Seattle WA
68478053	F	ATTCCGATGCGCGAAAA.. .	Boston MA
...	...	...	...

```
// attribute indexes
final int SID = 0;
final int SYMPTOMATIC = 1;
final int SEQUENCE = 2;
final int LOCATED = 3;
```

For each of the problems write a Spark function to compute the result. Assume you are given an RDD r. Compute the information (don't worry about outputting).

Here is a guide to the Spark objects and methods you might consider using:

Java Spark Objects	**Java Spark Methods**
Row RowFactory.create(Objects...) Dataset<Row> JavaRDD<Row> JavaPairRDD<K, V> Tuple2<>	d.filter(t -> f(t) == true/false) d.distinct() d.map(t -> RowFactory.create(A1, A2, …, An)) d.mapToPair(t -> new Tuple2<>(K, V)) d.reduceByKey((v1, v2) -> f(v1, v2)) d.max(Comparator) d.min(Comparator)

1. Count number of symptomatic samples that were found in each location. Get the max count (you don't need the associated location).

2. Output the relative frequency of each nucleotide for each sequence. Your results should look like the following:

nucleotide	rel_freq
A	0.235
C	0.41
G	0.29
T	0.265
...	...