Data Intensive Computing - Review Questions 2

Deadline: September 19, 2020

1. **1 point.** Briefly explain how you can use MapReduce to compute the multiplication of a $M \times N$ matrix and $N \times 1$. You don't need to write any code and just explain what the *map* and *reduce* functions should do.

---

2. **1 point.** What is the problem of skewed data in MapReduce? (when the key distribution is skewed)

---

3. **1 point.** Briefly explain the differences between Map-side join and Reduce-side join in Map-Reduce?

---

4. **1 point.** Explain briefly why the following code does not work correctly on a cluster of computers. How can we fix it?

```
val uni = sc.parallelize(Seq(("SICS", 1), ("KTH", 2)))
uni.foreach(println)
```

---

5. **1 point.** Assume you are reading the file `campus.txt` from HDFS with the following format:

```
SICS CSL
KTH CSC
UCL NET
SICS DNA
...
```

Draw the lineage graph for the following code and explain how Spark uses the lineage graph to handle failures.

```
val file = sc.textFile("hdfs://campus.txt")
val pairs = file.map(x => (x.split(" ")(0), x.split(" ")(1)))
val groups = pairs.groupByKey()
val uni = sc.parallelize(Seq(("SICS", 1), ("KTH", 2)))
val joins = groups.join(uni)
val sics = joins.filter(x => x.contins("SICS"))
val list = sics.map(x => x._2)
vali result = list.count
```