# CS226 Assignment 2 report

861310198 Tsung-Ying Chen

1. Which InputFormat did you use in the MapReduce program?

   FileInputFormat

   instruction:

   hadoop jar target/assignment2-1.0-SNAPSHOT.jar edu.ucr.cs.cs226.tchen063.KNN <input file> <output path> <K> <query point x> <query point y>

2. What is the input and output format of the map function?

```
public static class KnnMapper extends Mapper<Object, Text, NullWritable, IdDistance> {

    IdDistance outVal = new IdDistance();
    TreeMap<Double, String> KnnMap = new TreeMap<Double, String>();

    public void map(Object key, Text value, Context context) throws IOException, Interru

        String[] parts = value.toString().split(",");
        String id = parts[0];
        double x = Double.parseDouble(parts[1]);
        double y = Double.parseDouble(parts[2]);
        Double distance = Math.sqrt((qX - x) * (qX - x) + (qY - y) * (qY - y));
        KnnMap.put(distance, id);

        System.out.println(KnnMap);
    }
    @Override
    protected void cleanup(Context context) throws IOException, InterruptedException{
        for(Map.Entry<Double, String> entry : KnnMap.entrySet())
        {
              Double knnDist = entry.getKey();
              String knnId = entry.getValue();
              outVal.set(knnDist, knnId);
              context.write(NullWritable.get(), outVal);
        }
    }
}
```

| | input | output |
|---|---|---|
| key | Object | Null |
| value | Text (one sample) | IdDistance.class |

3. What is the logic of the map function?

   Calculate the distance of query point and each point at set P. Then output the point ID with the distance by a custom class IdDistance.class.

4. If a combiner function is used, what is the signature of the combiner function (input and output) and what is its logic?

   Same as reduce function.

5. If a reduce function is used, what is the signature of the reduce function (input and output) and what is its logic?

```
public static class KnnReducer extends Reducer<NullWritable, IdDistance, NullWritable, Text>{

    TreeMap<Double, String> KnnMap = new TreeMap<Double, String>();

    @Override
    public void reduce(NullWritable key, Iterable<IdDistance> values, Context context) throws I
        InterruptedException
    {
        for (IdDistance val : values){
            String id = val.getId();
            double tDist = val.getDistance();

            KnnMap.put(tDist, id);
            if (KnnMap.size() > K)
                KnnMap.remove(KnnMap.lastKey());
        }
        context.write(NullWritable.get(), new Text(KnnMap.toString()));
    }
```

   Use TreeMap to save points so that the points are sorted by their distance. And when the TreeMap size bigger than the selected K, remove the farthest point.

6. How many mappers and reducers are needed for your program?

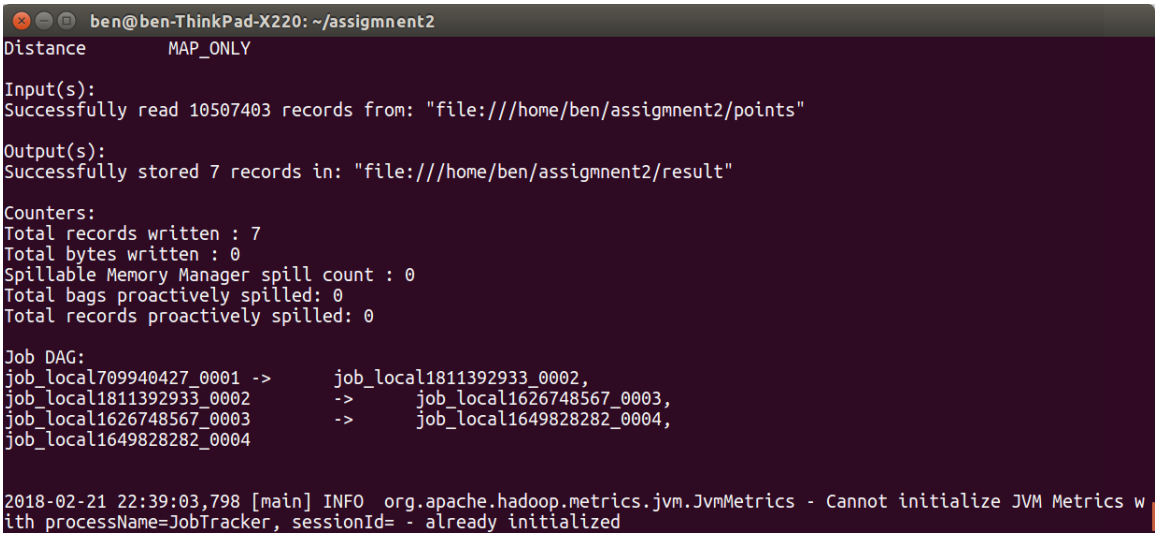
   mappers: 4

   reducer: 1

7. How many records are shuffled between the mappers and reducers?

   4

8. For the Pig Latin program, how many MapReduce jobs are needed to run the program? How does this compare to the MapReduce implementation?




As the figure showed, Pig Latin program used four jobs.