CSE 6242 HW3

**Q1a Description**

MapReduce works in two phases: the map phase and the reduce phase. In each phase, we need to specify the key-value pairs as input and output for the map function and reduce function. Map function can read the input file line by line and save each line as key-value pairs, while reduce function will load the pairs that generated from the map function one by one and calculate and write the result into the output file.

The objective of this program is to report the largest weight among all the weighted outbound edges for each node in the graph. The job is accomplished by a map function and reduce function, which is introduced below.

**Map function**: take a line “100 10 7” as an example to illustrate the function.

Step 1. Load one line as string: “100 10 7”

Step 2. Split the one line string by “\t” as [“100”, “10”, “7”]

Step 3. Specify the key-value pairs: key-“100”, value-“7”

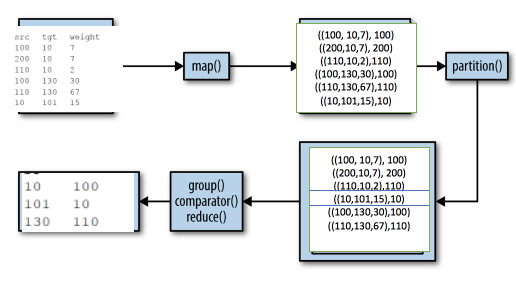
**Reduce function**: take key “100” as an example to illustrate the function.

Step 1. The reduce function sees the following input: (100, [7, 30])

Step 2. Iterate through the list and pick up the maximum reading: (100, 30)

The third part of the program is the code to run the MapReduce job. The Job’s setJarByClass() method is used to locate the relevant JAR file by looking for the JAR file containing this class. The input and output paths is specified by addInputPath() on FileInputFormat, and addOutputPath() on FileOutputFormat. The map and reduce type is specified by using setMapperClass() and setReducerClass() methods.  The setOutputKeyClass()  and setOutputValueClass()  methods control the output types for the map and the reduce functions, which are often the same, as they are in our case. If they are different, then the map output types can be set using the methods setMapOutputKeyClass()  and setMapOutputValueClass().

**Q1b Description**



The procedure of the Q1b is descripted in the figure above.