import java.io.IOException;

import java.util.ArrayList;

import java.util.Collections;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class MaxTemperatureReducer

extends Reducer<Text, IntWritable, Text, Text> {

ArrayList<Integer> tempList = new ArrayList<Integer>();

@Override

public void reduce(Text key, Iterable<IntWritable> values,

Context context)

throws IOException, InterruptedException {

double median = 0;

double total=0.0;

double MEAN = 0.0;

double temp= 0.0;

double num = 0.0;

double STDDEV = 0.0;

int maxValue = Integer.MIN\_VALUE;

int minValue = Integer.MAX\_VALUE;

for (IntWritable value : values) {

maxValue = Math.max(maxValue, value.get());

minValue = Math.min(minValue, value.get());

tempList.add(value.get());

}

Collections.sort(tempList);

int size = tempList.size();

for (int i : tempList) {

total += i;

}

MEAN=total/size;

for (int i : tempList) {

num = Math.pow(((double) i - MEAN), 2);

temp += num;

}

STDDEV =Math.sqrt(temp/size);

if(size%2 == 0){

int half = size/2;

median = tempList.get(half);

}else {

int half = (size + 1)/2;

median = tempList.get(half -1);

}

String maxminavg = Integer.toString(maxValue) + " " +Integer.toString(minValue)+" "+Double.toString(MEAN)+" "+Double.toString(median)+" "+Double.toString(STDDEV);

context.write(key, new Text(maxminavg));

}

}