import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.Scanner;

/\*\*

\* for a given shift, if two equals strings get the same hashCode we have a collision

\* @author Richein Metellus

\* @version 03/31/2017

\*/

public class Client {

public static void main(String[] args) throws FileNotFoundException {

ArrayList<String> stockData = new ArrayList<>();

//files for Testing

File amex = new File("C:\\Users\\Rich\\Documents\\EODdata-Symbols-20110305\\AMEX.txt");

File forex = new File("C:\\Users\\Rich\\Documents\\EODdata-Symbols-20110305\\FOREX.txt");

File index = new File("C:\\Users\\Rich\\Documents\\EODdata-Symbols-20110305\\INDEX.txt");

File nasdaq = new File("C:\\Users\\Rich\\Documents\\EODdata-Symbols-20110305\\NASDAQ.txt");

File nyse = new File("C:\\Users\\Rich\\Documents\\EODdata-Symbols-20110305\\NYSE.txt");

File otcbb = new File("C:\\Users\\Rich\\Documents\\EODdata-Symbols-20110305\\OTCBB.txt");

// read the file specified and add each ticker into the arrayList specified

tickerToArrayList(amex,stockData);

tickerToArrayList(forex,stockData);

tickerToArrayList(index,stockData);

tickerToArrayList(nasdaq,stockData);

tickerToArrayList(nyse,stockData);

tickerToArrayList(otcbb,stockData);

System.out.println(" | Collision |");

System.out.println("+-------------+-------------------+----------+");

System.out.println("| Shift | Total | Max +");

System.out.println("+-------------+-------------------+----------+");

for(int shift = 0; shift < 17; shift++)

{

int[][] hashCode\_Collision = new int[stockData.size()][2];

int indexPosition = 0; // index where to incremment the collision count.

int hashCodeCount = 0; // will keep tab of the number all the unique hasCode added to the 2 dim array. size of array[0][..]

int tickerHashCode;

for(String ticker: stockData)

{

tickerHashCode = hashCode(ticker,shift);

indexPosition = contains(hashCode\_Collision, hashCodeCount,tickerHashCode,0); // return the position/index if the the hashCode array contain the tickerHashcode

if(indexPosition != -1)

(hashCode\_Collision[indexPosition][1])++; //if hashcode already in the array, increase collision array at that index

else

{

hashCode\_Collision[hashCodeCount][0] = tickerHashCode ;

hashCodeCount++;

}

}

int totalCollision = 0;

int max = hashCode\_Collision[0][1];

for(int i = 0; i < hashCodeCount; i++)

{

totalCollision += hashCode\_Collision[i][1];

if (max < hashCode\_Collision[i][1])

max = hashCode\_Collision[i][1];

}

System.out.printf("| %2d | %,11d | %7d |%n", shift, totalCollision, max);

System.out.println("+-------------+-------------------+----------+");

}

}

public static int hashCode(String s, int shiftAmount)

{

int h = 0;

for(int i = 0; i< s.length(); i++)

{

h = (h << shiftAmount) | (h >>> 32- shiftAmount);

h += (int) s.charAt(i);

}

return h;

}

// this utility method add each ticker from lines of the file into the ArrayList Specified by the user

public static void tickerToArrayList(File file, ArrayList<String> Bag) throws FileNotFoundException

{

Scanner fileScanner = new Scanner(file);

boolean firstLine = true;

while(fileScanner.hasNextLine())

{

String ticker;

if(firstLine)

{

fileScanner.nextLine(); // to skip the first line

firstLine = false; // so only execute once.

}

else

{

String stockLine = fileScanner.nextLine();

Scanner stockLineScan = new Scanner(stockLine);

stockLineScan.useDelimiter("\t");

if(stockLineScan.hasNext())

{

ticker = stockLineScan.next();

Bag.add(ticker);

stockLineScan.nextLine(); // to skip description part of the ticker

}

}

}

}

public static int contains(int[][] array, int n, int targetNum, int column)

{

for(int i = 0; i < n; i++)

if(array[i][column] == targetNum){

return i;

}

return -1; // if not found, can't have index -1 so reasonable.

}

}

