

import java.io.FileNotFoundException;

import java.io.PrintWriter;

import java.util.Random;

public class permutationRandomString {

public static String generateString(String characters, int length)

{

Random randi = new Random();

char[] text = new char[length];

for (int i = 0; i < length; i++) {

text[i] = characters.charAt(randi.nextInt(characters.length()));

}

return new String(text);

}

void doPermutation(char [] str,int i,int n){

if (i==n) {

for (int m = 0; m < str.length; m++) {

System.out.print(str[m]);

}

System.out.println("");

} // end if

else{

for (int j = i; j <=n; j++) {

//swap str[i] and str[j]

char ch=str[i];

str[i]=str[j];

str[j]=ch;

doPermutation(str, i+1, n);

//swap str[i] and str[j]

ch=str[i];

str[i]=str[j];

str[j]=ch;

} // end for

} // end else

}

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

// Construct a PrintWriter for the output file

PrintWriter out = new PrintWriter("RuntimePerm.csv");

// Writes the header

out.println("Permutation Runtime O(2^n)");

out.println("n,time(ms)");

for (int n = 0; n < 11; n++)

{

// Use stopwatch to time selection sort

StopWatch timer = new StopWatch();

timer.start(); // Starts timer

// Generates Permutations

String str1=generateString("abcdedfghijklmnopqrstuvwxyz",n);

permutationRandomString thing=new permutationRandomString();

char []str=str1.toCharArray();

thing.doPermutation(str,0,(str.length-1));

timer.stop(); // Stops timer

// Prints n and time elapsed

out.println(n + "," + timer.getElapsedTime());

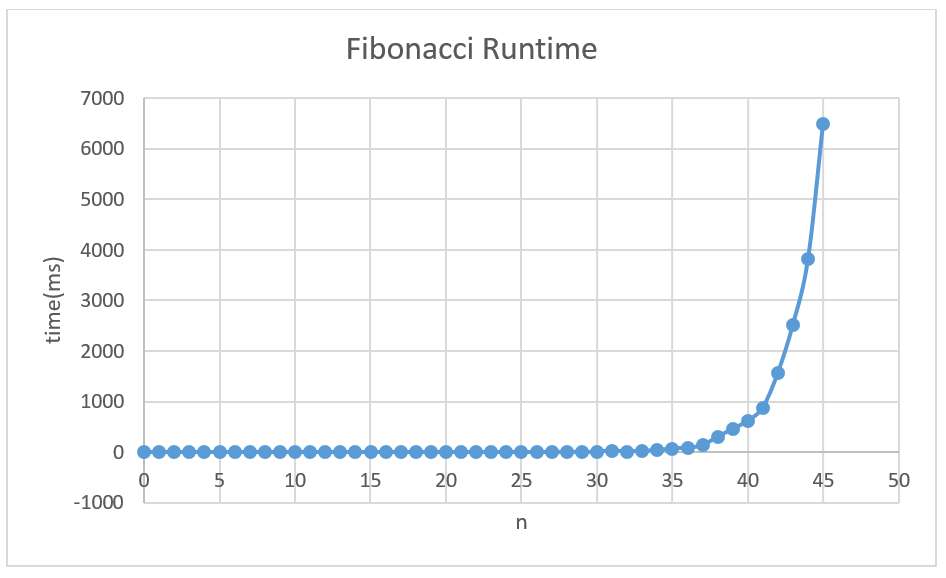
}

// Closes data file

out.close();

}

}



import java.io.FileNotFoundException;

import java.io.PrintWriter;

import java.util.Scanner;

/\*\*

This program computes Fibonacci numbers using a recursive method.

\*/

public class RecursiveFib

{

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

{

// Construct a PrintWriter for the output file

PrintWriter out = new PrintWriter("RuntimeFib.csv");

// Writes the header

out.println("Fibonacci Runtime");

out.println("n,time(ms)");

for (int n = 0; n < 46; n++)

{

// Use stopwatch to time selection sort

StopWatch timer = new StopWatch();

timer.start(); // Starts timer

// Generates fibonacci number

long f = fib(n);

System.out.println("fib(" + n + ") = " + f);

timer.stop(); // Stops timer

// Prints n and time elapsed

out.println(n + "," + timer.getElapsedTime());

}

// Closes data file

out.close();

}

/\*\*

Computes a Fibonacci number.

@param n an integer

@return the nth Fibonacci number

\*/

public static long fib(int n)

{

if (n <= 2) { return 1; }

else { return fib(n - 1) + fib(n - 2); }

}

}