Share 0 More Next Blog»

I, ME AND MYSELF !!!

THURSDAY, JANUARY 13, 2011

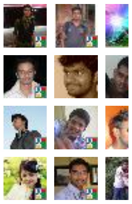
Pollard's Rho in Java

This is a Pollard's Rho implementation in java. Not very fast, but works for uva online judge. The reason behind using java is the default support of the BigInteger class.

```
import java.math.BigInteger;
import java.security.SecureRandom;
import java.io.*;
import java.util.*;

public class PollardRho {

    private final static BigInteger ZERO = new BigInteger("0");
    private final static BigInteger ONE  = new BigInteger("1");
    private final static BigInteger TWO  = new BigInteger("2");
    private final static SecureRandom random = new SecureRandom();

    static Vector<BigInteger> v = new Vector<BigInteger>();

    public static BigInteger rho(BigInteger N) {

        BigInteger divisor;
        BigInteger c  = new BigInteger(N.bitLength(), random);
        BigInteger x  = new BigInteger(N.bitLength(), random);
        BigInteger xx = x;

        if (N.mod(TWO).compareTo(ZERO) == 0) return TWO;

        do {
            x  =  x.multiply(x).mod(N).add(c).mod(N);
            xx = xx.multiply(xx).mod(N).add(c).mod(N);
            xx = xx.multiply(xx).mod(N).add(c).mod(N);
            divisor = x.subtract(xx).gcd(N);
        } while((divisor.compareTo(ONE)) == 0);

        return divisor;
    }

    public static void factor(BigInteger N) {

        if (N.compareTo(ONE) == 0) return;

        if (N.isProbablePrime(20)) {
            v.add(N);
            return;
        }

        BigInteger divisor = rho(N);
        factor(divisor);
        factor(N.divide(divisor));
    }

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

        String string = "";
        InputStreamReader input = new InputStreamReader(System.in);
        BufferedReader reader = new BufferedReader(input);
```

Join this s
with Google Friend

Members (198)



Already a member?

SUBSCRIBE

Posts

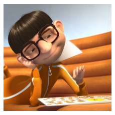
Comments

BLOG HITS

BLOG ARCHIV

- ► 2014 (6)
- ► 2013 (19)
- ► 2012 (14)
- ▼ 2011 (15)
  - ► July (3)
  - ► May (1)
  - ► March (3)
  - ▼ January (
    - A C++ Mis
    - Huffman's
    - Pollard's F
    - Java Acce
    - Blogger S
    - Story of C
    - SPOJ - PC
    - CSE 202 F
- ► 2010 (33)
- ► 2009 (27)

ABOUT ME



V
p

```
        while(null != (string = reader.readLine())) {
            BigInteger N = new BigInteger(string);
            v.clear();
            factor(N);
            Collections.sort(v);
            for(int i = 0; i < v.size(); i++) System.out.println(v.get(i));
            System.out.println();
        }
    }
}
```

I have seen this piece of code years ago somewhere in the internet, but can't remember exactly where. So, I will be glad if anyone can comment/mail me the original source. However, for spoj, I need to write a much much better version of this in C++ :(Exam sucks life...

Posted by Zobayer Hasan at 3:41 AM

3 comments:

Masum February 28, 2012 at 2:33 PM

I think by this time a C++ code should be posted. :)

Reply

Anonymous September 3, 2012 at 2:16 AM

this is the code from Robert Sedgewick's page "intro to cs".

Reply

Replies

Zobayer Hasan September 3, 2012 at 3:44 AM

Haven't read that book yet, but thanks for the info :)

Reply

Enter your comment...

Comment as: Google Accour ▼

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

CATAGORIES

academic study (23) access modifiers (1) algorithm (50) analysis (6) apache (1) backtrack (1) bash (1) beginner (17) bfs (2) bigint (1) binary indexed tree (2) binary tree (1) bit (1)
blogger (5) bpm (2) brainfuck (1) brute force (1) bst (1) c (5) C++ (41) changes (1) character device driver (1) chat (3) client (3) combinatorics (2) command prompt (1) common (1) comparator (1) comp
geometry (2) confusion (1) connected component (1) console (1) constructible polygon (1) contest (11) crc (1) cse (5) css (1) customize (1) data mining (2) data structure (16) database (3) DCEPC206
device driver (1) dfs (1) disjoint set (1) divide and conquer (3) dp (3) driver (1) dual boot (1) dynamic programming (9) encoding (1) encryption (1) error (2) esoteric language (2) euler circuit (3) euler path
(1) expression evaluation (1) extended euclid (1) facebook (3) factorization (2) fibonacci (1) fix time (1) function (1) funny (15) gcd (2) geometry (5) git (3) github (2) gns3 (2) graph (9) GUANGGUN (1) h
hiding w indow (1) hints (15) holi (1) hopcroft karp (1) huffman (1) incorrect clock (1) inner class (1) instance (1) jar (1) java (8) javascript (1) jdbc (1) kernel programming (2) lab (5) lazy propagation (1) like (1
(6) ls (1) makefile (1) math (21) matrix (3) matrix algebra (2) matrix exponentiation (2) matrix multiplication (2) maxflow (1) maximum bipartite matching (2) maximum flow (1) merge sort (3) mistake (
module compiling (2) multichat (3) mysql (1) networking (2) number system (1) number theory (8) online judge (4) operating system (1) os (1) other (8) parallel programming (3) pattern (1) phi (1)

practice (1) primes (5) primit (1) priority queue (1) problem (17) problem classifier (2) problem solving (54) problems solving (1) programming (68) pruning (1) pthreade
qualification round (1) queen (1) range maximum query (1) recursion (6) reflection (1) repository (4) rip (1) rmq (1) segment tree (2) server (3) shell (1) shell script (1) sieve (4) simulation (3) socket (3) soluti
sphere online judge (27) spoj (27) static routing (1) syntax highlighting (1) system programming (4) table tag (1) tc (1) template (4) thread (3) time mismatch (1) time setting (1) topcoder (2) topology (
tree (3) tutorial (9) ubuntu (1) usaco (2) uva (5) uva online judge (5) vector (1) version control (1) w eb server (1) windows (3)

I am only one, but still I am one.
I cannot do everything, but still I can do something.
And because I cannot do everything I will not refuse to do the something that I can do.
{Helen Keller}

Picture Window template. Template images by Ollustrator. Powered by Blogger.