暗号の数理レポート #2

2011–1–27　　河野

課題 Miller-Rabin 法 により，10 進数で 47 桁のランダムな素数を，確率 $1 - \frac{1}{10^9}$ で生成する Java プログラムを作成せよ．

- ランダムな素数とは，そのプログラムを実行するたびに，10 進数で 47 桁，すなわち $10^{46} \sim 10^{47} - 1$ の範囲にある素数がランダムに出力される，ということ。
- プログラムはワークステーション室の環境で動くことを確認すること。
- Java program には BigInteger クラスを使うのが簡単である。
  http://java.sun.com/j2se/1.5.0/ja/docs/ja/api/java/math/BigInteger.html 参照。
- BigInteger(int numBits, Random rnd) のより $0 \sim 2^{\mathrm{numBits}}$ の範囲の整数がランダムに生成される。例えば

```
//////////////////////////////////////////////////////
import java.math.BigInteger;
import java.util.Random;
import java.io. \uff0a ;
public class BigPrime {
  public static void main (String args[ ]){
    BigInteger p = new BigInteger ( 200, new Random( ) );
    ..................................................
  }
}
//////////////////////////////////////////////////////
```

  などとすると $0 \sim 2^{200}$ の範囲の整数 p がランダムに生成される。

- BigInteger クラスには BigInteger(int bitLength, int certainty, Random rnd) という constructor もあり，これは，指定された bitLength を持つ素数を確率 certainty で生成する。当然，これを使ってはいけない。また，同様の働きをする
  probablePrime(int bitLength, Random rnd)
  nextProbablePrime()
  という method もあるが，当然，これも使ってはいけない．

  さらに isProbablePrime(int certainty) という，与えられた整数が素数であるかどうかを判定するための method があるが，当然，これを本質的に使ってはいけない。素数の判定は，Miller-Rabin 法によって行わなければならないので，例えば，ランダムに数を取ってこの method で判定する，という方法はダメである。ただし，自分が作ったプログラムがどの程度の性能を持っているのか，をチェックするのに使うのは構わない．

  これら以外の constructor, method については，どれを使っても良い．特に，
  gcd(BigInteger val)
  modPow(BigInteger exponent, BigInteger m)
  という method は役に立つ．

- プログラムを作成した人は (1) 簡単なアルゴリズムの説明，(2) ワークステーション室で使用する人のためのマニュアル (何をするプログラムか，どの様に実行するか，を含む)，(3) 特記事項 (後の項を参考) があればそれを mail の本文にそれぞれ独立した項目として書き，プログラムを独立したファイル (そのままで実行可能な状態のファイル) として添付して crypto@math.cs.kitami-it.ac.jp まで送ること。勿論名前，学生番号も忘れずに。
- ただし「実行可能」の意味は次の通り；一連の操作が必要な場合はその手順を順に記したものでもよい。例えば実行には subdirectory XXX にある file1,file2,file3 が必要で，そのファイルが存在した場合 exec-file という名前のファイルが実行可能とする。file1,file2,file3 を例えば zip でまとめ yyy とする。unzip yyy とすると subdirectory XXX にされるように設定してあるとき，ファイルとして exec-file と yyy を添付し，リストとして,「unzip yyy; ./exec-file」と書いたものを沿えて提出。ただし unzip 等が任意の directory で実行可能でないときは full-path で書くこと。shell script が書ける人は shell script の形でもよい (というかその方が私は楽です)。
- user interface はどのようなものでもかまわないが，自分で使用する場合使い易いようなものにすることを考慮すること。
- 前項のことの他何か自分のプログラムの「売り」があれば特記事項として記載すること。水準に達しているレポートには 60 点が与えられる。これに加え「売り」の内容に応じて $5 \times n$ (n は自然数) 点が加点される。
- ただしプログラム･説明文を含めて自分で書いたと思われないレポート (ネットからの copy&paste を含む) ないし 他のレポートと同一と判断されるレポートは採点の対象にはならない。
- 締切は 3 月 31 日とする。