**Sequence Analysis Lab 3**

<https://en.wikipedia.org/wiki/Miller%E2%80%93Rabin_primality_test>

**Document Id 32**

**Practical Task Probabilistic Tests for Large Primes**

Using the pseudo code given below and the above link implement the

Miller–Rabin primality test to discover if **2932031007403** is a prime number

We define some parameters k is an integer probability parameter >= 1 ,

s is an integer exponent of 2 >= 1 and d is an integer odd number >= 1

and **n** the prime to test

Step 1 eliminate numbers <= 3 return true (is a prime)

Step 2 assign the probability parameter say 10

Step 3 using the fact **n** **-1** is even then for the expression **n** **-1 =**  2*s*·*d*, find s and d

Something like

while (d mod 2 == 0) {

s++;

d = d/2;

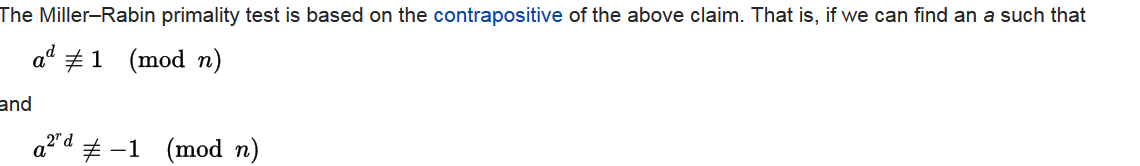
}

Step 4 in a loop for 0 to k using parameter k, for each iteration generate a

random integer **a** in the range of [2, **n** **-1**]

Step 5 is based on the formulas below, we have **a and d, we choose r from**

**0 ≤ r ≤ *s* − 1.**



We test the random integer with the expressions above, if the random integer satisfies the expressions then **p** is not prime

Suggestion

Use java’s BigInteger class

BigInteger a = generated random Integer

a.modPow(d, n).equals(BigInteger.***ONE***)