Package 'bigIntegerAlgos'

April 25, 2018

Type Package

Title R Tool for Factoring Big Integers

Version 0.1.1

Author Joseph Wood

Maintainer Joseph Wood <jwood000@gmail.com>

Description Features the multiple polynomial quadratic sieve algorithm for factoring large integers and a vectorized factoring function that returns the complete factorization of an integer. Utilizes the C library GMP (GNU Multiple Precision Arithmetic) and classes created by Antoine Lucas et al. found in the 'gmp' package.

License GPL (>= 2)

BuildResaveData no

Encoding UTF-8

SystemRequirements gmp (>= 4.2.3)

Suggests testthat, numbers (>= 0.6-6), RcppAlgos (>= 1.0.1)

Depends gmp

NeedsCompilation yes

URL https://github.com/jwood000/bigIntegerAlgos, http://mathworld.wolfram.com/QuadraticSieve.html

BugReports https://github.com/jwood000/bigIntegerAlgos/issues

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2018-04-25 16:20:36 UTC

R topics documented:

divisorsBig *Vectorized Factorization (Complete) with GMP*

Description

Quickly generates the complete factorization for many (possibly large) numbers.

Usage

```
divisorsBig(v, namedList = FALSE)
```

Arguments

v	Vector of integers, numerics, string values, or elements of class bigz.
namedList	Logical flag. If TRUE and the length(v) > 1, a named list is returned. The default is FALSE.

Details

Highly optimized algorithm to generate the complete factorization for many numbers. It is built specifically for the data type that is used in the gmp library (i.e. mpz_t).

The main part of this algorithm is essentially the same algorithm that is implemented in divisorsRcpp from the RcppAlgos package. A modified merge sort algorithm is implemented to better deal with the mpz_t data type. This algorithm avoids directly swapping elements of the main factor array of type mpz_t but instead generates a vector of indexing integers for ordering.

See this stack overflow post for examples and benchmarks : R Function for returning ALL factors.

Value

- Returns an unnamed vector of class bigz if length(v) == 1 regardless of the value of namedList.
- If length(v) > 1, a named/unnamed list of vectors of class bigz will be returned.

Author(s)

Joseph Wood

References

Divisor

See Also

divisorsRcpp, divisors, factorize

Examples

```
## Get the complete factorization of a single number
divisorsBig(10^15)

## Or get the complete factorization of many numbers
set.seed(29)
myVec <- sample(-1000000:1000000, 1000)
system.time(myFacs <- divisorsBig(myVec))

## Return named list
myFacsWithNames <- divisorsBig(myVec, namedList = TRUE)
```

`quadraticSieve` *Prime Factorization with the Quadratic Sieve*

Description

Get the prime factorization of a number, n, using the Quadratic Sieve.

Usage

```
quadraticSieve(n)
```

Arguments

`n`	An integer, numeric, string value, or an element of class bigz.

Details

First, trial division is carried out to remove small prime numbers, then a modified version of Pollard's rho algorithm that is constrained is called to quickly remove further prime numbers. Next, we check to make sure that we are not passing a perfect power to the main quadratic sieve algorithm. After removing any perfect powers, we finally call the quadratic sieve with multiple polynomials in a recursive fashion until we have completely factored our number.

Value

Vector of class bigz

Author(s)

Joseph Wood

References

- Pomerance, C. (2008). Smooth numbers and the quadratic sieve. In Algorithmic Number Theory Lattices, Number Fields, Curves and Cryptography (pp. 69-81). Cambridge: Cambridge University Press.
- Silverman, R. D. (1987). The Multiple Polynomial Quadratic Sieve. Mathematics of Computation, 48(177), 329-339. doi:10.2307/2007894
- Integer Factorization using the Quadratic Sieve
- From https://codegolf.stackexchange.com/ (Credit to user primo for answer) P., & Chowdhury, S. (2012, October 7). Fastest semiprime factorization. Retrieved October 06, 2017

See Also

`factorize`

Examples

```
mySemiPrime <- prod(nextprime(urand.bigz(2, 40, 17)))
quadraticSieve(mySemiPrime)
```

Index