

## Number Theory

The [NumberTheory](#) package updates and replaces the [numtheory](#) package. There are several new applications and examples that use the new NumberTheory commands and, in addition, many commands have been added to the context menu.

*with(NumberTheory)*

[*AreCoprime, CalkinWilfSequence, CarmichaelLambda, ContinuedFraction, ContinuedFractionPolynomial, CyclotomicPolynomial, Divisors, FactorNormEuclidean, HomogeneousDiophantine, ImaginaryUnit, InhomogeneousDiophantine, IntegralBasis, InverseTotient, IsCyclotomicPolynomial, IsMersenne, IsSquareFree, IthMersenne, JacobiSymbol, KroneckerSymbol, Landau, LargestNthPower, LegendreSymbol, Möbius, ModExtendedGCD, ModularLog, ModularRoot, ModularSquareRoot, Moebius, MultiplicativeOrder, Möbius, NearestLatticePoint, NextSafePrime, NumberOfIrreduciblePolynomials, NumberOfPrimeFactors,  $\Omega$ ,  $\Phi$ , PrimeCounting, PrimeFactors, PrimitiveRoot, PseudoPrimitiveRoot, QuadraticResidue, RepeatingDecimal, RootsOfUnity, SumOfDivisors, SumOfSquares, ThueSolve, Totient,  $\lambda$ ,  $\mu$ ,  $\phi$ ,  $\pi$ ,  $\sigma$ ,  $\tau$ ,  $\varphi$ ]*

### MathApps

 <p>Sieve of Eratosthenes</p>	 <p>GCD and Euclid's Algorithm</p>	 <p>Collatz Conjecture</p>
 <p>Fibonacci Numbers</p>	$4 = 2 + 2$ $8 = 3 + 5$ $12 = 5 + 7$ $16 = 3 + 13$ <p>Goldbach's Conjecture</p>	$  \begin{array}{ccccccc}  & & & & 1 & & & & \\  & & & & 1 & & 1 & & \\  & & & 1 & 2 & 1 & & & \\  & & 1 & 3 & 3 & 1 & & & \\  & 1 & 4 & 6 & 4 & 1 & & & \\  1 & 5 & 10 & 10 & 5 & 1 & & &   \end{array}  $ <p>Pascal's Triangle</p>
$  \begin{array}{l}  196 + 691 = 887 \\  887 + 788 = 1675 \\  1675 + 5761 = 7436 \\  7436 + 6347 = 13783 \\  \vdots  \end{array}  $ <p>The 196 Algorithm</p>	 <p>abc Conjecture</p>	 <p>Euler's Identity</p>
 <p>The Juggler Sequence</p>		

Many **NumberTheory** commands are available in the right-click context menu. For example, fractions can be shown as either a **RepeatingDecimal** or a **ContinuedFraction**:

Repeated Decimal	
1. Enter a fraction. For example, enter <b>3/7</b> .	$\frac{3}{7}$
2. Right-click ( <b>Control</b> -click for Macintosh) the fraction, and then select <b>Show as Repeating Decimal</b> .	$\frac{3}{7}$ repeated decimal = $0.\overline{428571}$

Continued Fraction	
1. Enter a fraction. For example, enter <b>3/7</b> .	$\frac{3}{7}$
2. Right-click ( <b>Control</b> -click for Macintosh) the fraction, and then select <b>Show as Continued Fraction</b> .	$\frac{3}{7}$ continued fraction = $0 + \frac{1}{2 + \frac{1}{3 + 0}}$

## ▼ Examples: Prime Numbers

A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. The top level **ithprime(i)** command returns the **ith** prime. For example, the first ten primes are given by the following sequence:

*seq(ithprime(i), i = 1..10)*

*2, 3, 5, 7, 11, 13, 17, 19, 23, 29*

The top level **isprime** command determines if a given number is prime:

*isprime(28)*

*false*

*isprime(29)*

*true*

The **Divisors** command can also verify that a number is prime. If the divisors of a given integer are only 1 and itself, then the number is prime.

*Divisors*(28)

{1, 2, 4, 7, 14, 28}

*Divisors*(29)

{1, 29}

The **SumOfDivisors** command returns the sum of the divisors of an integer:

*SumOfDivisors*(28)

56

The top level **ifactor** command gives the integer factorization of an integer:

*ifactor*(28)

$(2)^2 (7)$

The **PrimeFactors** command returns a list of factors for a given integer that are primes without multiplicity:

*PrimeFactors*(28)

{2, 7}

The **NumberOfPrimeFactors** command returns the number of Prime Factors of an integer, counted with multiplicity:

*NumberOfPrimeFactors*(28)

3

The top level **nextprime** (or **prevprime**) commands return the next (or previous) prime number after (or before) the given integer:

*nextprime*(29)

31

The **PrimeCounting** command returns the number of primes less than a given integer:

*PrimeCounting*(31)

11

Two integers are relatively prime (coprime) if their greatest common divisor (gcd) is 1. The **AreCoprime** command tests if a sequence of integers or Gaussian integers are coprime:

*AreCoprime*(5, 8)

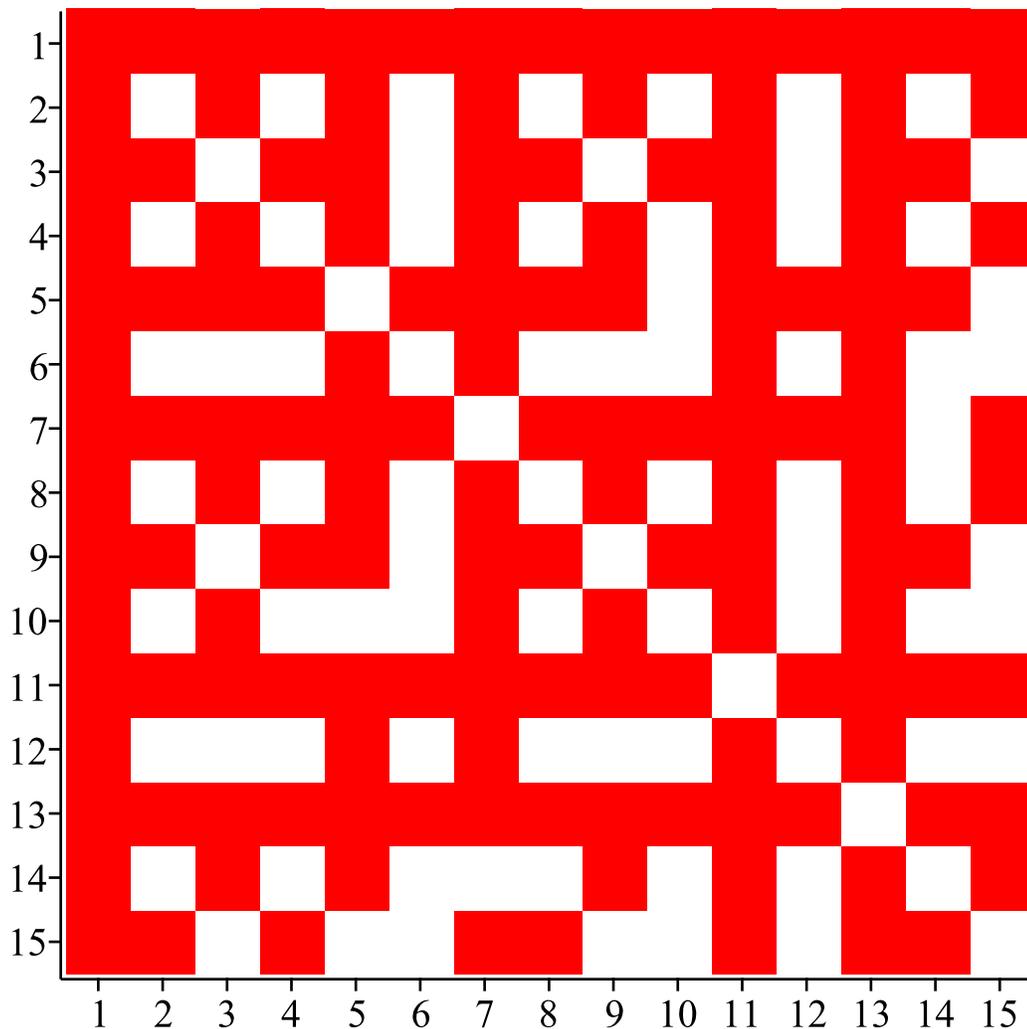
*true*

*AreCoprime*(2, 8)

*false*

The following plot shows the coprimes for the integers 1 to 15:

*Statistics:-HeatMap(Matrix(15, (i,j) → `if`(AreCoprime(i,j), 1, 0)), color = ["White", "Red"]);*



An integer is called square-free if it is not divisible by the square of another number other than 1. All prime numbers are square-free:

*IsSquareFree*(31)

*true*

*IsSquareFree*(7·101)

*true*

*IsSquareFree*(3·7<sup>2</sup>)

*false*

## ▼ Mersenne Primes

Mersenne Primes are prime numbers that are one less than a power of 2. These are numbers of the form:

$$M_n = 2^n - 1, \text{ where } n \text{ is a positive integer.}$$

The *IthMersenne*(i) command returns the exponent for the i<sup>th</sup> Mersenne prime

number:

*IthMersenne*(4)

7

$2^7 - 1$

127

*isprime*( $2^7 - 1$ )

*true*

The *nextprime* command returns the next prime number after the current value.

*nextprime*(127)

131

The *IsMersenne*(*n*) command checks if a positive integer, *n*, is a Mersenne exponent such that  $2^n - 1$  is a Mersenne prime:

$2^2 - 1$

3

*IsMersenne*(2)

*true*

$2^{11} - 1$

2047

*IsMersenne*(11)

*false*

*ifactor*( $2^{11} - 1$ )

(23) (89)

## ▼ Examples: Arithmetic Functions

### ▼ Euler's Totient Function

Euler's totient function is an arithmetic function that counts the positive integers less than or equal to a given value  $n$  that are coprime to  $n$ . For a positive integer  $n$ , another number  $k$  is said to be coprime, or relatively prime, if the greatest common divisor  $\gcd(n, k) = 1$ . For example, 14 and 15 are coprime because the gcd of 14 and 15 is 1, but 14 and 21 are not coprime since their gcd is 7.

Euler's totient function is also known as Euler's phi function, denoted as

$\phi(n)$

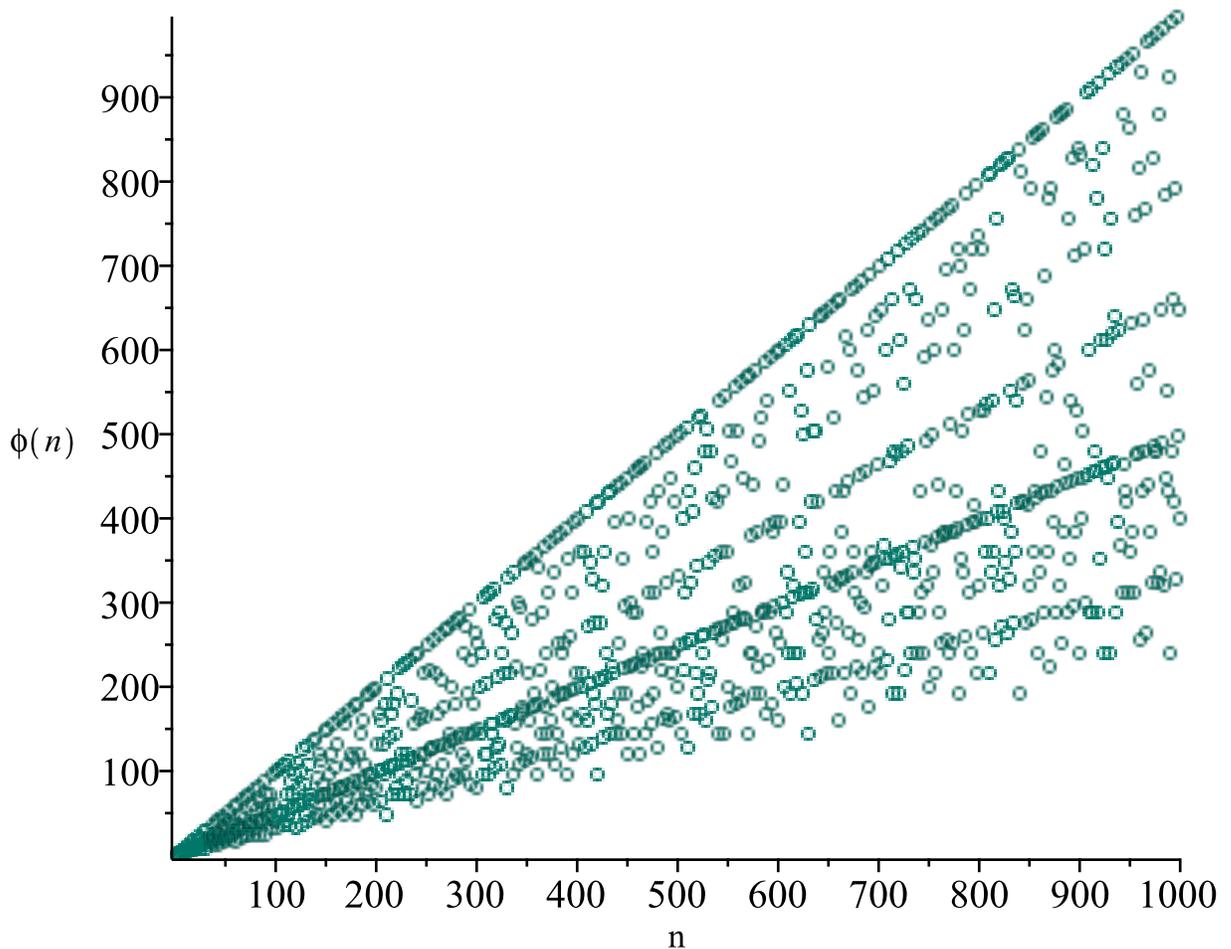
or  $\phi(n)$ . As such,  $\phi(n)$  and  $\text{varphi}(n)$  are aliases for the **Totient** command.

`[Totient(21),  $\phi(21)$ ,  $\varphi(21)$ ]`

`[12, 12, 12]`

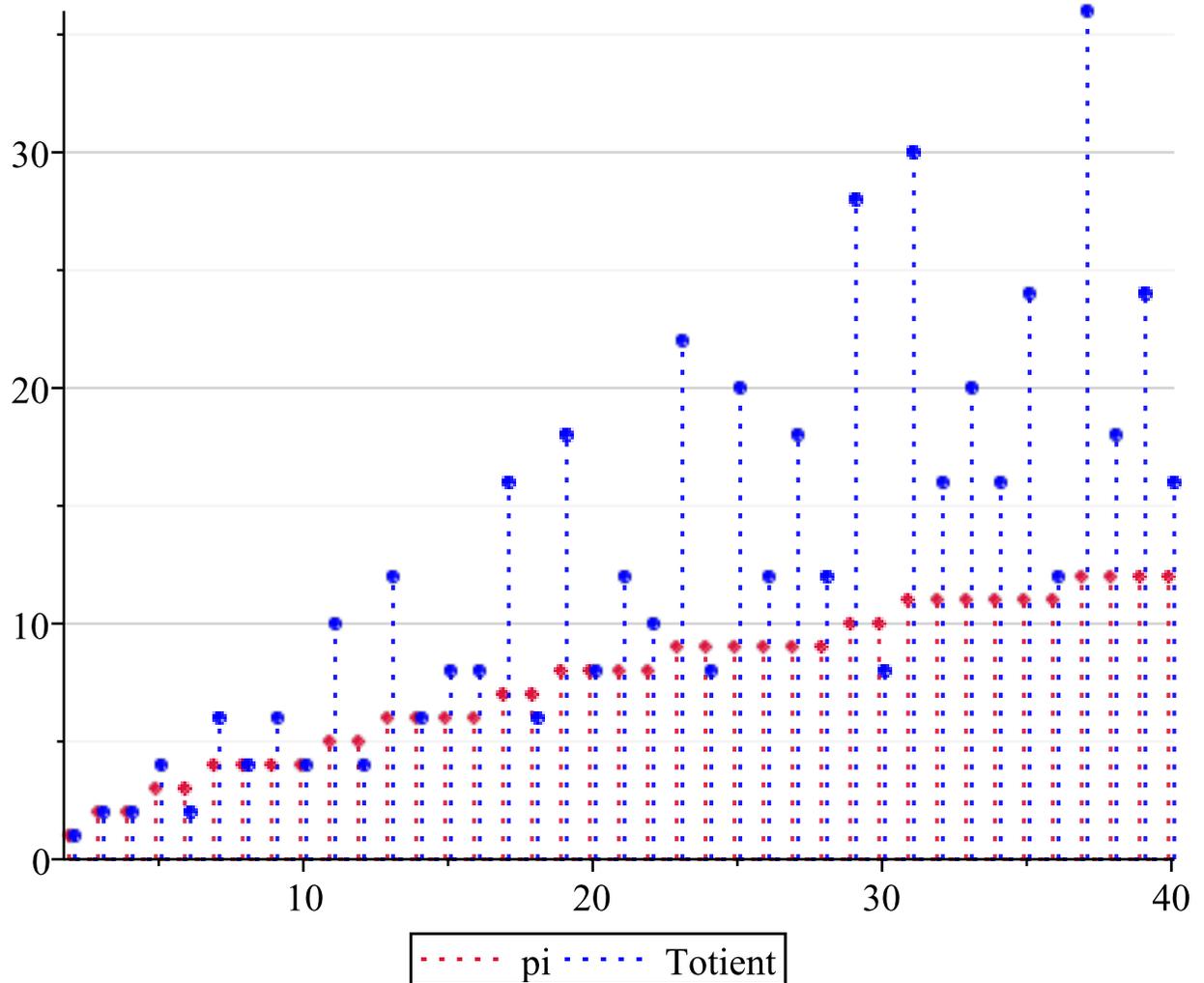
To visualize the first thousand values for  $\phi(n)$ :

```
plots:-pointplot([seq([n, Totient(n)], n = 2 ..1000)], labels = ["n",  $\phi(n)$ ], symbol = circle, color = "Niagara BlueGreen", size = [600, "golden"])
```



The **PrimeCounting(n)** (or **pi(n)**) command returns the number of primes less than an integer **n**. The following plot compares **pi(n)** with **phi(n)** for the first 40 values for **n**:

```
plots:-display( DynamicSystems:-DiscretePlot([seq(i - 0.1, i = 2 ..40)],  
[ seq(PrimeCounting(n), n = 2 ..40)], style = stem, symbol = soliddiamond, color = "Crimson", legend = "pi"),  
DynamicSystems:-DiscretePlot([seq(i + 0.1, i = 2 ..40)], [ seq(Totient(n), n = 2 ..40)], style = stem, symbol = solidcircle, color = "Blue", legend = "Totient"),  
linestyle = dot, transparency = .1, size = [800, 400], axis[2] = [gridlines = [ thickness = 0, color = "LightGrey"]])
```



## ▼ Möbius function

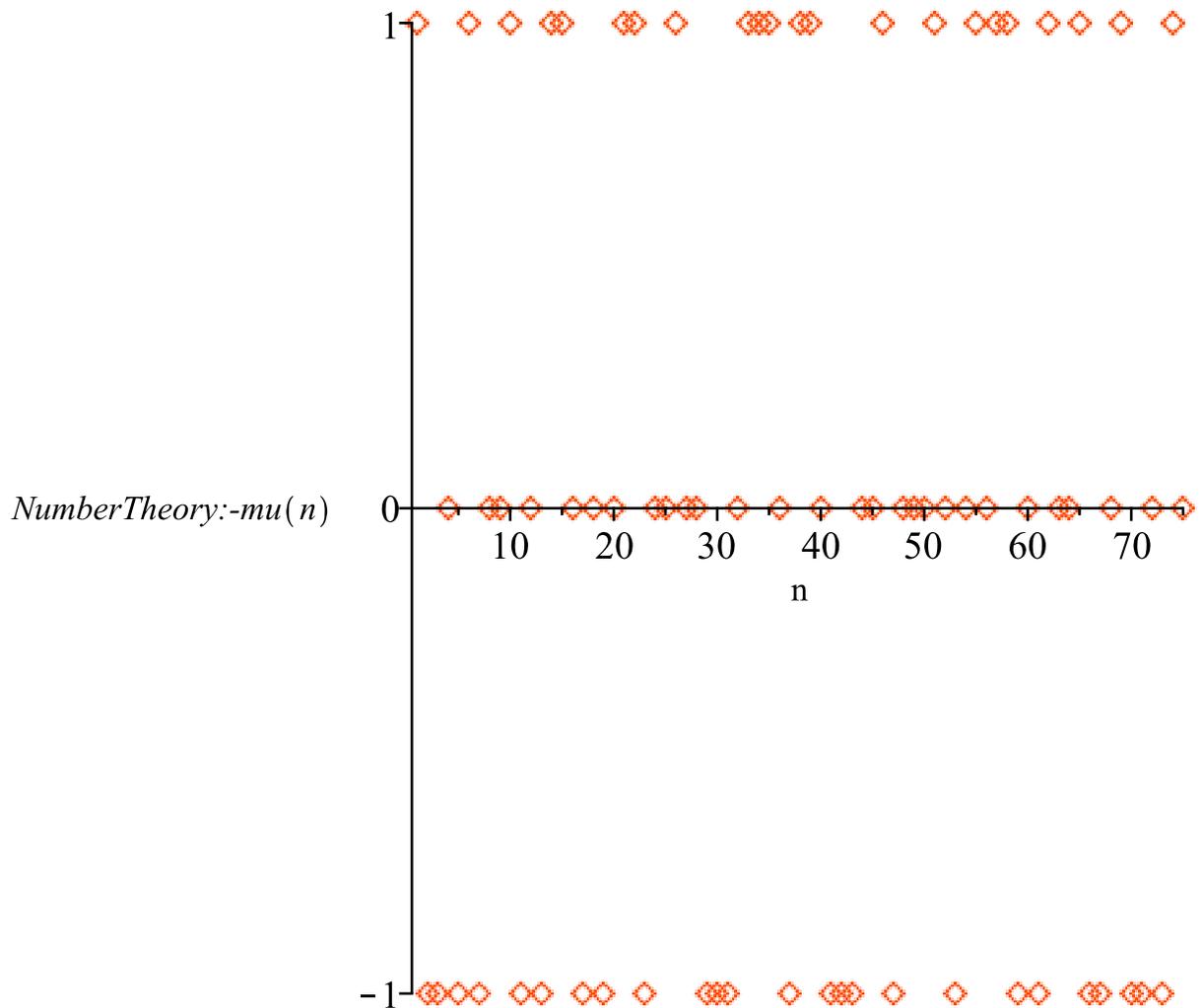
The Möbius function is defined as a multiplicative arithmetic function,  $\mu(r)$ , where:

$$\mu(r) = \begin{cases} 1 & \text{"r is a square-free positive integer with an even number of prime factors"} \\ -1 & \text{"r is a square-free positive integer with an odd number of prime factors"} \\ 0 & \text{"r has a squared prime factor"} \end{cases}$$

For example,  $\mu(2) = -1$  because 2 is a square-free positive integer which has 1 prime factor.  $\mu(4) = 0$  since 4 is not a square-free positive integer.

The first 75 values for the Möbius function are plotted below:

```
plots:-pointplot([seq([n, Moebius(n)], n = 1..75)], labels = ["n",  $\mu(n)$ ], symbolsize = 15, color = "OrangeRed", size = [600, 400], tickmarks = [default, [-1, 0, 1]])
```



## ▼ Examples: Calkin-Wilf Tree

The vertices of the Calkin-Wilf tree are labeled with rational numbers  $\frac{a}{b}$ . The root vertex is defined to be  $\frac{1}{1}$  and for any vertex  $\frac{a}{b}$ , its children are  $\frac{a}{a+b}$  and  $\frac{a+b}{b}$ . Every positive rational number occurs exactly once in the Calkin-Wilf tree.

The first 16 terms of the Calkin-Wilf sequence are:

`seq(CalkinWilfSequence(n), n = 1 ..16)`

$$1, \frac{1}{2}, 2, \frac{1}{3}, \frac{3}{2}, \frac{2}{3}, 3, \frac{1}{4}, \frac{4}{3}, \frac{3}{5}, \frac{5}{2}, \frac{2}{5}, \frac{5}{3}, \frac{3}{4}, 4, \frac{1}{5}$$

The Calkin-Wilf tree can be created using the [GraphTheory](#) package:

`DrawTree := proc(n, {graphstyle := tree})`

