

# Package ‘collatz’

August 19, 2022

**Version** 0.1.0

**License** Apache License (= 2.0) | file LICENSE

**Title** Functions Related to the Collatz/Syracuse/ $3n+1$  Problem

**Description** Provides the basic functionality to interact with the Collatz conjecture.  
The parameterisation uses the same (P,a,b) notation as Conway's generalisations.  
Besides the function and reverse function, there is also functionality to retrieve the hailstone sequence, the ``stopping time"/^ ``total stopping time", or tree-graph.  
The only restriction placed on parameters is that both P and a can't be 0.

**Author** Nathan Levett [aut, cre]

**Maintainer** Nathan Levett <nathan.a.z.levett@gmail.com>

**Date** 2022-08-08

**URL** <https://github.com/Skenvy/Collatz>, <https://github.com/Skenvy/Collatz/tree/main/R>

**BugReports** <https://github.com/Skenvy/Collatz/issues/new/choose>

**Depends** R ( $\geq 3.5.0$ ),  
gmp

**Suggests** roxygen2,  
testthat ( $\geq 3.0.0$ ),  
devtools,  
covr,  
DT,  
pkgdown,  
servr,  
tinytex,  
knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.2.1

## R topics documented:

assertSaneParameterication . . . . .	2
collatz . . . . .	2
collatzFunction . . . . .	3
reverseFunction . . . . .	3

---

assertSaneParameterication

*Sane Parameter Check*

---

### Description

Handles the sanity check for the parameterisation (P,a,b)

### Usage

```
assertSaneParameterication(P, a, b)
```

### Arguments

P	Modulus used to divide n, iff n is equivalent to (0 mod P).
a	Factor by which to multiply n.
b	Value to add to the scaled value of n.

### Details

Required by both the function and reverse function, to assert that they have sane parameters, otherwise will force stop the execution.

---

collatz

*Collatz*

---

### Description

Functions related to the Collatz/Syracuse/3N+1 problem.

### Details

Provides the basic functionality to interact with the Collatz conjecture. The parameterisation uses the same (P,a,b) notation as Conway's generalisations. Besides the function and reverse function, there is also functionality to retrieve the hailstone sequence, the "stopping time"/"total stopping time", or tree-graph. The only restriction placed on parameters is that both P and a can't be 0.

---

collatzFunction	<i>The Collatz function</i>
-----------------	-----------------------------

---

**Description**

Returns the output of a single application of a Collatz-esque function.

**Usage**

```
collatzFunction(n, P = 2, a = 3, b = 1)
```

**Arguments**

n	(numeric bigz bigq) The value on which to perform the Collatz-esque function
P	(numeric bigz bigq): Modulus used to divide n, iff n is equivalent to (0 mod P). Default is 2.
a	(numeric bigz bigq) Factor by which to multiply n. Default is 3.
b	(numeric bigz bigq) Value to add to the scaled value of n. Default is 1.

**Details**

This function will compute and return the result of applying one iteration of a parameterised Collatz-esque function. Although it will operate with integer inputs, for overflow safety, provide a gmp Big Integer ('bigz').

**Value**

a numeric, either in-built or a bigz | bigq from the gmp library. If either n or P are bigz, then the result of n/P will be a bigq although its denominator(~) will return 1.

---

reverseFunction	<i>The "inverse"/"reverse" Collatz function</i>
-----------------	---

---

**Description**

Calculates the values that would return the input under the Collatz function.

**Usage**

```
reverseFunction(n, P = 2, a = 3, b = 1)
```

**Arguments**

n	(numeric bigz bigq) The value on which to perform the reverse Collatz function
P	(numeric bigz bigq) Modulus used to divide n, iff n is equivalent to $(0 \bmod P)$ Default is 2.
a	(numeric bigz bigq) Factor by which to multiply n. Default is 3.
b	(numeric bigz bigq) Value to add to the scaled value of n. Default is 1.

**Details**

Returns the output of a single application of a Collatz-esque reverse function. If only one value is returned, it is the value that would be divided by P. If two values are returned, the first is the value that would be divided by P, and the second value is that which would undergo the multiply and add step, regardless of which is larger.

**Value**

A vector of either numeric or bigz | bigq type