Module Report: Angad's Library of Mathematical Functions

Module Overview

Module Name: Angad's Library of Mathematical Functions

Version: 0

Author: Angad Singh

Date Created: 06.05.2022

Module Description: This module contains a comprehensive collection of number theoretic mathematical functions. These functions cover a wide range of operations including prime testing, divisor counting, summation, factorial computation, and various number transformations.

Key Features

Mathematical Constants

The module defines several mathematical constants that are crucial in various computations:

- Pi
- Tau
- Euler's Number
- Euler-Mascheroni Constant
- Golden Ratio
- Square Roots of 2, 3, and 5
- Riemann Zeta Function Values for 2 and 3

Core Functions

Prime-Related Functions

- is_prime(n, k=10): Determines whether a number is prime using deterministic primality testing.
- nthpri(n): Finds the nth prime number.
- pi(n): Returns the number of primes less than or equal to n.
- distinct_Prime_Factors(Num): Returns the number of distinct prime factors of a number.

- Total_Prime_Factors(Num): Returns the total number of prime factors of a number.

Divisors and Factors

- d(Num): Returns the total number of divisors of a number.
- sigma(Num): Returns the sum of all positive divisors of a number.
- gcd(m, n): Returns the greatest common divisor of two numbers.

Special Number Transformations

- btod(Num): Converts a number from binary to decimal.
- dtob(Num): Converts a number from decimal to binary.
- factorial(Num): Returns the factorial of a number.
- rev(n): Returns the reverse of a number.

Sum of Squares and Cubes

- r_2(n): Returns the number of ways to write a number as the sum of two squares.
- sqr_3(n): Returns the number of ways to write a number as the sum of three squares.
- cbr_2(n): Returns the number of ways to write a number as the sum of two cubes.
- cbr_3(n): Returns the number of ways to write a number as the sum of three cubes.

Special Functions

- SawTooth(x): Returns the value of the sawtooth function used in signal analysis.
- DedekindSum(a, b, c): Computes the Dedekind sum used in analytic number theory.
- Derivative(k, n): Calculates the derivative of a specific function.

Potential Improvements

1. Code Optimization: Some functions, such as `is_prime`, could be optimized further by reducing the complexity and avoiding repetitive calculations.

2. Code Consistency: The module has a mix of different coding styles, particularly in naming conventions. Standardizing function names and variable names could improve readability and

maintainability.

3. Documentation: The comments in the code are helpful, but adding more detailed docstrings and examples for each function would greatly benefit users.

Conclusion

The module is a robust and versatile library for number theoretic functions. With some refinements, it can become an even more powerful tool for mathematical computations.