Ministry of Education, Culture and Research of the Republic of Moldova

Technical University of Moldova

Department of Software and Automation Engineering

**REPORT**

Laboratory work No. 6

Discipline: Algorithms’ Analysis

Topic: Study and empirical analysis of algorithms that determine a N decimal digit of PI

Elaborated: Tudor Sclifos,

student group FAF-211

Verified: Fiștic Cristofor,

university assistant

Chișinău – 2023

Algorithm analysis

**Objective:**

Implement 3 algorithms: bbp, chudnovky and monte-carlo for determining the n-th decimal digit of pi**.**

**Tasks:**

1 Implement at least 2 algorithms that determine the Nth decimal digit of Pi in a programming language. (For ten you need to implement 3 algorithms)

2 Choose metrics for comparing algorithms

3 Perform empirical analysis of the proposed algorithms

4 Make a graphical presentation of the data obtained

5 Make a conclusion on the work done.

**Theoretical Notes:**

The BBP algorithm employs a combination of binary and hexadecimal representations to calculate the individual digits of π. It uses the formula:

π = ∑(k = 0 to ∞) [1/16^k \* (4/(8k + 1) - 2/(8k + 4) - 1/(8k + 5) - 1/(8k + 6))]

The algorithm allows for the computation of π without requiring the previous digits to be known. It calculates each decimal place independently, making it suitable for parallel computation and efficient computation of specific digits.

The Chudnovsky algorithm is based on the following formula:

π = 426880 \* sqrt(10005) / Σ(k = 0 to ∞) [(6k)! \* (545140134k + 13591409) / ((3k)! \* (k!)^3 \* (-640320)^(3k))]

This formula utilizes the concept of infinite series to approximate π. The algorithm involves summing an infinite number of terms, with each term contributing to the final value of π.

The key feature of the Chudnovsky algorithm is its rapid convergence, meaning it achieves a high precision for π with relatively few iterations. It benefits from the factorials and power of -640320 being precomputed to reduce computational complexity.

The Monte Carlo method is a statistical technique that can be used to estimate the value of π (pi) based on random sampling. It takes advantage of the relationship between the area of a circle and the area of a square to approximate π.

The underlying principle of the Monte Carlo method is that as the number of randomly generated points increases, the ratio of points falling inside the circle to the total number of points approximates the ratio of the areas of the circle to the square. Since the area of the circle is πr^2 and the area of the square is (2r)^2 = 4r^2, the ratio provides an estimation of π/4. Multiplying the ratio by 4 yields an estimate of π.

**Introduction:**

In this laboratory work I have to implement algorithms and apply them to get the n-th term. Also I need to analyse the outputs.

**Comparison metric:**

The comparison metric for this laboratory work will be considered the time of execution of each algorithm (T(n)).

**Input format:**

The input for this laboratory work are the list of n:

n\_list = [10, 50, 100, 200, 300, 500, 800, 1000, 1200, 1500]

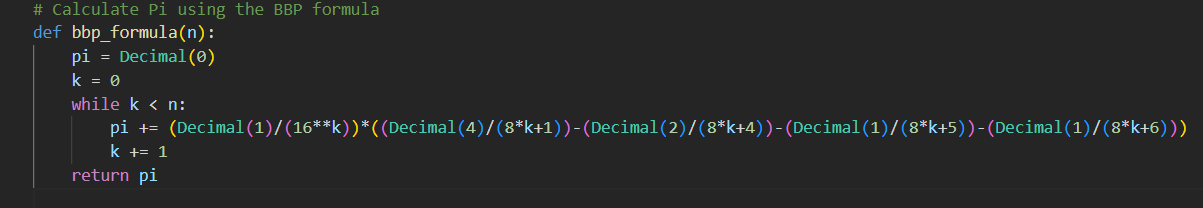
**IMPLEMENTATION**

**BBP**

The BBP algorithm employs a combination of binary and hexadecimal representations to calculate the individual digits of π. It uses the formula:

π = ∑(k = 0 to ∞) [1/16^k \* (4/(8k + 1) - 2/(8k + 4) - 1/(8k + 5) - 1/(8k + 6))]

.



**Figure 1. Implementation BBP**

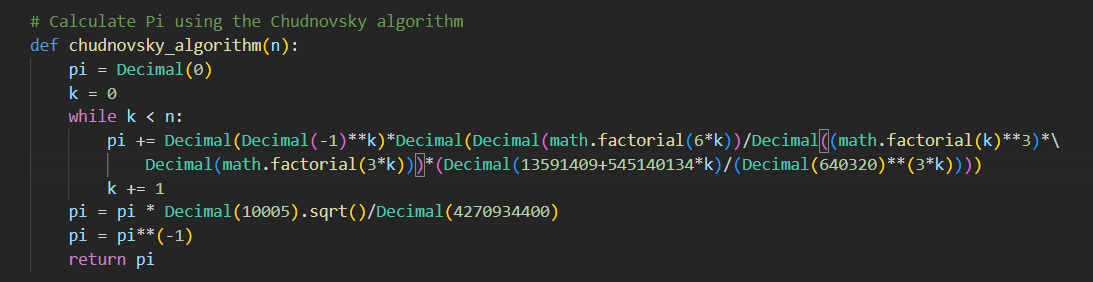
**Chudnovsky**

**Implementation:**

The Chudnovsky algorithm is based on the following formula:

π = 426880 \* sqrt(10005) / Σ(k = 0 to ∞) [(6k)! \* (545140134k + 13591409) / ((3k)! \* (k!)^3 \* (-640320)^(3k))]

This formula utilizes the concept of infinite series to approximate π. The algorithm involves summing an infinite number of terms, with each term contributing to the final value of π.

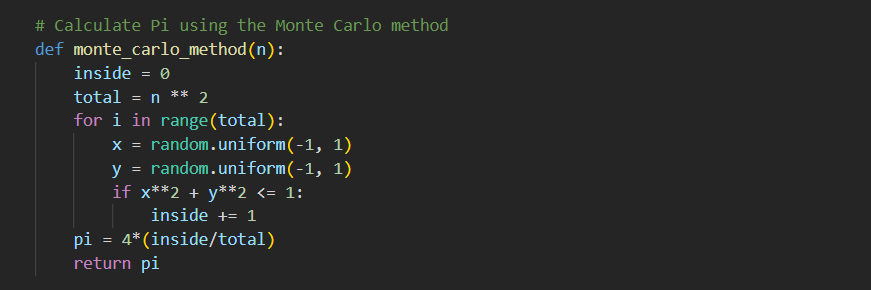


**Figure 2. Implementation Dijkstra**

**MonteCarlo**

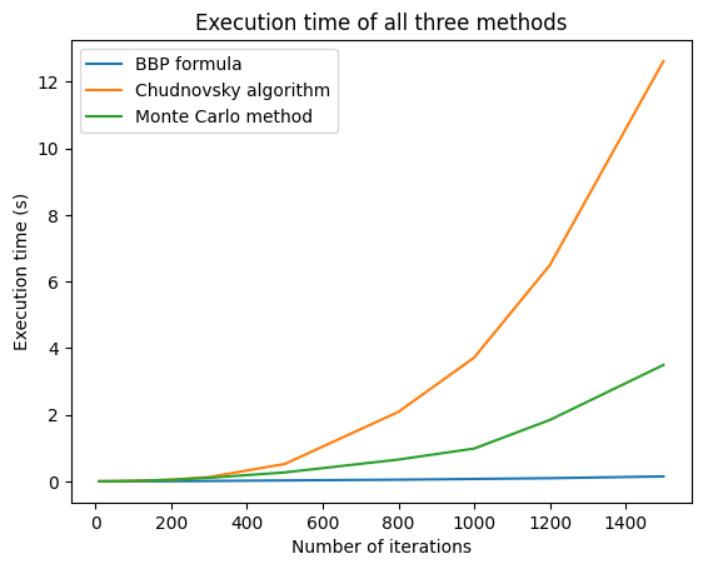
**Implementation:**

The underlying principle of the Monte Carlo method is that as the number of randomly generated points increases, the ratio of points falling inside the circle to the total number of points approximates the ratio of the areas of the circle to the square. Since the area of the circle is πr^2 and the area of the square is (2r)^2 = 4r^2, the ratio provides an estimation of π/4. Multiplying the ratio by 4 yields an estimate of π.



**Figure 3. Implementation MonteCarlo**

**Results for the sparse tree:**



**Figure 4. Plot with results for the n-th term.**

**Conclusion:**

In conclusion, the last three algorithms discussed—BBP algorithm, Chudnovsky algorithm, and Monte Carlo method—are all powerful tools for calculating or approximating the value of π (pi) with different approaches:

The BBP algorithm uses a mathematical formula involving binary and hexadecimal representations to calculate individual digits of π. It allows for the efficient computation of specific decimal or hexadecimal digits without requiring previous digits.

The Chudnovsky algorithm is a fast and highly precise algorithm that utilizes an infinite series formula to approximate π. It converges rapidly and has been instrumental in setting records for calculating π to trillions of decimal places.

The Monte Carlo method employs random sampling within a unit square and inscribed circle to estimate π. By generating a large number of random points and calculating the ratio of points inside the circle to the total, π can be approximated. The method offers flexibility and is widely applicable, but convergence speed can be slower compared to other algorithms.

Each algorithm has its own strengths and applications. The BBP algorithm is useful for calculating specific digits, the Chudnovsky algorithm excels in high precision calculations, and the Monte Carlo method provides a probabilistic estimation of π.

**Link to GitHub:** <https://github.com/SexomQ/AlgorithmsAnalysis-labs>