Ministerul Educaţiei și Cercetării al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică

Laboratory work 3:
Empirical analysis of algorithms for obtaining Eratosthenes Sieve

Elaborated:
st. gr. FAF-211 Echim Mihail

Verified:
asist. univ. Fiștic Cristofor

Chişinău - 2023

ALGORITHM ANALYSIS

Objective
Study and analyze different algorithms for obtaining Eratosthenes Sieve

Tasks:

1 Implement the algorithms listed below in a programming language
2 Establish the properties of the input data against which the analysis is performed
3 Choose metrics for comparing algorithms
4 Perform empirical analysis of the proposed algorithms
5 Make a graphical presentation of the data obtained
6 Make a conclusion on the work done.

Theoretical Notes:

An alternative to the mathematical analysis of complexity is empirical analysis.

This may be useful for obtaining preliminary information on the complexity class of an algorithm; comparing the efficiency of two (or more) algorithms for solving the same problems; comparing the efficiency of several implementations of the same algorithm; obtaining information on the efficiency of implementing an algorithm on a particular computer.

In the empirical analysis of an algorithm, the following steps are usually followed:

1. The purpose of the analysis is established.
2. Choose the efficiency metric to be used (number of executions of an operation (s) or time execution of all or part of the algorithm.
3. The properties of the input data in relation to which the analysis is performed are established (data size or specific properties).
4. The algorithm is implemented in a programming language.
5. Generating multiple sets of input data.
6. Run the program for each input data set.
7. The obtained data are analyzed.

The choice of the efficiency measure depends on the purpose of the analysis. If, for example, the aim is to obtain information on the complexity class or even checking the accuracy of a theoretical estimate then it is appropriate to use the number of operations performed. But if the goal is to assess the behavior of the implementation of an algorithm then execution time is appropriate.

After the execution of the program with the test data, the results are recorded and, for the purpose of the analysis, either synthetic quantities (mean, standard deviation, etc.) are calculated or a graph with appropriate pairs of points (i.e. problem size, efficiency measure) is plotted.

Introduction:

The Sieve of Eratosthenes is an ancient algorithm used to find all prime numbers up to a given limit. The algorithm is named after the ancient Greek mathematician Eratosthenes who first described it.

The algorithm works by first creating a list of all integers from 2 up to the given limit. Then, starting with the first prime number (which is 2), the algorithm marks all multiples of 2 as composite (not prime). Next, it moves to the next unmarked number, which is the next prime number (which is 3), and marks all multiples of 3 as composite. The algorithm continues this process, moving to the next unmarked number and marking all of its multiples as composite until it has processed all numbers up to the given limit.

At the end of the process, all unmarked numbers are prime. The algorithm is very efficient and can find all primes up to very large numbers quickly. However, it requires a significant amount of memory to store the list of integers up to the given limit.

Comparison Metric:

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

Input Format:

As input, each algorithm will receive arrays randomly filled with integers bigger than 0 and smaller than 101. The algorithms will be of sizes: 10, 100, 1000, 10000, and 100000. I tried using 10^6 as well, but most algorithms gave either maximum recursion error or memory error, so I decided to stop at 10^5.