**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук

Кафедра      информационных технологий

«УТВЕРЖДАЮ»

Заведующий кафедрой

информационных технологий

д.ф.-м.н., проф.

\_\_\_\_\_\_\_\_\_\_\_\_Ю.Н. Орлов

«  »                     2021г.

**ОТЧЕТ**

**по лабораторной работе**

ТЕМА «**Пи - алгоритм Spigot**»

по дисциплине «Компьютерный практикум по ИТ»

Выполнил:

Студент группы НПИбд-02-21

Студенческий билет № 123456789

           Пономарев Илья Юрьевич

(Подпись)

«  »                     2021г.

Руководитель:

к.ф.-м.н., доцент кафедры информационных технологий

Виноградов Андрей Николаевич

 (Подпись)

 «  »                     2021г.

Оценка:                                                         

Москва 2021

Цель работы:

Реализация вычисления n-го числа Pi с помощью алгоритма Spigot.

Ход работы

В ходе работы программы был использован язык Javascript. Чтобы разрабатывать консольное приложение, использовалась платформа nodeJS

Алгоритм Spigot (алгоритм «краника») - Идея алгоритма заключается в том, что за одну итерацию мы получаем ровно одну цифру числа Пи и потом её не используем.

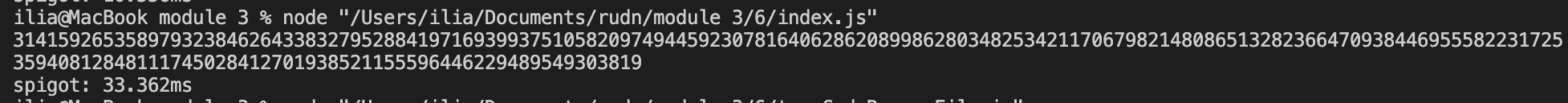
Алгоритм выполнения

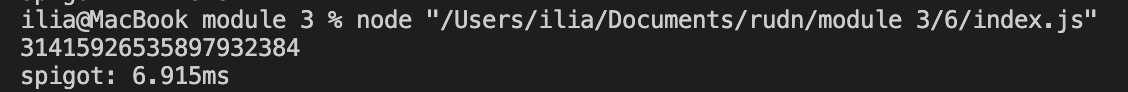
Изображение выглядит как стол

Автоматически созданное описание

Как мы можем видеть, время выполнения и количество элементарных действий алгоритма растет экспоненциально

Скриншоты (примеры) выполнения программы





let counter = 0 //счетчик элементарных действий

const mod = (m, n) => {

counter += 3

return ((m % n) + n) % n

}

const calculate = (NDIGITS) => {

let LEN = (NDIGITS / 4 + 1) \* 14, // формула длины массива

out = "",

arr = [],

i = 0, //индекс массива

// алгоритм работает быстрее, если перезаписать длину массива и не высчитывать каждый раз переменную len

len = LEN,

d = 0,

e = 0,

f = 10000, //знаменатель

g = 0,

digit = 0;

for (; arr.length !== len; arr.push(0)); //заполняем массив 0

for (;

(i = len -= 14) > 0;) {

for (; --i > 0;) { //перебираем массив с конца

counter += 1

d \*= i; //сумма

// подсчет суммы

if (digit === 0) {

counter += 1

d += 2000 \* f // умножение суммы на знаменатель

} else {

counter += 1

d += arr[i] \* f; //умножаем на знаменатель сумму предыдущего числа и

}

counter += 2

g = i \* 2 - 1; // знаменатель

arr[i] = mod(d, g); //сохранение остатка от деления суммы на знаменатель

counter += 1

d = Math.floor(d / g); //округленный результат деления суммы на знаменатель

}

counter += 2

digit = Math.floor(e + d / f);

out += digit; //вывод

d = e = mod(d, f); //сохраняем в переменную "e", потому переменная d изменяется

}

return out;

}

console.time('spigot')

console.log(calculate(10000));

console.log(counter);

console.timeEnd('spigot')

let counter = 0 //счетчик элементарных действий

const mod = (m, n) => {

counter += 3

return ((m % n) + n) % n

}

const calculate = (NDIGITS) => {

let LEN = (NDIGITS / 4 + 1) \* 14, // формула длины массива

out = "",

arr = [],

i = 0, //индекс массива

// алгоритм работает быстрее, если перезаписать длину массива и не высчитывать каждый раз переменную len

len = LEN,

d = 0,

e = 0,

f = 10000, //знаменатель

g = 0,

digit = 0;

for (; arr.length !== len; arr.push(0)); //заполняем массив 0

for (;

(i = len -= 14) > 0;) {

for (; --i > 0;) { //перебираем массив с конца

counter += 1

d \*= i; //сумма

// подсчет суммы

if (digit === 0) {

counter += 1

d += 2000 \* f // умножение суммы на знаменатель

} else {

counter += 1

d += arr[i] \* f; //умножаем на знаменатель сумму предыдущего числа и

}

counter += 2

g = i \* 2 - 1; // знаменатель

arr[i] = mod(d, g); //сохранение остатка от деления суммы на знаменатель

counter += 1

d = Math.floor(d / g); //округленный результат деления суммы на знаменатель

}

counter += 2

digit = Math.floor(e + d / f);

out += digit; //вывод

d = e = mod(d, f); //сохраняем в переменную "e", потому переменная d изменяется

}

return out;

}

console.time('spigot')

console.log(calculate(10000));

console.log(counter);

console.timeEnd('spigot')