|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №1**

**«Измерение времени выполнения алгоритма»**

**ДИСЦИПЛИНА: «Логика и теория алгоритмов»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-42Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Амеличев Г.Э. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2022

**Цель:** выработать навыки оценки вычислительной сложности алгоритма.

**Задачи:** в лабораторной работе необходимо реализовать заданный алгоритм и исследовать его вычислительную сложность.

**Вариант №2**

Поиск в массиве размера N смежного подмассива размера M с наибольшим средним значением элементов, посредством перебора всех подмассивов.

**Псевдокод алгоритма:**

бинарный\_код = 0

среднее\_арифметическое =

пока бинарный\_код + 1 ≠ 2N

нц

бинарный\_код = бинарный\_код + 1

количество\_единиц = 0

текущая\_позиция = 0

сумма = 0

пока b не переберет все биты бинарного\_кода с конца

нц

текущая\_позиция = текущая\_позиция + 1

если b = 1, то

ну

количество\_единиц = количество\_единиц + 1

сумма = сумма + бит бинарного\_кода на текущей\_позиции с конца

ку

кц

если количество\_единиц = М, то

если сумма / М > среднее\_арифметическое, то

ну

среднее\_арифметическое = сумма / М

искомый\_код = бинарный\_код

ку

кц

**Листинг:**

import random

import math

from datetime import datetime

import csv

def Algorithm(array, N, M): # N \* 2^N

code = 0 # c1

average = -math.inf # c2

biggestCode = None # c3

while code + 1 != 2\*\*N: # 2^N \* c4

code += 1 # (2^N - 1) \* c5

size = 0 # (2^N - 1) \* c6

position = 0 # (2^N - 1) \* c7

summ = 0 # (2^N - 1) \* c8

for b in bin(code)[2:][::-1]: # (2^N - 1) \* c9

position += 1 # (2^N \* (N-1) + 1) \* c10

if b == '1': # (2^N \* (N-1) + 1) \* c11

size += 1 # 2^(N-1) \* N \* c12

summ += array[-position] # 2^(N-1) \* N \* c13

if size == M: # (2^N - 1) \* c14

if summ / M > average: # N! / (M!(N-M)!) \* c15

average = summ / M # c16 || N! / (M!(N-M)!) \* c16

biggestCode = code # c17 || N! / (M!(N-M)!) \* c17

return biggestCode

N\_min = 10

N\_max = 25

N\_delta = 1

with open('data.csv', 'w', encoding='utf-8', newline='') as f:

writer = csv.writer(f)

for N in range(N\_min, N\_max + 1, N\_delta):

M = N // 2

array = [random.randint(-10, 10) for i in range(N)]

start = datetime.now()

biggestCode = Algorithm(array, N, M)

end = datetime.now()

result = []

for i in range(len(bin(biggestCode)) - 2):

if bin(biggestCode)[-i - 1] == '1':

result.append(array[-i - 1])

print(N, end - start, array, result[::-1])

writer.writerow([N, end - start])

**Результат:**

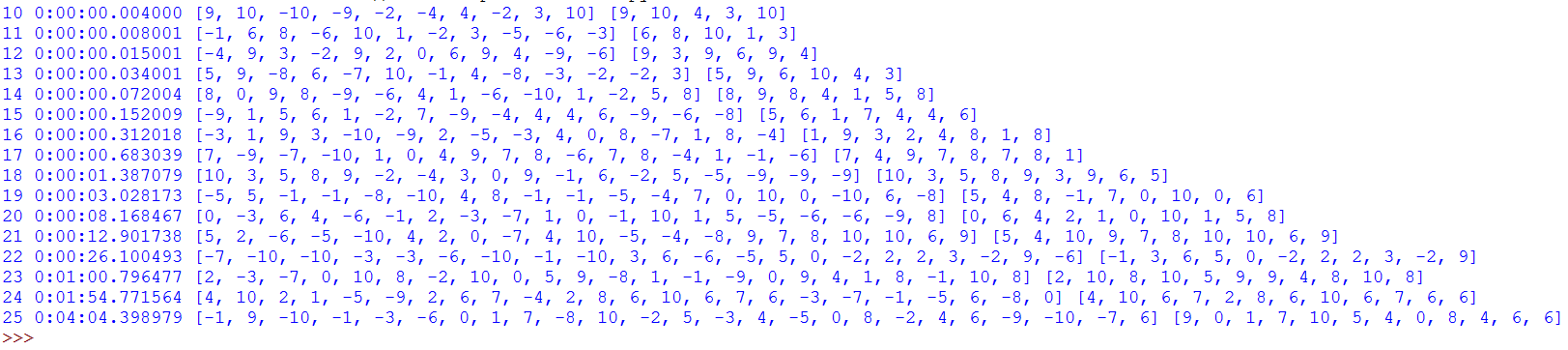


Рис. 1. Результат

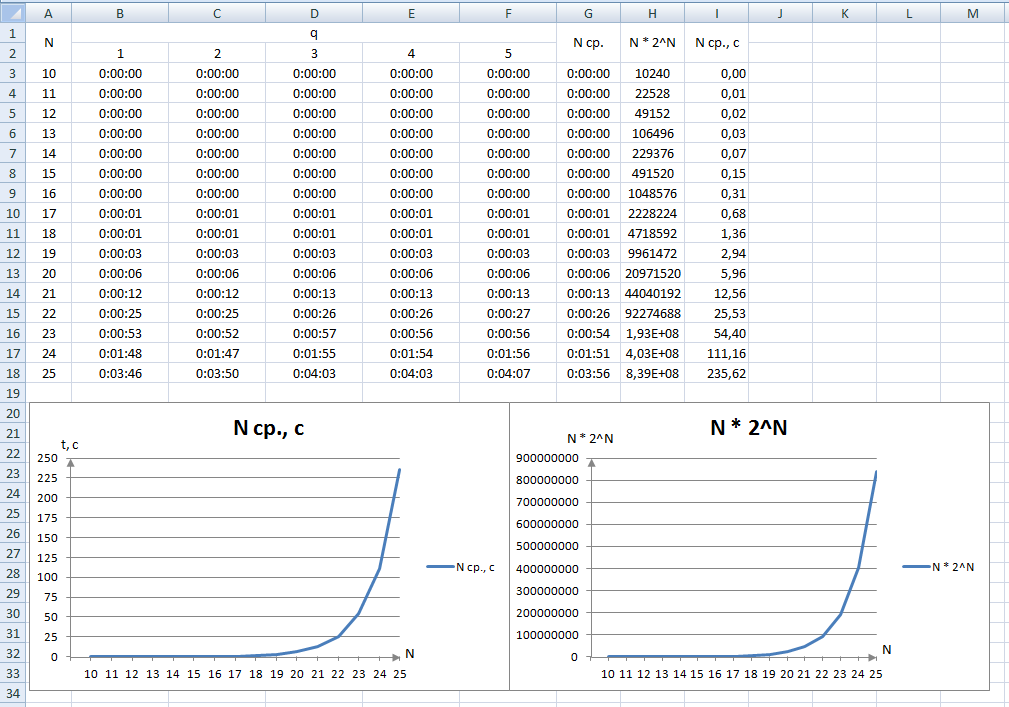


Рис. 2. Графики

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки расчета вычислительной сложности алгоритма, составления псевдокода.