Министерство образования и науки РФ

Новосибирский государственный технический университет

Кафедра систем сбора и обработки данных

**Лабораторная работа №1**

Вариант 13.

Определение суммы простых чисел.

Факультет: АВТФ Преподаватель: Квашнина Е. А.

Группа: АТ-33

Студент: Мельничук В.

Новосибирск, 2025

1. Формулировка задания

1. Реализовать функцию, которая на вход получает натуральное четное число и возвращает два простых числа, которые в сумме составляют входное число (с подсчетом количества итераций).

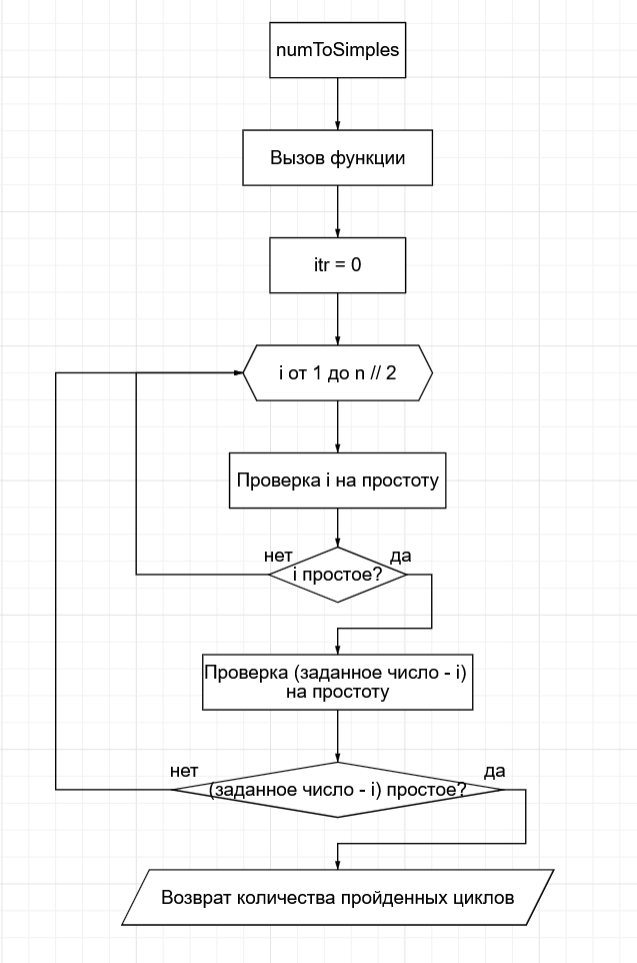
2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующих значений : 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

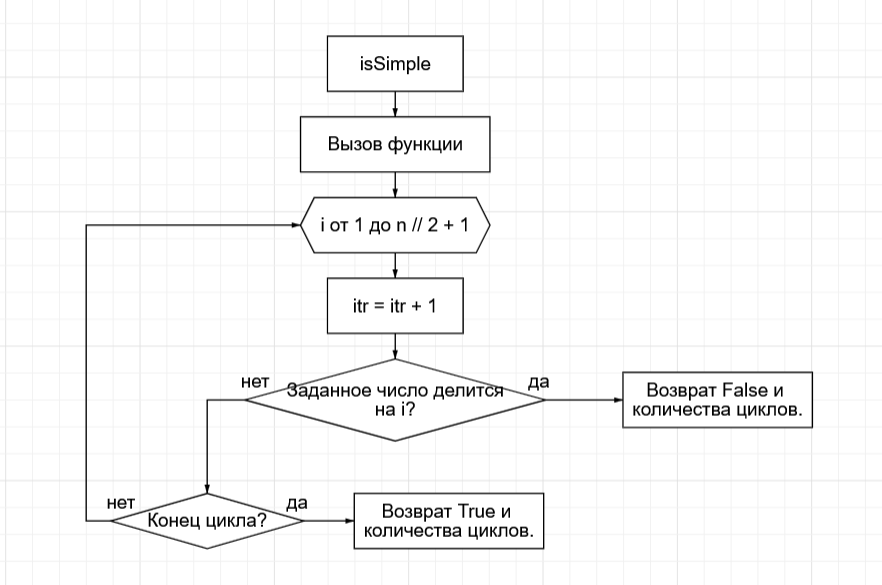
Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |
| --- | --- | --- | --- |
| Num | Тэ1 | Т1 | Тэ1/T1 |
| 8 | … | … | … |
| 16 | … | … | … |
| … | … | … | … |

По результатам сделайте вывод какой асимптотической сложности получился реализованный алгоритм.

1. Блок-схемы основных алгоритмов





1. Листинг программы

# Вариант 13.

# 1. Реализовать функцию, которая на вход получает натуральное четное число и возвращает два простых числа,

# которые в сумме составляют входное число (с подсчетом количества итераций).

# 2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующих значений:

# 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

def numToSimples(inp):

itr = 0

for i in range(1, inp // 2):

simp, itr = isSimple(i, itr)

if simp:

simp, itr = isSimple(inp - i, itr)

if simp:

return itr

def isSimple(num, iterations):

for i in range(2, num // 2 + 1):

iterations += 1

if num % i == 0:

return False, iterations

return True, iterations

def main():

tests = [8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096]

print("num | Te1 | T1 | Te1/T1 ")

print('-' \* 35)

for i in tests:

print(f'{i:<8}|', end='')

test = numToSimples(i)

print(f' {test:<7}|', end='')

print(f' {(i)//2:<7}|', end='')

print(f' {test/((i)//2)}')

if \_\_name\_\_ == '\_\_main\_\_':

main()

1. Вывод

Выполнив подсчеты, вычисленной сложностью алгоритма оказалась O(n3/8), однако в среднем программа завершалась за O(n/2), потому что одним из простых чисел в большинстве случаев оказывалась 1, из-за чего функция определения простоты вызывалась всего лишь один раз (для единицы функция определения простоты завершалась за одну итерацию).