Отчёт по лабораторной работе №6  
Разложение чисел на множители

Студент: Леонова Алина Дмитриевна, 1032212306

Группа: НФИмд-01-21

Преподаватель: Кулябов Дмитрий Сергеевич,

д-р.ф.-м.н., проф.

Москва 2021

Содержание

# 1 Цель работы

Целью данной работы является ознакомление с методом разложения чисел на множители и реализация этого метода на выбранном языке программирования.

# 2 Задание

1. Реализовать рассмотренный алгоритм программно.
2. Разложить на множители данное преподавателем число.

# 3 Теоретическое введение

Процесс разложения составного числа на множителе является факторизацией. В отличие от задачи распознавания простоты числа, факторизация предположительно является вычислительно сложной задачей [1].

## 3.1 -метод Полларда

-алгоритм предложен Джоном Поллардом в 1975 году для факторизации целых чисел. Данный алгоритм основывается на алгоритме Флойда поиска длины цикла в последовательности и некоторых следствиях из парадокса дней рождения. Алгоритм наиболее эффективен при факторизации составных чисел с достаточно малыми множителями в разложении. Сложность алгоритма оценивается как .

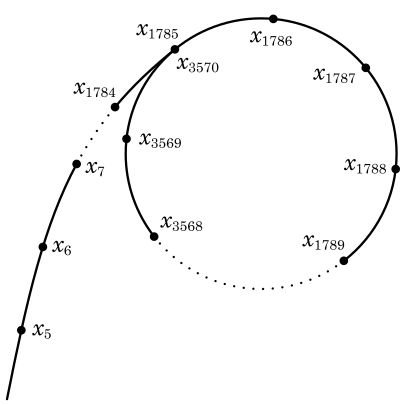


Figure 1: Числовая последовательность зацикливается, начиная с некоторого n. Цикл может быть представлен в виде греческой буквы ро

-алгоритм Полларда строит числовую последовательность, элементы которой образуют цикл, начиная с некоторого номера n, что может быть проиллюстрировано, расположением чисел в виде греческой буквы (см. рис. 1), что послужило названием семейству алгоритмов [2].

## 3.2 Пример из задания

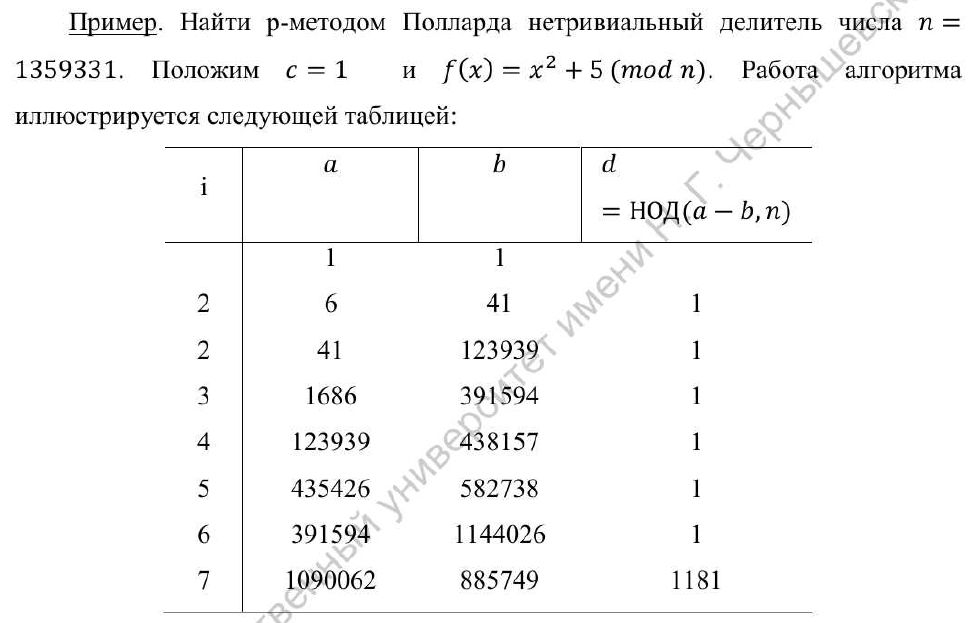


Figure 2: Пример разложения числа ро-методом Полларда

Пример работы алгоритма, на котором требуется проверить свою реализацию (см. рис. 2).

# 4 Выполнение лабораторной работы

## 4.1 Промежуточные функции

Функция для нахождения наибольшего общего делителя a и b - Алгоритм Евклида. Взят из лабораторной работы №4.

# Алгоритм Евклида  
def nod(a, b):  
 if a == 0 or b == 0:  
 return max(a, b)  
 if a == 1 or b == 1:  
 return 1  
 if a < b:  
 a, b = b, a  
 d = nod(a % b, b)  
 return d

Функция eval\_ для нахождения результата переданной как строки функции f с переданными аргументами x и n:

# Функция  
def eval\_(f, x, n):  
 return eval(f)

## 4.2 -метод Полларда

Функция, реализующая -метод Полларда, следуя алгоритму из задания. Возвращение ко 2 шагу реализовано с помощью использования бесконечного цикла.

def Pollard(n, c, f):  
 print('n = ', n, '; c = ', c,'; f = ', f)  
 a, b = c, c  
  
 while True:  
 a = eval\_(f, a, n) % n  
 b = eval\_(f, eval\_(f, b, n), n) % n  
 print('a = ',a,' b = ',b)  
   
 if a - b < 0:  
 d = 1  
 else:  
 d = nod(a-b, n)  
   
 if 1 < d and d < n:  
 return d  
 if d == n:  
 return print('Делитель не найден')  
 if d == 1:  
 print('1')

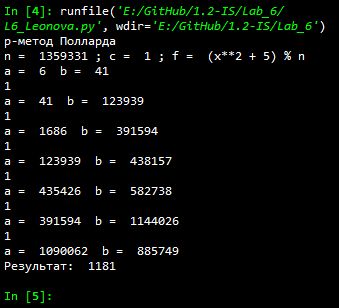


Figure 3: Результат выполнения L6\_Leonova.py

Результат выполнения программы, проверка реализации -метода Полларда, разложение на множители данного в задании числа (см. рис. 3).

# 5 Выводы

Цель лабораторной работы была достигнута, метод разложения чисел на множители - -Метод Полларда - был реализован на языке программирования Python.

# Список литературы

1. Факторизация целых чисел [Электронный ресурс]. Википедия, 2021. URL: <https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D1%86%D0%B5%D0%BB%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB>.

2. Ро-алгоритм Полларда [Электронный ресурс]. Википедия, 2021. URL: <https://ru.wikipedia.org/wiki/%D0%A0%D0%BE-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9F%D0%BE%D0%BB%D0%BB%D0%B0%D1%80%D0%B4%D0%B0>.