

МИНОБРНАУКИ РОССИИ

Федеральное государственное образовательное учреждение

высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Кибернетики

Кафедра Информационной безопасности

Практическое задание

по дисциплине

«Криптографические протоколы»

Выполнил

студент группы ККСО-01-16

Козеев Б.Н.

Научный руководитель

Никитин А.П.

Москва

2021

# Общее описании алгоритма

Линейный конгруэнтный метод был придуман Д.Г. Лемером в 1949 году. Основа метода данная рекуррентная формула –



где m >= 2 модуль, a – множитель 0 <= a < m, c – приращение 0 <= c < m, X – начальное значение 0 <= X < m.

Чтобы длина периода в последовательности была равна модулю, необходимо, чтобы выполнялись следующие свойства:

1. Числа c и m взаимно просты;
2. b = a -1 кратно p, для каждого простого p, являющегося делителем m;
3. b кратно 4, если m кратно 4.

При выполнении этих свойств последовательность гарантированно будет длинной m.

# Описание криптографических свойств алгоритма и принципов его построения

Хоть данный метод и порождает статистически хорошую псевдослучайную последовательность чисел, но он не является криптографически стойким. Генераторы на основе линейного конгруэнтного метода являются предсказуемыми, поэтому их нельзя использовать в криптографии. Впервые генераторы на основе линейного конгруэнтного метода были взломаны Джимом Ридсом, а затем Джоан Бояр. Другие исследователи расширили идеи Бояр, разработав способы вскрытия любого полиномиального генератора. Таким образом, была доказана бесполезность генераторов на основе конгруэнтных методов для криптографии. Однако генераторы на основе линейного конгруэнтного метода сохраняют свою полезность для некриптографических приложений, например, для моделирования. Они эффективны и в большинстве используемых эмпирических тестов демонстрируют хорошие статистические характеристики.

Хоть криптографическая полезность этого метода небольшая, но его часто встраивали в rand() функции в языках C/C++.

# Количество раундов (шагов, циклов)

Раундов нет

# Известные оценки качества выходной последовательности по открытым источникам

Данная последовательность легко ломается, об этом сказано выше

# Обоснование выбора данного алгоритма

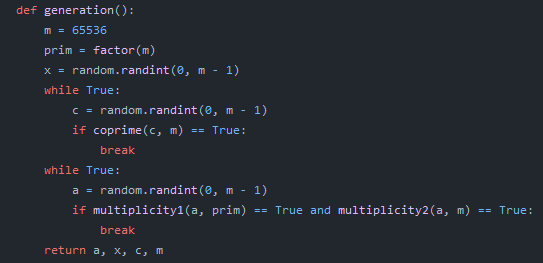
Он использовался в реализациях rand().

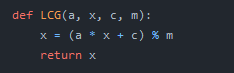
# Тестирование и описании характеристик реализации

# Общее описание программной реализации алгоритма

Программная реализация выглядит так.  
Здесь при запуске указывается сколько чисел ему надо будет сгенерировать.

Так же в программе учтены вышеуказанные свойства для достижения наибольшего периода.

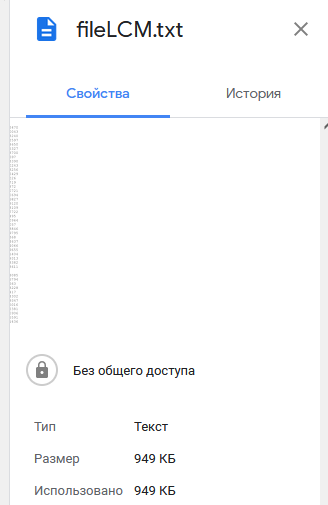


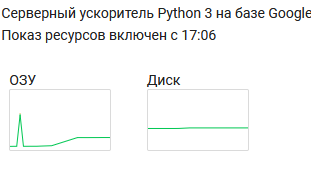


# Скорость выполнения данного алгоритма

Файл 1 мб –

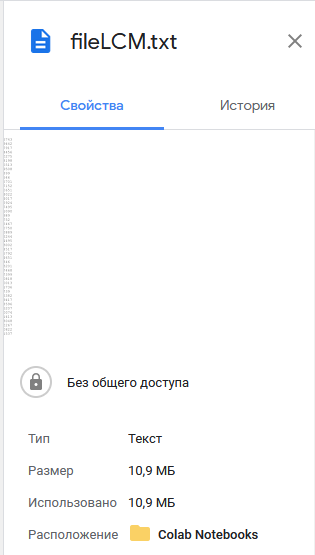


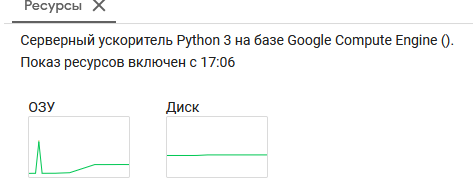




Файл 10 мб –

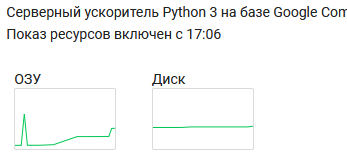


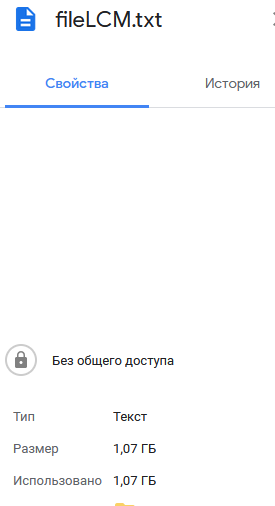




Файл 1гб –







# Описание тестового стенда

Тестовый стенд такой же, как и в прошлых отчетах

# Фиксация нагрузки на ОЗУ и ЦП во время выполнения тестового задания

Представлена ниже