МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт Компьютерных Наук и Кибербезопасности

Высшая школа технологий искуственного интелекта

Направление 02.03.01 Математика и Компьютерные Науки

Лабораторная работа №1

По дисциплине:
«Теория алгоритмов»

Тема Работы:

«Реализация клеточного автомата»

Обучающийся: ______________ Черепанов Михаил Дмитриевич

Руководитель: ______________ Востров Алексей Владимирович

«____» ____________ 20__ г.

Санкт-Петербург 2023

Содержание

Введение

В данной лабораторной работе требовалось:

Реализовать двумерный клеточный автомат с окрестностью фон Неймана в соответствии с заданным номером. Граничный условия предложить самостоятельно. Учесть возможность ввода различных начальных условий - как вручную, так и случайным образом. Также пользователь определяет ширину поля.

Мой номер: 6509750, или в двоичной системе:

11000110101010010110110.

Работа выполнена на языке программирования : Python;

Среда разработки: Pycharm;

1 Описание предметной области

Клеточный автомат (КА) представляет собой совокупность клеток, организованных в периодическую решетку с определенными правилами перехода. Эти правила определяют состояние каждой клетки на следующей итерации, учитывая состояния её ближайших соседей в текущий момент времени. Обычно анализируются автоматы, где состояние зависит от самой клетки и её непосредственных соседей.

Структура клеточного автомата включает в себя набор объектов (ячеек), образующих регулярную решетку. Состояние каждой ячейки в момент времени n определяется переменной, которая может быть целым, действительным или комплексным числом либо набором чисел. Изменения состояний клеток происходят синхронно с дискретными временными интервалами, согласно локальным вероятностным правилам, зависящим от состояний соседних ячеек. Эти правила остаются постоянными со временем.

Клеточный автомат представляет собой дискретную динамическую систему, характеризующуюся полностью в терминах локальных зависимостей. Термин "дискретное пространство"относится к пространству, состоящему из дискретного множества элементов. Решетка клеточного автомата является экземпляром этого пространства, а каждая клетка обладает определенным значением из заданного множества. Каждая клетка может содержать или иметь определенное значение, находиться в определенном состоянии, которое кодируется этим значением.

Совокупность состояний всех клеток решётки называется состоянием решётки. Изменение состояния решётки происходит в соответствии с определенным законом, который называется правилами клеточного автомата. Каждое изменение состояния решётки называется итерацией. В данной модели время дискретно, и каждая итерация соответствует определенному моменту времени.

Правила определяют, какое значение должно находиться в клетке на следующей итерации, в зависимости от значений некоторых других клеток в текущий момент времени, а также, возможно, от значения самой клетки в текущий момент. Если новое состояние клетки зависит от её текущего состояния, то клеточный автомат называется автоматом с клетками с памятью; в противном случае - автоматом с клетками без памяти.

Множество клеток, влияющих на значение данной клетки, за исключением самой клетки, называется окрестностью клетки. Окрестность клетки удобно определять с использованием метрики на решётке, поэтому мы будем рассматривать решётку как дискретное метрическое пространство для удобства. Одно из ключевых отличий клеточной системы от других вычислительных систем заключается в том, что обе её части - архитектурная и данные - состоят из принципиально идентичных элементов. Таким образом, вычислительная система может оперировать своей материальной частью, модифицировать её, расширять и строить аналогичные ей. Несмотря на то, что Джон фон Нейман предложил системы такого класса, такая параллельная архитектура получила название "не-фон-неймановской поскольку он разработал последовательную архитектуру раньше.

1.1 Классификация клеточных автоматов по типу поведения:

Стивен Вольфрам в книге «A New Kind of Science» предложил 4 класса, на которые все клеточные автоматы могут быть разделены в зависимости от их типа поведения.

На рис. 1 можно увидеть схему разделения всех клеточных автоматов, предложенную Стивином Вольфрамом.

Рис. 1: Классы, предложенные Стивеном Вольфрамом

2 Математическое описание

Двумерный клеточный автомат можно определить как множество конечных автоматов на плоскости, помеченных целочисленными координатами (i, j), каждый из которых может находиться в одном из состояний $\sigma_{i,j}$:

$\sigma_{i,j} \in \Sigma \equiv \{0, 1, 2, \ldots, k-1, k\}$

Изменение состояний автоматов происходит согласно правилу перехода:

$$\sigma_{i,j}(t+1) = \phi(\sigma_{k,l}(t) | \sigma_{k,l}(t) \in \mathcal{N}),$$

,

где $\mathcal{N}$ - некоторая окрестность точки (i,j).

В данной работе используется окрестность Фон Неймана, которая определяется как:

$$\mathcal{N}_N^1(i,j) = \{\sigma_{k,l}|\ |i-k|+|j-l| \leq 1\},$$

2.1 Граничные условия

Граничные условия в двумерном клеточном автомате определяют, как взаимодействуют клетки на границе пространства. Существует несколько типов граничных условий, которые часто используются в клеточных автоматах. Вот некоторые из них:

- **Замкнутые граничные условия:**

  При таких условиях автомат рассматривается как замкнутый на себя. Это означает, что клетки на одной границе взаимодействуют с клетками на противоположной границе. Это создает эффект тора, где автомат представляет собой бесконечную поверхность.

- **Отражающие граничные условия:**

  Здесь клетки на границе взаимодействуют с клетками внутри автомата, как если бы граница была зеркальным отражением. Это создает эффект замкнутого пространства.

- **Фиксированные граничные условия:**

  Клетки на границе считаются фиксированными и не изменяются в процессе эволюции автомата. Это означает, что взаимодействие происходит только с клетками внутри автомата.

- **Нейтральные граничные условия**

  Клетки за пределами границы считаются нейтральными или "пустыми и они не взаимодействуют с клетками внутри автомата.

2.2 Правило клеточного автомата

Правило клеточного автомата это набор условий, при которых происходит смена значения ячейки клеточного автомата. Количество всех возможных правил перехода зависит от количества состояний ячейки |G| , количества соседних ячеек в соответствующей окрестности |D| и равно N=$|G|^{|G|^{|D|}}$.

От правила, заданого для конечного автомата зависит то, как изменяются его клетки с течением времени.

Например, в наиболее известном клеточном автомате «Игра жизнь » правило задано следующим образом:

- Живая клетка с меньше чем двумя живыми соседями умирает;

- Живая клетка с двумя или тремя живыми соседями продолжает жить;
- Живая клетка с более чем тремя живыми соседями умирает;
- Мёртвая клетка с ровно тремя живыми соседями становится живой;

Кроме этого, правило может быть задано вектором, в котором каждое значение - одно из состояний клетки. Выбор этого значения зависит от конкретного состояния окрестности |D| в данный момент времени.

3 Особенности реализации

Лабораторная работа реализована на языке Python.

Правило КА задается вектором 6509750.

Используются нейтральные граничные условия, то есть клетки за границами клеточного автомата считаются пустыми.

При создании поля пользователь через консоль вводит:

Размер поля;

Продолжительность такта;

Способ задания начального состояния КА - автоматически или вручную.

Пример ввода данных через косоль можно увидеть на на рис. 2.

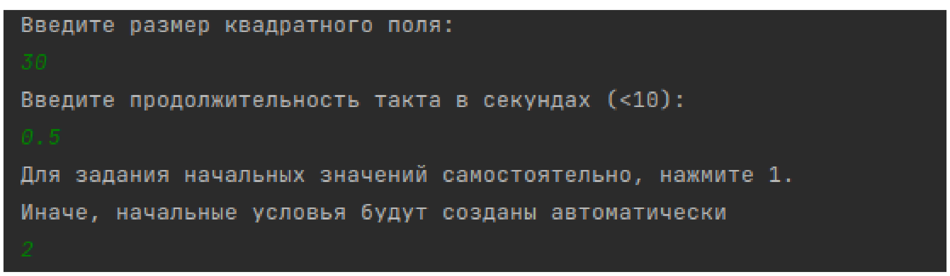

Рис. 2: Ввод данных для создания клеточного автомата

На рис. 3 можно увидеть созданное поле клеточного автомата с начальным состоянием, заданным автоматически:

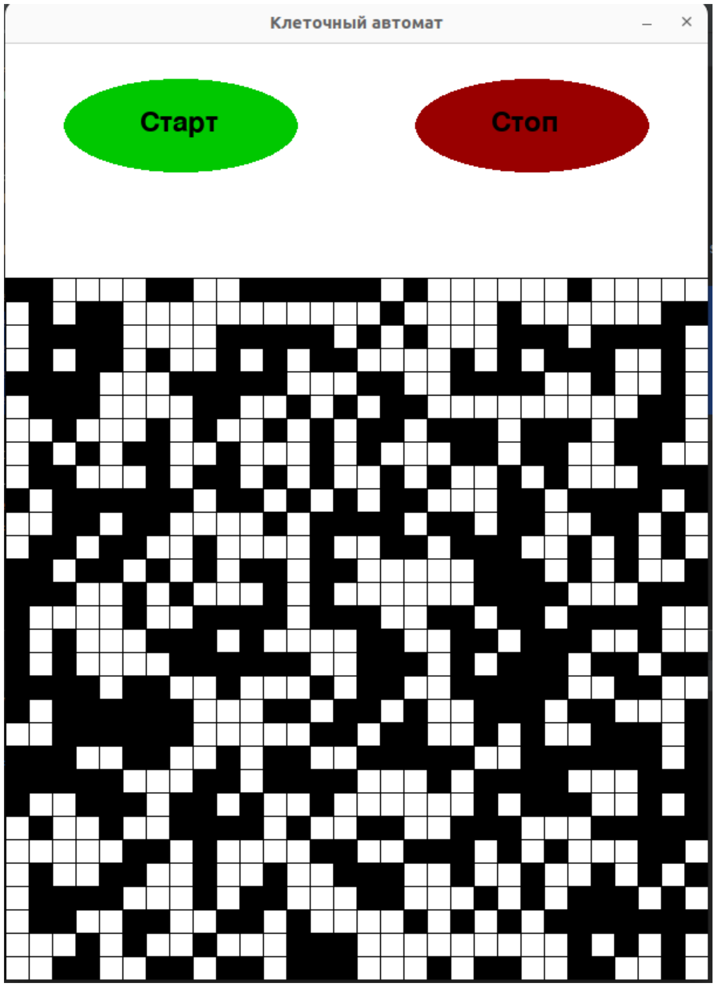

Рис. 3: Поле клеточного автомата

Ниже приведена реализация основных функций данной работы:

3.1 Реализация одного такта КА

Вход: Вектор правила, матрица КА на текущем такте.
Выход: Матрица КА на следующем такте.

```
def run_one_takt_roles(self):
    matrix = copy.deepcopy(self.matrix)
    for i in range(0,self.n):
        for j in range(0,self.n):
            amount = self.get_value_neighbors_fon_neiman(i, j)
            matrix[i][j] = vector_roles[amount]
    self.matrix = matrix
```

3.2 Получение состояния окрестности Фон Неймана

Вход: Номер строки клетки, номер столбца клетки, матрица состояния КА.

Выход: Значение, обозначабщее состояние окрестности для дальнейшего поиска в векторе правил КА.

```
def get_value_neighbors_fon_neiman(self, row, col):
    value=0
    value += self.matrix[row][col] *16
    if 0<=row-1<self.n:
        value+=self.matrix[row - 1][col]*2

    if 0<=row+1<self.n:
        value += self.matrix[row + 1][col]*4
    if 0 <= col - 1 < self.n:
        value += self.matrix[row][col - 1] * 8
    if 0 <= col + 1 < self.n:
        value += self.matrix[row][col + 1] * 1
    return value
```

4 Результат работы программы

На рис. 4 - 8 можно увидеть результат работы программы при разных размерах поля. В левой части скриншота показано состояние клеточного, заданное случайно. В правой части показано состояние автомата через количество тактов, достаточное для того, чтобы автомат стабилзировался.

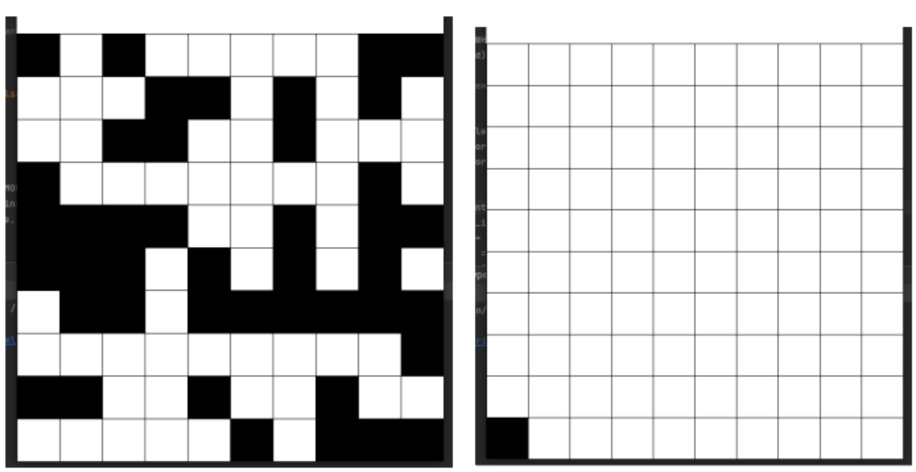

Рис. 4: Запуск программы на поле размера 10*10

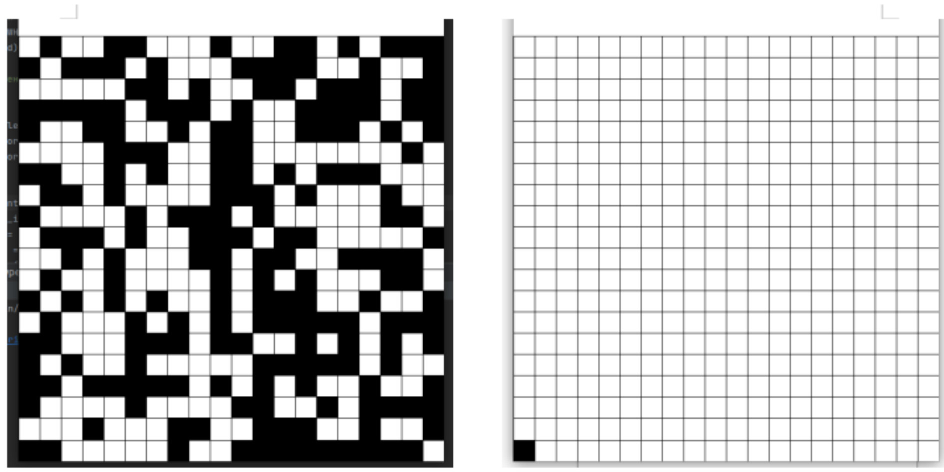

Рис. 5: Запуск программы на поле размера 20*20

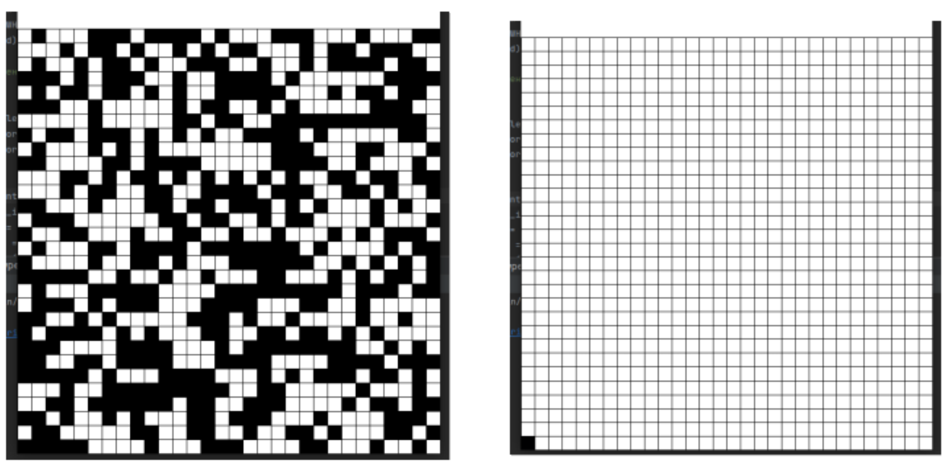

Рис. 6: Запуск программы на поле размера 30*30

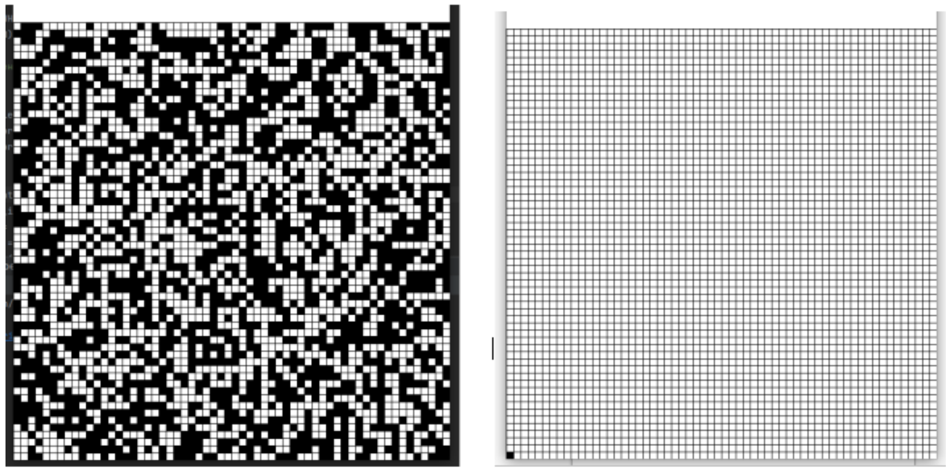

Рис. 7: Запуск программы на поле размера 70*70

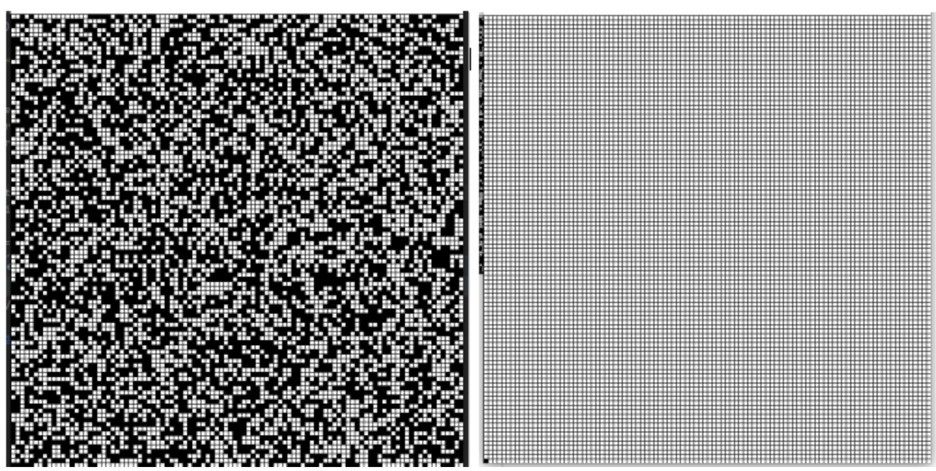

Рис. 8: Запуск программы на поле размера 100*100

По рис. 4 - 8 видно, что клеточный автомат при каждом запуске приходит в стабильное состояние, а значит его можно отнести к классу №1 по классификации Вольфрама.

Так же в ходе анализа автомата было замечено, что «живые» клетки автомата всегда сдвигаются вниз и влево, что можно увидеть на рис. 9. В левом верхнем углу представленно начальное состояние автомата, внизу скриншота конечное состояние автомата.

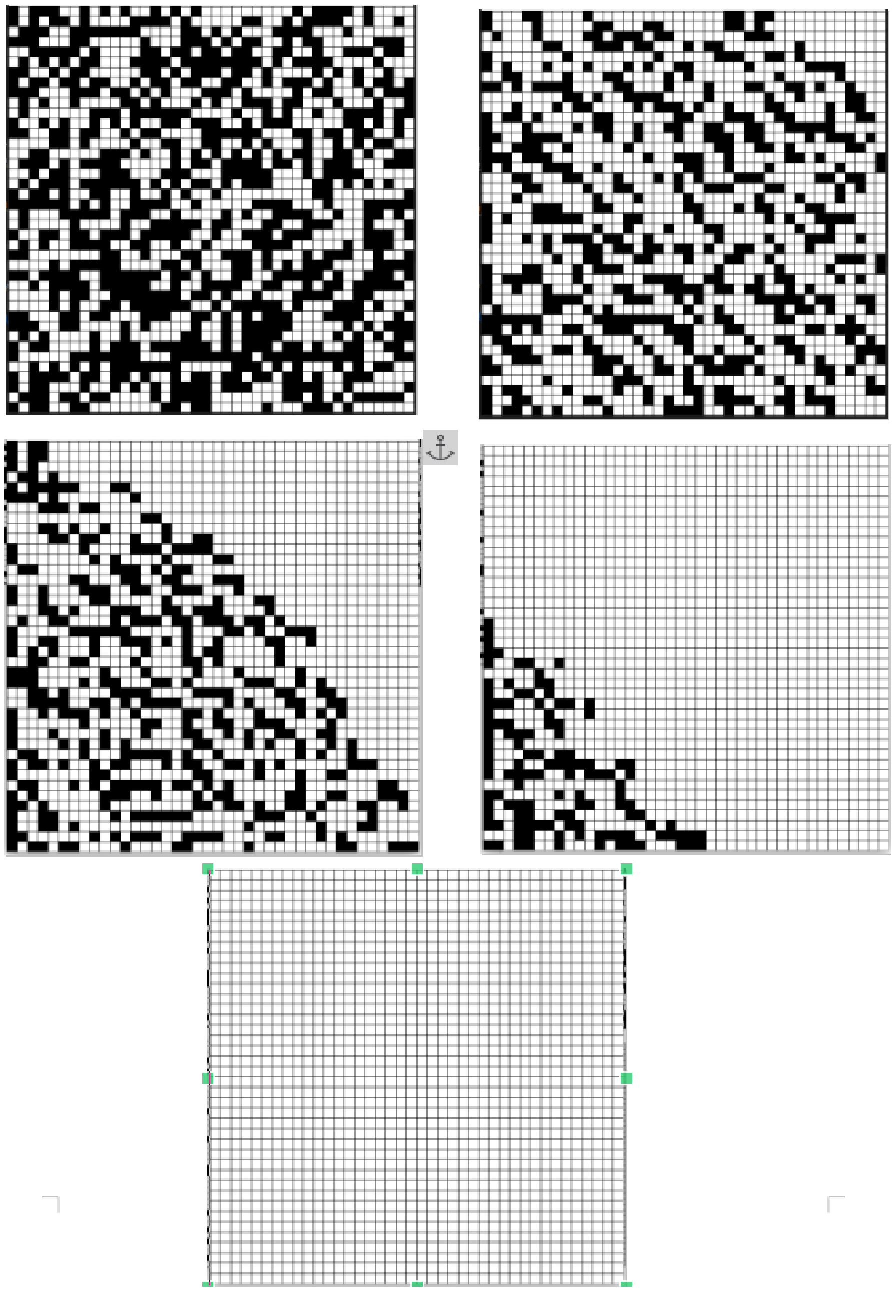

Рис. 9: Эволюция случайно заданного клеточного автомата. Поле размера 40*40

Заключение

В заключение можно сказать, что в ходе выполнения работы был реализован двумерный клеточный автомат с окресностью Фон Неймана в соответствии с правилом 6509750.

Граничные уловия - нейтральные, то есть клетки за границами поля по умолчанию являются пустыми.

Пользователь сам определяет ширину поля клеточного автомата.

Начальные условия могут вводиться как пользователем, так и задаваться случайным образом.

Было выявлено, что построенный автоматотносится к классу №1 по класификации Вольфрама. Автомат всегда стабилизируется через достаточно большое количество операций.

К **преимуществам реализации** можно отнести:

Реализация автомата с графическим интерфейсом.

Состояние ячеек клеточного автомата можно изменять в процессе работы программы, путем нажатия на ячейку.

К **недостаткам реализации** можно отнести:

Невозможность запустить работу автомата по одной итерации(Это возможно, если задать достаточно большой переод такта и каждый раз нажимать на кнопку СТОП, но это неудобно)

Масштабирование:

Программу легко масштабировать путем задания других правил, и другого способа заания правил, т.к. все функции реализованны независимо друг от друга и изменение одной из них не повлеечет необходимость в изменении остальных.

Список использованных источников

[1] Ф.А Новиков. Дискретная математика для программистов: Учебник для вузов. 3-е изд. — СПб: Питер, 2008. —384 с.

[2] А.В. Вострову. Лекция 4 по дисциплине «Теория алгоритмов».2023г. 77 сл.

[3] Стивен Вольфрам.«A New Kind of Science» [Электронный ресурс] URL: https://www.wolframscience.com/nks/ (Дата обращения: 19.12.2023)