

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра информатики, математического  
 и компьютерного моделирования**

**ОТЧЕТ**

о выполнении лабораторной работы «Компьютерная модель эпидемии»

по предмету «Математическое и компьютерное моделирование»

Направление подготовки

01.03.02 «Прикладная математика и информатика»

Выполнил  
студент группы Б8117-01.03.02

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Бондаревский А.С.

Проверил д.ф.-м.н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Пермяков М. С.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

**г. Владивосток**

**2020**

Оглавление

[Цель 3](#_Toc31150929)

[Формализация задачи 3](#_Toc31150930)

[Описание 3](#_Toc31150931)

[Метод решения 3](#_Toc31150932)

[Анализ модели 7](#_Toc31150933)

[Вывод 8](#_Toc31150934)

# Цель

Цель данной лабораторной работы – реализовать компьютерную модель эпидемии с помощью клеточного автомата.

# Формализация задачи

Задача данной лабораторной работы – написать программу с помощью языка программирования Python, которая реализует эпидемию с помощью клеточного автомата, и получить статистику на основе данной программы.

# Описание

Клеточный автомат – дискретная модель, включающая решётку ячеек, каждая из которых может находиться в одном из конечного множества состояний, таких как:

* 0 – Здоровый человек;
* 1 – Больной человек;
* 2 – Выздоравливающий человек;
* 3 – Переболевший и получивший иммунитет человек;

Решетка может быть любой размерности. Для работы клеточного автомата требуется задание начального состояния всех ячеек и правила перехода ячеек из одного состояния в другое.

При этом будем считать, что эпидемия начинается, если более 5% клеток находятся в состоянии 1 или 2.

А также существует ряд параметров:

* Вероятность заразить соседнюю клетку; (От 0 до 1)
* Количество клеток вокруг заразной, которую она способна заразить; (От 1 до 8)
* Размерность решетки;

# Метод решения

Для выполнения поставленной задачи воспользуемся средствами языка программирования Python, библиотеки numpy, pygame, pylab, random.

**Код программы:**

**import** random  
**import** pylab  
**import** pygame  
**import** numpy **as** np  
  
BLACK = (0, 0, 0)  
WHITE = (255, 255, 255)  
GREY = (128, 128, 128)  
GREEN = (0, 255, 0)  
RED = (255, 0, 0)  
WIDTH = 8  
HEIGHT = 8  
MARGIN = 1  
count\_rect = 50  
  
  
**def** paint\_matrix(matrix, screen, time):  
 screen.fill(BLACK)  
 **for** i **in** range(count\_rect):  
 **for** j **in** range(count\_rect):  
 **if** matrix[i][j] == 0:  
 color = WHITE  
 **elif** matrix[i][j] == 1:  
 color = GREY  
 **elif** matrix[i][j] == 2:  
 color = RED  
 **elif** matrix[i][j] == 3:  
 color = GREEN  
 pygame.draw.rect(screen, color,  
 [(MARGIN + WIDTH) \* j + MARGIN, (MARGIN + HEIGHT) \* i + MARGIN, WIDTH, HEIGHT])  
 time.tick(10)  
 pygame.display.flip()  
  
  
**def** initialize\_model\_matrix():  
 matrix = np.zeros((count\_rect, count\_rect))  
 **return** matrix  
  
  
**def** set\_cells(matrix, count):  
 cells\_count = count  
 random\_dict = {}  
 **for** i **in** range(count\_rect):  
 random\_dict[i] = []  
 **while** cells\_count > 0:  
 random\_row = random.randint(0, count\_rect-1)  
 random\_col = random.randint(0, count\_rect-1)  
 **if** random\_col **not in** random\_dict[random\_row]:  
 matrix[random\_row, random\_col] = 1  
 random\_dict[random\_row].append(random\_col)  
 cells\_count -= 1  
  
  
**def** check\_cells(matrix):  
 **for** i **in** range(count\_rect):  
 **for** j **in** range(count\_rect):  
 **if** matrix[i][j] **in** [1, 2]:  
 **return True  
 return False  
  
  
def** count\_cells(matrix):  
 count = 0  
 **for** i **in** range(count\_rect):  
 **for** j **in** range(count\_rect):  
 **if** matrix[i][j] **in** [1, 2]:  
 count = count + 1  
 **return** count  
  
  
**def** model\_cells(matrix, q, p, screen, time):  
 q\_cells = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 0), (1, 1)]  
 iterations\_result = 0  
 **while** check\_cells(matrix):  
 **for** i **in** range(count\_rect):  
 **for** j **in** range(count\_rect):  
 **if** matrix[i][j] == 0:  
 q\_count = q  
 **if** (i **in** [0, count\_rect-1]) **and** (j **in** [0, count\_rect-1]) **and** (q\_count > 3):  
 q\_count = 3  
 **if** ((((i **in** [0, count\_rect-1]) **and** (j **in** [x **for** x **in** range(1, count\_rect-1, 1)])) **or** ((i **in** [x **for** x **in** range(1, count\_rect-1, 1)]) **and** (j **in** [0, count\_rect-1]))) **and** (q\_count > 5)):  
 q\_count = 5  
 used\_indexes = []  
 selected\_cells = []  
 **while** q\_count > 0:  
 random\_index = random.randint(0, 7)  
 **if** ((random\_index **not in** used\_indexes) **and** (count\_rect > i + q\_cells[random\_index][0] > -1) **and** (count\_rect > j + q\_cells[random\_index][1] > -1)):  
 used\_indexes.append(random\_index)  
 selected\_cells.append(q\_cells[random\_index])  
 q\_count -= 1  
 **for** cell **in** selected\_cells:  
 **if** matrix[i + cell[0]][j + cell[1]] **in** [1, 2]:  
 random\_p\_num = random.random()  
 **if** random\_p\_num < p:  
 matrix[i][j] = 1  
 **elif** matrix[i][j] == 1:  
 matrix[i][j] = 2  
 **elif** matrix[i][j] == 2:  
 matrix[i][j] = 3  
 **if** count\_cells(matrix) > 50:  
 iterations\_result += 1  
 paint\_matrix(matrix, screen, time)  
 **return** iterations\_result  
  
  
model\_matrix = initialize\_model\_matrix()  
set\_cells(model\_matrix, 10)  
q\_parameters = []  
iterations = []  
SIZE = HEIGHT \* count\_rect  
WINDOW\_SIZE = [SIZE, SIZE]  
screen = pygame.display.set\_mode(WINDOW\_SIZE)  
**for** q **in** range(1, 9, 1):  
 time = pygame.time.Clock()  
 current\_matrix = np.copy(model\_matrix)  
 iterations.append(model\_cells(current\_matrix, q, 0.5, screen, time))  
 q\_parameters.append(q)  
pylab.plot(q\_parameters, iterations)  
pylab.show()

# Анализ модели

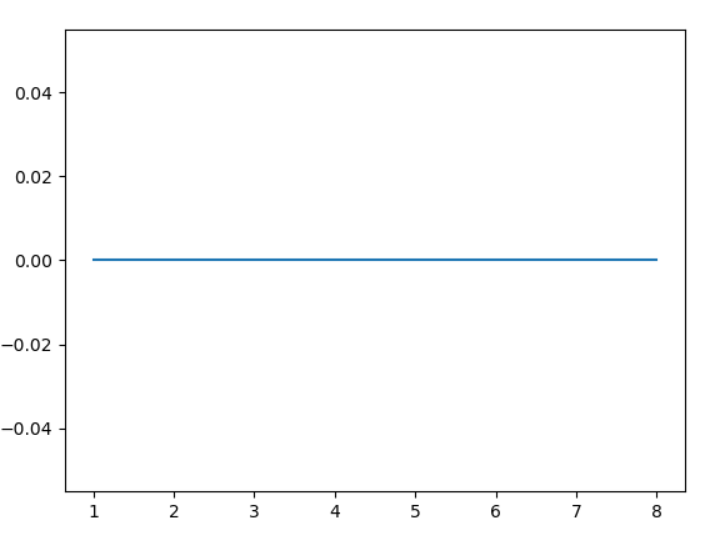
Количество ячеек в строке и столбце = 50

p – Вероятность заразить клетку;

Ось ординат – кол-во итераций во время эпидемии;

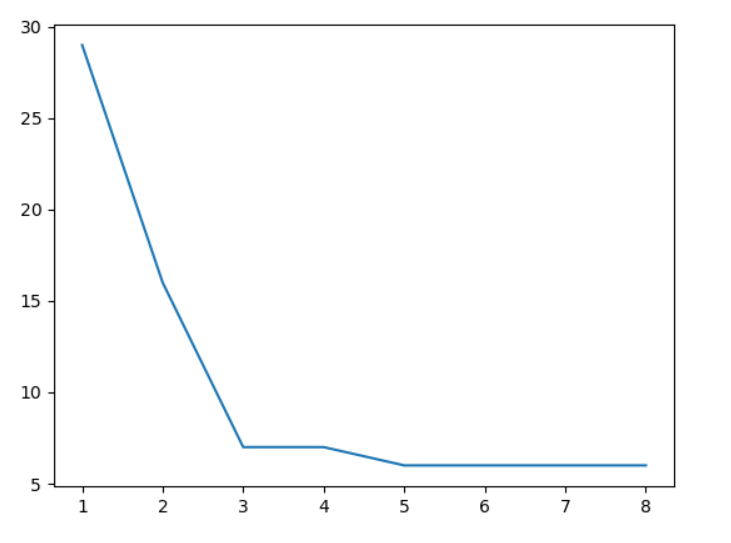
Ось абсцисс – кол-во ячеек, которые может заразить клетка;(q)

p = 0



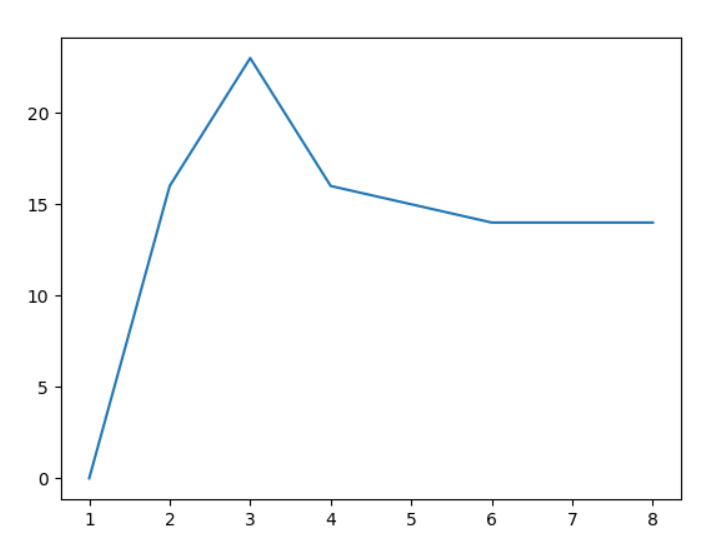
Очевидно, что при значении p = 0 (То есть, шанса заразить клетки нет) количество дней эпидемии равно 0;

p = 1



Так же очевидно, что при 100% заражении, самая длительная эпидемия при наименьшем кол-ве клеток, которых способна заразить болеющая клетка.

p = 0.5



Из последних двух примеров видно, что эпидемия длится дольше всего, при достаточном проценте заболевания и достаточно малом количестве q.

# Вывод

Таким образом, мы построили компьютерную модель эпидемии с помощью клеточного автомата и получили статистику на основе неё.