Группа	3532703/00001
ФИО	Кимельфельд Дмитрий Петрович

Отчёт №10.

Дата сдачи 28.11.2020 г.

Задание: Написать программу, моделирующую размножение бактерий в чаше Петри.

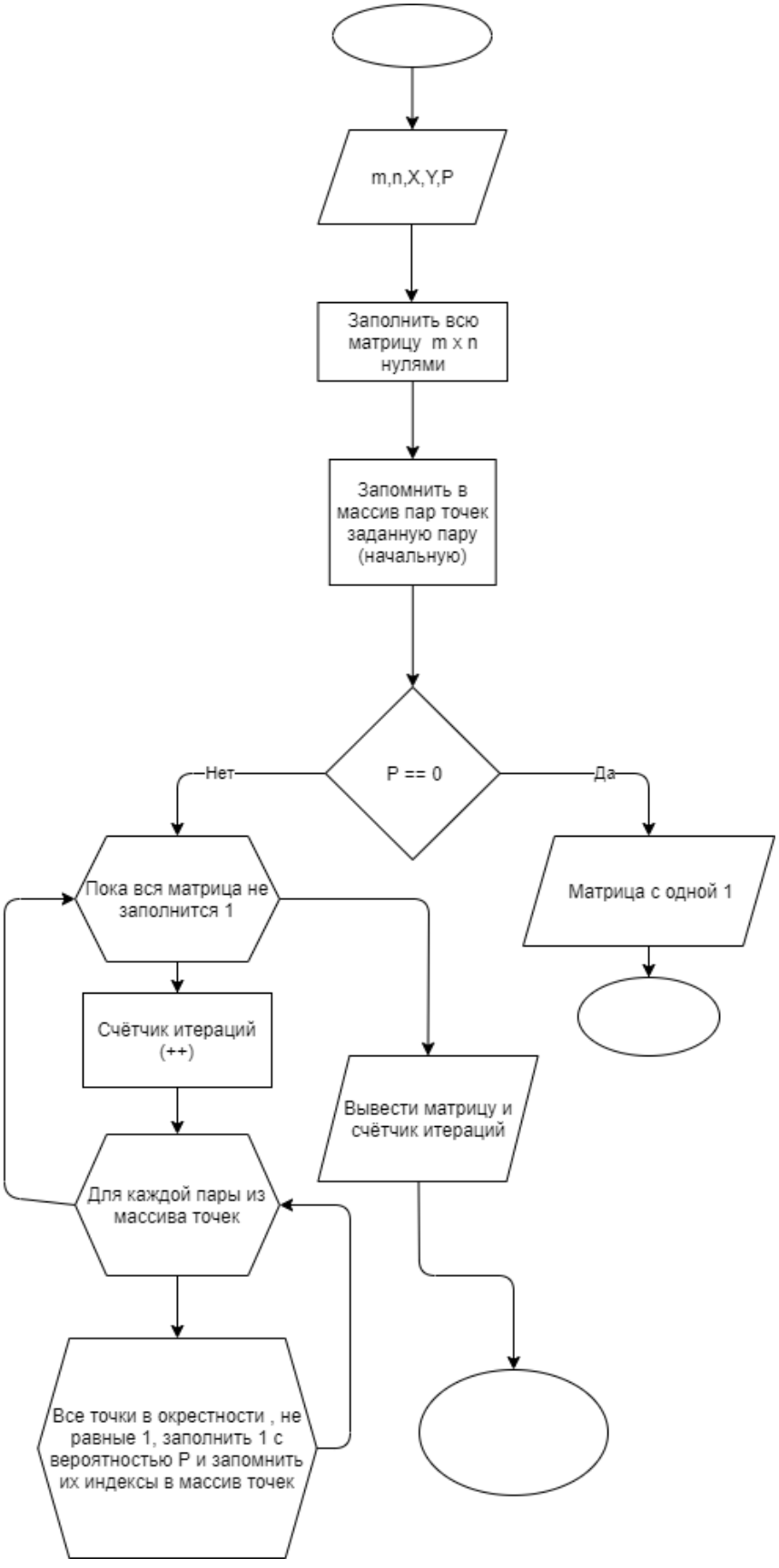
- Данные на вход: размеры матрицы m x n, координаты первой бактерии (x,y), такие, что 1<=x<=m, 1<=y<=n, 0<=P<=1 – вероятность генерации новой бактерии в клетке.
- На первом шаге матрица размерами m x n заполнена нулями. На втором шаге появляется одна бактерия в клетке (x, y). На третьем шаге в клетках, соседних с первой клеткой генерируются новые бактерии с вероятностью P. Соседними считаются клетки по вертикали, горизонтали и диагонали, координаты которых отличаются от исходной не более, чем на 1. «Новая» клетка, которая появилась на данном шаге, не генерирует бактерии на этом же шаге, но начинает генерировать бактерии на следующем и всех последующих шагах, пока не заполнит все соседние клетки.
- На каждом шаге выводить на экран номер шага и матрицу
- Для заданных **m и n** вычислить число итераций (шагов) N, за которое заполнится матрица, при различных **x, y, P**. Как число итераций зависит от указанных параметров?
- Для фиксированных m,n,x,y,P вычислить среднее число итераций <N> за k экспериментов. Как меняется число итераций в зависимости от k?
- Рекомендуемые параметры m=n=10, m=n=30, m=n=100, 1<=x<=m, 1<=y<=n, P=0, 0.05, 0.1, 0.2, 0.5, 0.6, 0.9, 1

Оглавление

Описание алгоритма решения задачи

Создаётся массив пар, в который запоминаем индексы элементов матрицы, которые уже заполнили единицами. Затем, проходя каждую пару индексов, вокруг каждой точки проверяю остальные элементы. Если единица уже установлена в соседнем месте, то переходим к следующей соседней точке, до тех пор, пока не проверим все соседние места. Индексы элементов, которые заполнили единицами, запоминаем также в массив пар точек.

Блок-схема программы



Текст программы

```
#include <iostream>
using namespace std;
struct Pair
{
	int x;
	int y;
};
int main() {
	setlocale(LC_ALL, "Russian");
	int n; // m - строк , n  - столбцов
	int m;
	int P;
	int i0, j0;
	cout << "Введите количество строк в массиве: ";
	cin >> m;
	cout << endl << "Введите количнство столбцов в массиве: ";
	cin >> n;
	cout << endl << "Введите вероятность генерации новой бактерии в клетке: ";
	cin >> P;
	cout << endl << "Введите координаты изначальной бактерии: ";
	cin >> i0 >> j0;

	int **matrix = new int*[m];
	Pair *points = new Pair[m*n]; // points [0].x;  points[0].y


	for (int i = 0; i < m; i++) {
		matrix[i] = new int[m];
	}
	for (int i = 0; i < m; i++) {
		for (int j = 0; j < n; j++, cout << "\t") {
			matrix[i][j] = 0;
			cout << matrix[i][j];
		}
		cout << endl;
	}
	cout << endl;
	int k = 1; // Следующая память под точку из массива
	int g = 0;
	int limit = 1;
	points[0].x = i0;
	points[0].y = j0;
	matrix[points[0].x][points[0].y] = 1;


	for (int i = 0; i < m; i++) {
		for (int j = 0; j < n; j++, cout << "\t") {
			cout << matrix[i][j];
		}
		cout << endl;
	}
	if (P == 0) {
		return 0;
	}
	int size = (m * n) - 1;
	int it = 0;
	while (size != 0) { // Заполнение матрицы
		it++;
		for (int pointer = 0 ; pointer < limit; pointer++ ) { // Проход по массиву
точек
			i0 = points[pointer].x;
			j0 = points[pointer].y;
			for (int i = i0 - 1; i <= i0 + 1; ++i) { // Заполнение бактериями
```

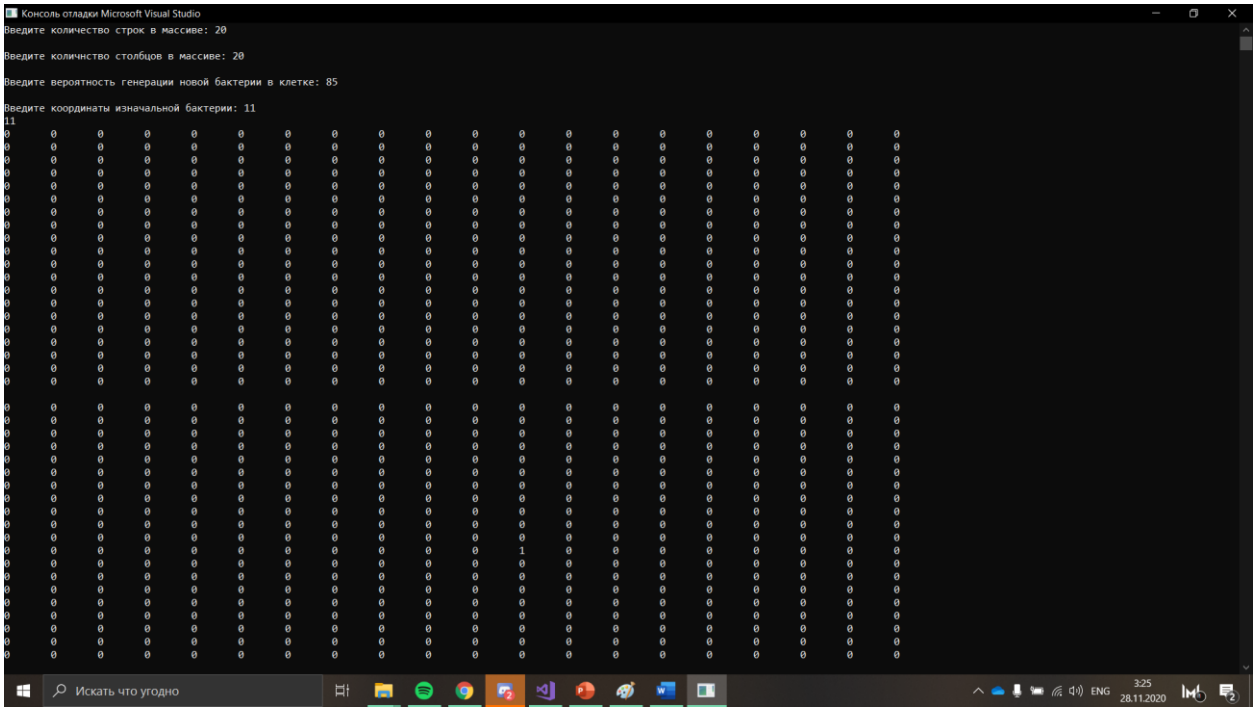
```
                        for (int j = j0 - 1; j <= j0 + 1; ++j) {
                              if ((0 <= i) && (i < m ) && (0 <= j) && (j < n) && (i
!= i0 || j != j0)) {
                                    if (((rand() % 100) <= P) && (matrix[i][j] !=
1)) {
                                          matrix[i][j] = 1;
                                          size--;
                                          points[k].x = i; // Запоминает точки,
заполненные единицами
                                          points[k].y = j; // от 1 до p
                                          k++;
                                          g++;
                                    }
                              }
                        }
                  }
            }
            cout << endl;
            for (int i = 0; i < m; i++) {
                  for (int j = 0; j < n; j++, cout << "\t") {
                        cout << matrix[i][j];
                  }
                  cout << endl;
            }
            limit +=g ;
            g = 0;
      }
      cout << endl << "Количество итераций:" << it;
      return 0;
}
```

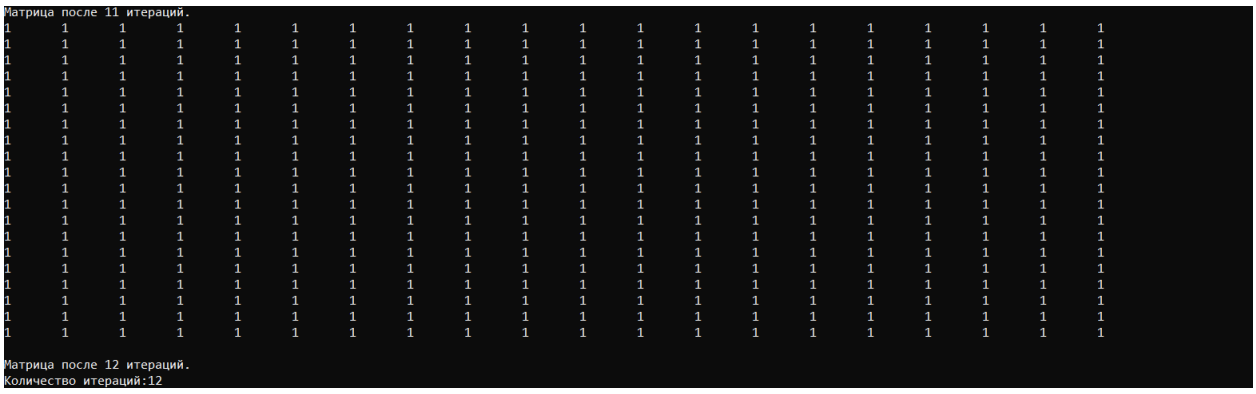
Перечень ошибок при отладке программы

1. При установление лимита неправильно записал суммирование
(=+) , а не (+=) . Из- за этого цикл заполнения зацикливался.

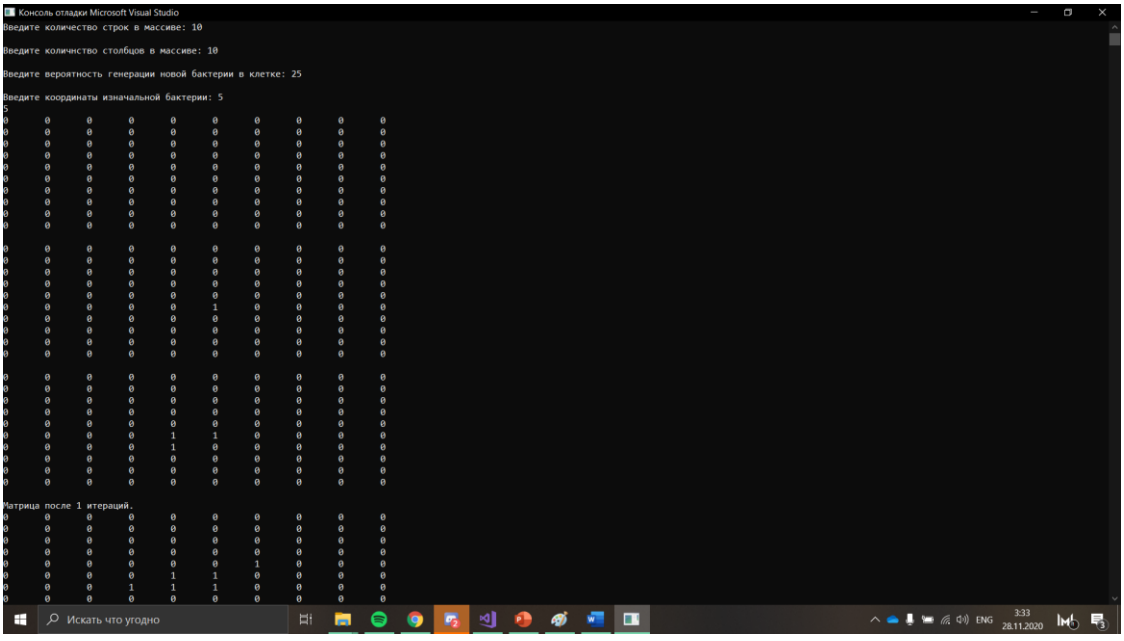
Тестовые примеры работы программы

Пример 20x20

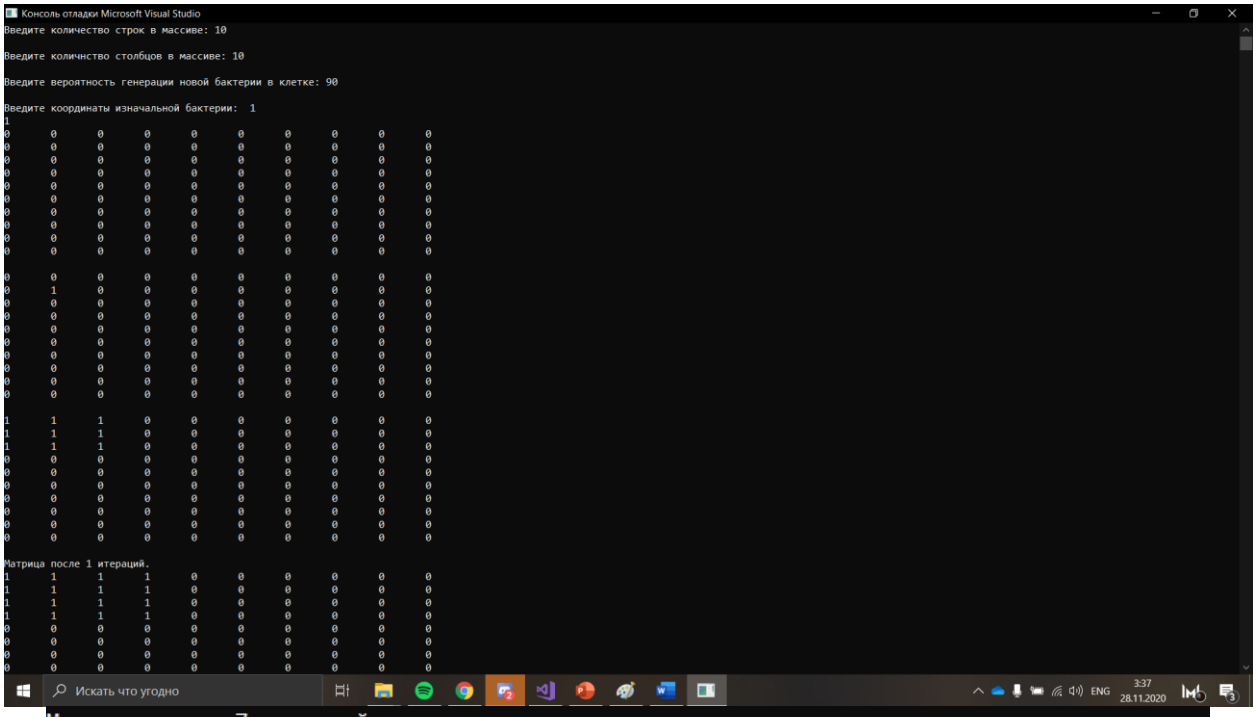




Примеры 10x10



```
Матрица после 9 итераций.
1       1       1       1       1       1       1       1       1       0
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       0
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       0       0       1       1

Матрица после 10 итераций.
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1

Матрица после 11 итераций.
Количество итераций:11
```

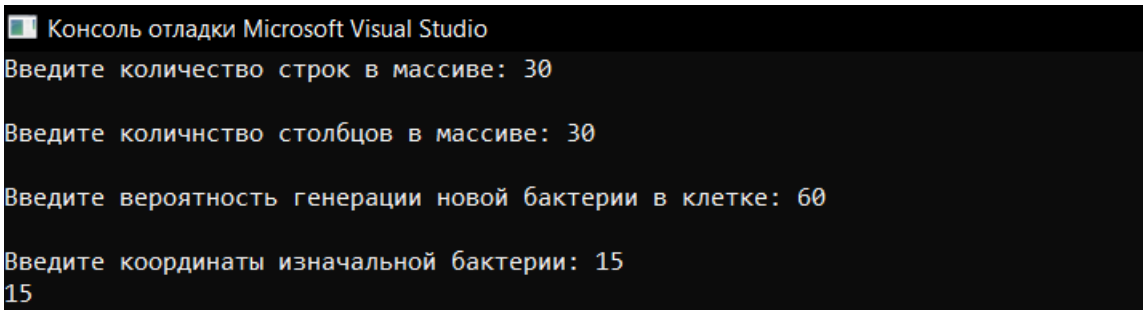



```
Матрица после 7 итераций.
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1

Матрица после 8 итераций.
Количество итераций:8
```

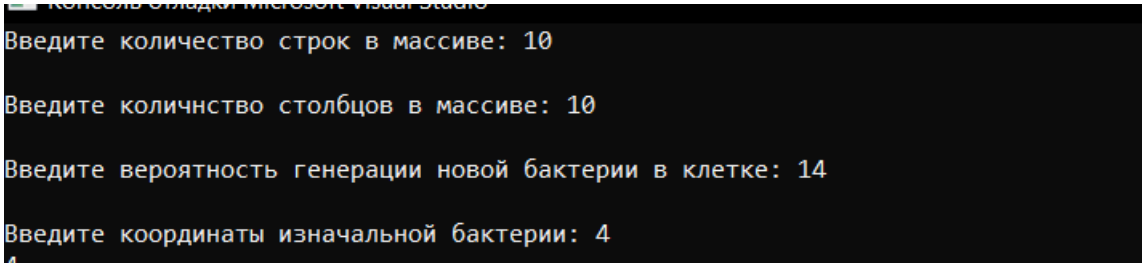
```
Введите количество строк в массиве: 10

Введите количнство столбцов в массиве: 10

Введите вероятность генерации новой бактерии в клетке: 60

Введите координаты изначальной бактерии: 9
9
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0

0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       1

0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       1       0
0       0       0       0       0       0       0       0       1       1

Матрица после 1 итераций.
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       1       0
```

```
0       0       0       0       0       1       1       1       1       0
0       1       1       1       1       1       1       1       1       1
0       1       1       1       1       1       1       1       1       1
0       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
0       1       1       1       1       1       1       1       1       1
0       1       1       1       1       1       1       1       1       1
0       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1

Матрица после 9 итераций.
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
0       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1

Матрица после 10 итераций.
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1
1       1       1       1       1       1       1       1       1       1

Матрица после 11 итераций.
Количество итераций:11
```

При фиксированных данных.



№	Кол-во итераций
1	20
2	20
3	18
4	19
5	19
6	19
7	18
8	20
9	20
10	18
<it>=19	



№	Кол-во итераций
1	15
2	20
3	16
4	18
5	20
6	18
7	18
8	18
9	16
10	18
11	15
12	18
13	18
14	20
15	19
<it>=18	

Выводы

В ходе этой работы я изучил динамические массивы и структуры, пронаблюдал особенность реализации rand () , которая выдает одинаковые “псевдослучайные” числа для каждого запуска программы.

Из работы со статистикой я заключил, что при равномерном уменьшении вероятности количество итераций непропорционально увеличивается, из-за того, что “размножение” — это геометрическая прогрессия: более низкая вероятность является причиной увеличения итераций относительно более высоких вероятностей.