Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине: «Общесистемное программное обеспечение параллельных вычислительных систем»

Студент	Гольцов Илья Сергеевич
Группа	РК6-32М
Тип задания	лабораторная работа №1
Тема лабораторной работы	Волновой алгоритм «Эхо»

Студент _____ **Гольцов И. С.**
подпись, дата *фамилия, и.о.*

Преподаватель _____ **Грошев С. В.**
подпись, дата *фамилия, и.о.*

Оценка _____

Москва, 2022 г.

Оглавление

Задание на лабораторную работу

Написать программу иллюстрирующую работу волнового алгоритма «Эхо» для распределенной сети, представленной в виде заданного графа.

Выполнение лабораторной работы

Алгоритм эха - централизованный алгоритм для сетей с произвольной топологией. Алгоритм выделяет остовное дерево путем наводнения сообщениями toc. Разосланные маркеры возвращаются методом «Эха» обратно по ребрам дерева. Таким образом, сценарий работы алгоритма:

1. Инициатор отправляет сообщения всем соседям;
2. После получения первого сообщения любой инициализатор перенаправляет сообщения всем своим соседям, за исключением того, от которого было получено это сообщение;
3. Как только инициатор получит сообщение от всех своих соседей, он отправляет эхо родительскому процессу;
4. Как только инициатор получит сообщения от всех своих соседей, он принимает решение.

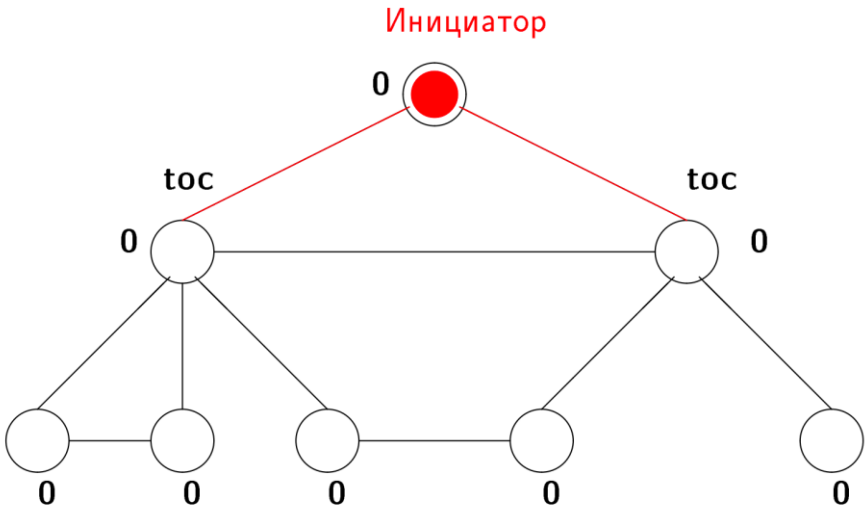

Рисунок 1 - Шаг алгоритма №1

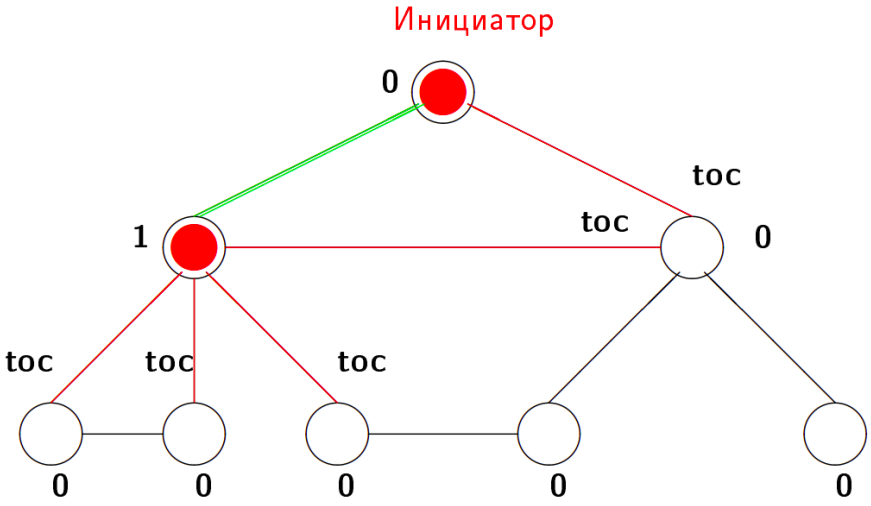

Рисунок 2 - Шаг алгоритма №2

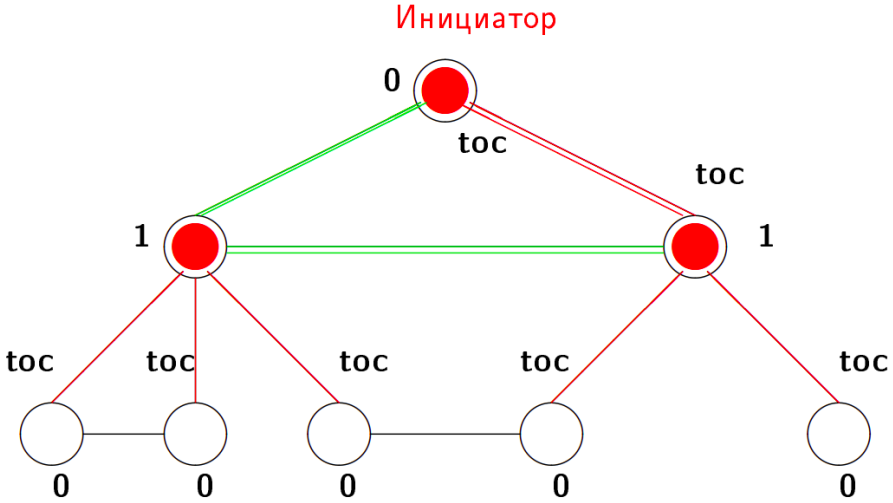

Рисунок 3 - Шаг алгоритма №3

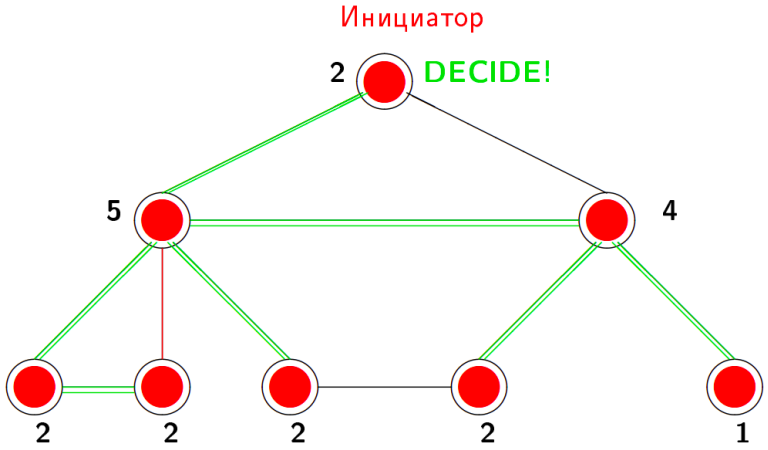

Рисунок 4 – Заключительный шаг алгоритма

Ниже представлен псевдокод волнового алгоритма «Эхо»:

```
var  rec_p    : integer init 0; (*Подсчет числа принятых сообщений*)
     father_p : ℙ       init undef ;

Для инициатора:
     begin forall q ∈ Neigh_p do send tok to q ;
           while rec_p < #Neigh_p do
                 begin receive tok ; rec_p := rec_p + 1 end;
           decide
     end

Для неинициатора:
     begin receive tok from neighbor q ; father_p := q ; rec_p := rec_p + 1
           forall q ∈ Neigh_p, q ≠ father_p do send tok to q ;
           while rec_p < #Neigh_p do
                 begin receive tok ; rec_p := rec_p + 1 end;
           send tok to father_p
     end
```

Рисунок 5 - Псевдокод волнового алгоритма «Эхо»

В рамках выполнения лабораторной работы была разработана программа, иллюстрирующая работу волнового алгоритма «Эхо». Решение производилось на графе, представленном на рисунке 6.

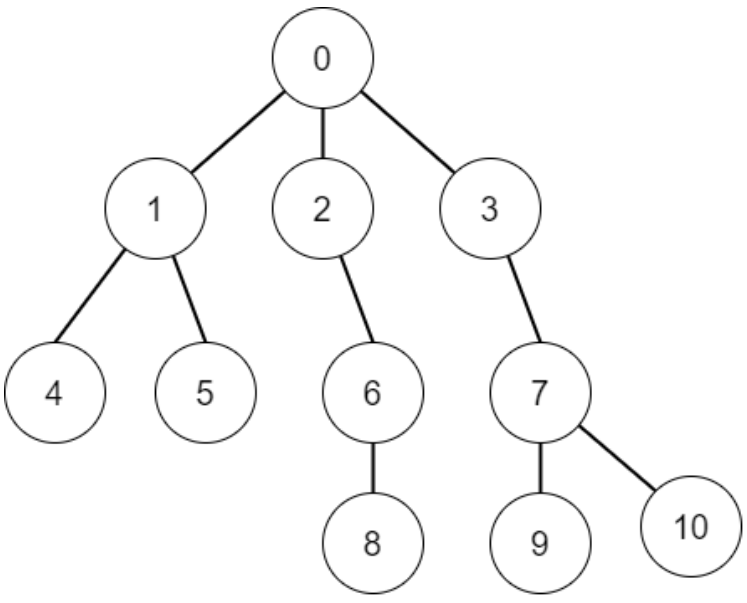

Рисунок 6 – Граф, используемый для решения

Граф задается в программе парами чисел, представляющих его ребра. В начале работы программы происходит считывание структуры графа. На основе заданной структуры строится матрица смежности.

{Adjacency Matrix}

N	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(0)	0	1	1	1	0	0	0	0	0	0	0
(1)	1	0	0	0	1	1	0	0	0	0	0
(2)	1	0	0	0	0	0	1	0	0	0	0
(3)	1	0	0	0	0	0	0	1	0	0	0
(4)	0	1	0	0	0	0	0	0	0	0	0
(5)	0	1	0	0	0	0	0	0	0	0	0
(6)	0	0	1	0	0	0	0	0	1	0	0
(7)	0	0	0	1	0	0	0	0	0	1	1
(8)	0	0	0	0	0	0	1	0	0	0	0
(9)	0	0	0	0	0	0	0	1	0	0	0
(10)	0	0	0	0	0	0	0	1	0	0	0

Рисунок 7 – Полученная программой матрица смежности

Вершиной-инициатором выбирается корневая. Затем происходит рассылка сообщений всем потомкам инициатора. Потомки, получая сообщения, запоминают их родителя и увеличивают счетчик числа полученных сообщений. При этом происходит последовательный перебор всех вершин, что имитирует канал передачи данных между вершинами распределенной сети. Каждая вершина в один момент времени производит некоторую операцию в зависимости от своего состояния. То есть вершина либо производит рассылку сообщений, либо ожидает, либо принимает сообщения. В итоге, когда сообщение доходит до инициатора и общее число сообщений равно удвоенному количеству ребер, алгоритм заканчивает свою работу, происходит принятие дальнейшего решения. Иллюстрация работы алгоритма представлена на рисунке 8.

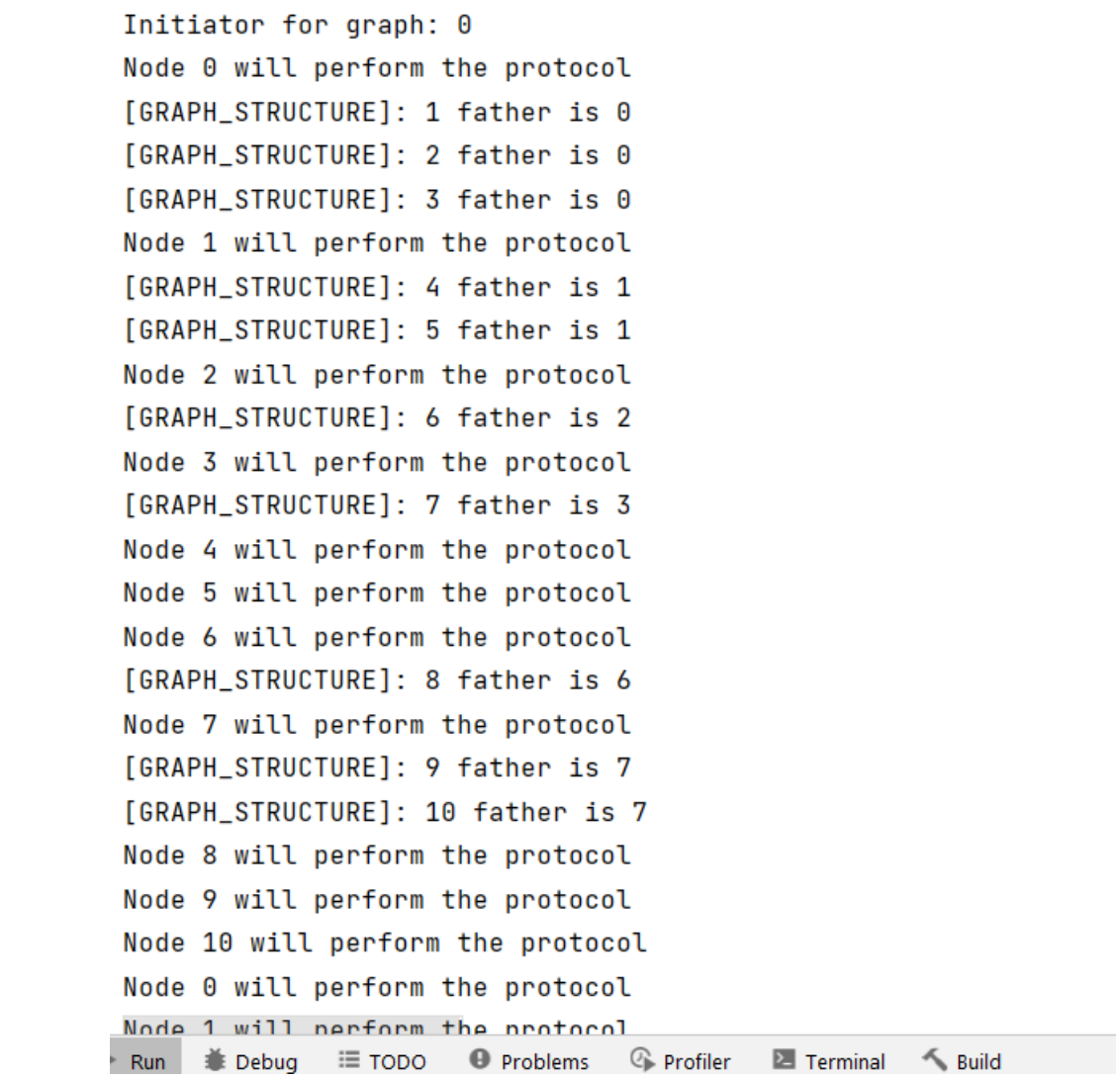

Рисунок 8 – Результат работы разработанного алгоритма

Заключение

В ходе выполнения лабораторной работы был изучен волновой алгоритм «Эхо» и реализован программный код на языке Java, имитирующий работу распределенной сети на заданном графе, в ходе которой при помощи исследуемого алгоритма происходит рассылка сообщений всем вершинам графа и возврат сообщений инициатору, принимающему дальнейшее решение.

Код программы

Ниже представлен программный код, отвечающий за выполнение алгоритма:

Листинг 1 – Функция, отвечающая за выполнение алгоритма

```
public void executeEchoWave(Graph g) {
    int graphSize = g.getGraphSize();
    GraphNode[] nodes = g.getGraphNodes();
    EchoGraph graph = (EchoGraph) g;

    int initiator = 0;
    ((EchoGraphNode) nodes[initiator]).setFather(-1);
    System.out.println("Initiator for graph: " + initiator);

    boolean isEnd = false;
    while (!isEnd) {
        for (int j = 0; j < graphSize; j++) {
            System.out.println("Node " + j + " will perform the protocol");

            if (!nodes[j].isVisited()) {
                if (((EchoGraphNode) nodes[j]).hasFather()) {
                    graph.sentMessagesToNeighbors(j);
                } else {
                    System.out.println("Node " + j + " stayed IDLE. (has not
yet discovered)");
                }
            } else {
                if (j == initiator) {
                    if (((EchoGraphNode) nodes[j]).canDecide()) {
                        isEnd = true;
                        System.out.println("DECISION IS MADE!");
                        if (graph.getEdges() * 2 != graph.getMessages()) {
                            throw new IllegalStateException("THEOREM
INVALID");
                        }
                        break;
                    }
                } else {
                    graph.echo(j);
                }
            }
        }
    }
}
```