

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Дата сдачи на проверку:

«__» _____ 2025 г.

Проверено:

«__» _____ 2025 г.

ГРАФЫ. ОРГРАФЫ. МАТРИЦА СМЕЖНОСТИ.

Отчёт по лабораторной работе №2

по дисциплине

«Дискретная математика»

Разработал студент гр. ИВТб-1301-05-00 _____ /Черкасов А. А./
(подпись)

Проверила преподаватель _____ /Пахарева И. В./
(подпись)

Работа защищена «__» _____ 2025 г.

Киров

2025

Цель

Цель работы: Разработать программу, генерирующую матрицу смежности случайным образом, анализировать полустепень захода вершин графа по заданию.

Задание

- Орграф задается матрицей смежности, которая формируется случайным образом. Размерность $4 \leq n \leq 10$ вводится с клавиатуры.
- Найти кол-во вершин с заданной полустепенью захода, которая вводится с клавиатуры, вывести множества дуг соответственно найденным вершинам (имя множества - номер вершины).

Решение

Схема алгоритма решения представлена на рисунке 1. Примеры работы программы представлены на рисунках 2.1 и 2.2. Исходный код решений представлен в Приложениях A1 и A2.

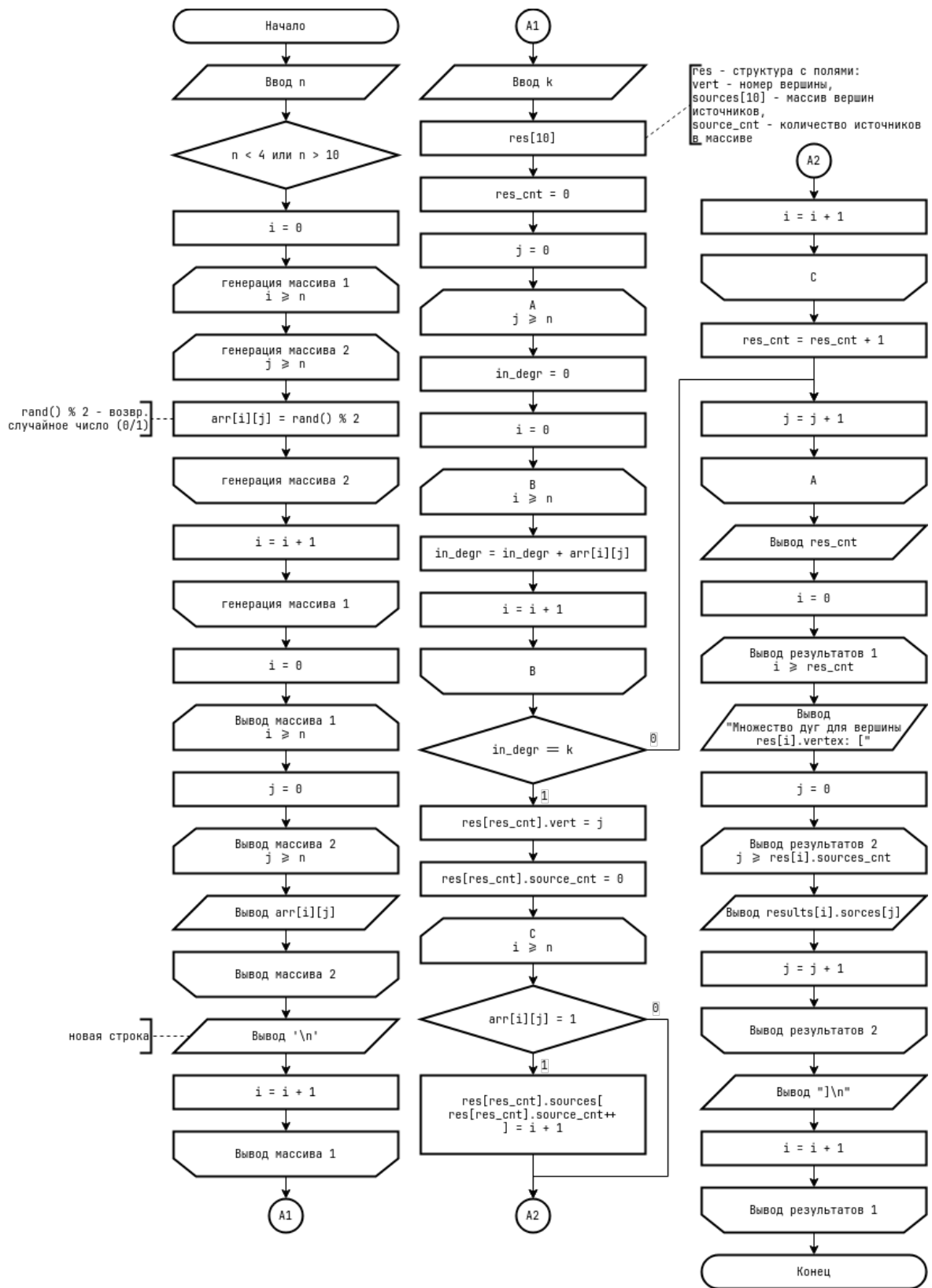


Рисунок 1 - Схема алгоритма основной программы.

```

> cargo run --release -- -viz
    Finished `release` profile [optimized] target(s) in 0.02s
    Running `target/release/lab2 -viz`
Введите размерность матрицы (4-10):
6

Граф сохранён в файле: graph.dot
Для просмотра установите Graphviz и выполните:
dot -Tpng graph.dot -o graph.png && open graph.png

Сгенерированная матрица смежности:
[1, 0, 1, 1, 0, 1]
[0, 0, 0, 1, 0, 1]
[1, 1, 1, 1, 1, 1]
[1, 1, 0, 0, 1, 1]
[1, 1, 0, 0, 1, 0]
[0, 1, 0, 1, 1, 0]

Введите искомую полустепень захода:
2

Результаты:
Найдено вершин: 1
Множество дуг для вершины 3: [1, 3]
> cargo run --release -- -viz
    Finished `release` profile [optimized] target(s) in 0.01s
    Running `target/release/lab2 -viz`
Введите размерность матрицы (4-10):
8

Граф сохранён в файле: graph.dot
Для просмотра установите Graphviz и выполните:
dot -Tpng graph.dot -o graph.png && open graph.png

Сгенерированная матрица смежности:
[0, 1, 1, 1, 1, 1, 0, 1]
[1, 0, 1, 1, 0, 0, 0, 0]
[1, 1, 1, 1, 0, 1, 0, 0]
[1, 0, 1, 0, 1, 1, 0, 0]
[0, 0, 1, 0, 1, 0, 0, 0]
[0, 0, 1, 1, 1, 1, 0, 0]
[0, 0, 0, 1, 1, 0, 0, 1]
[0, 1, 0, 1, 1, 0, 0, 0]

Введите искомую полустепень захода:
4

Результаты:
Найдено вершин: 1
Множество дуг для вершины 6: [1, 3, 4, 6]
> dot -Tpng graph.dot -o graph.png

```

Рисунок 2.1 - Пример работы программы.

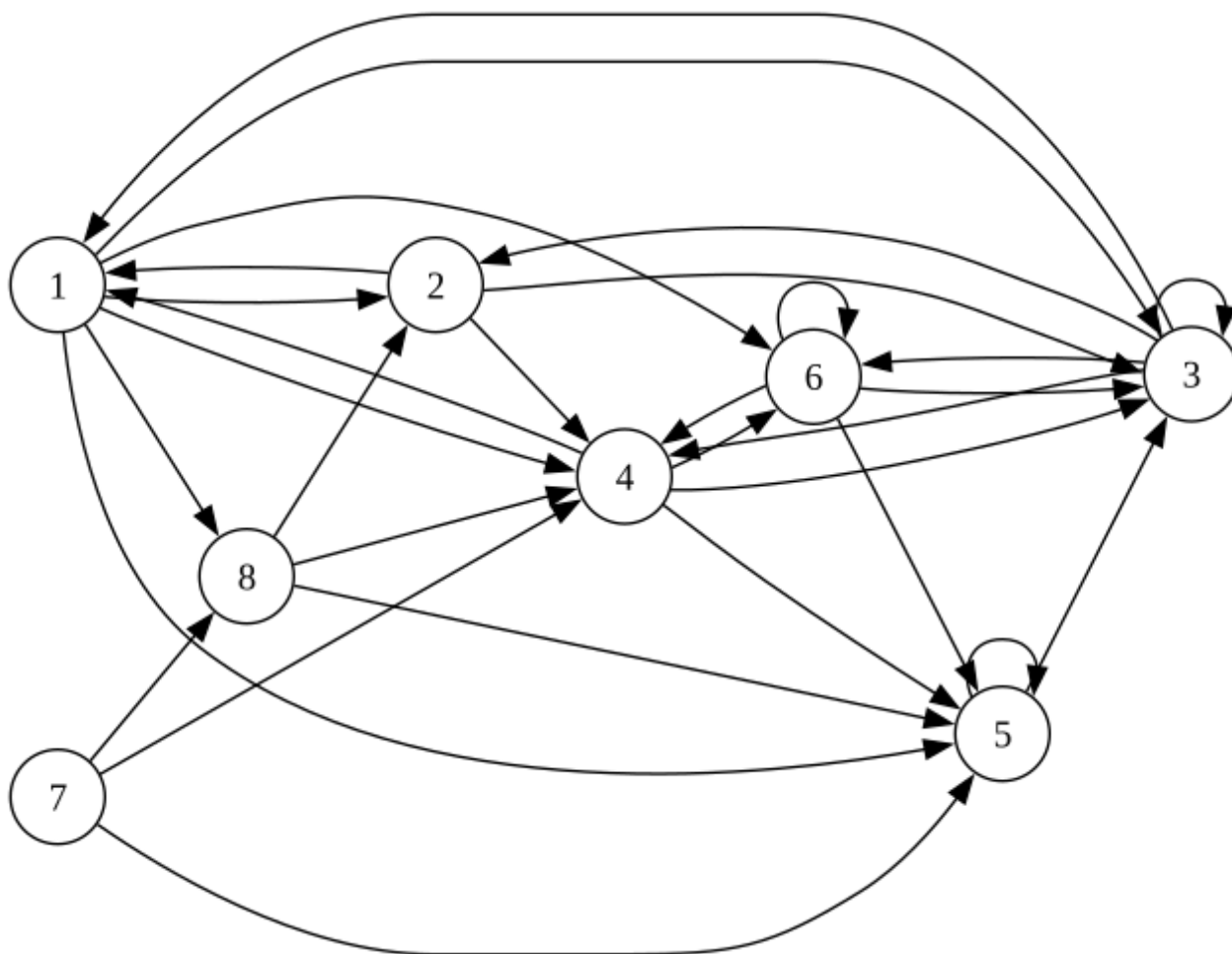


Рисунок 2.2 - Сгенерированный по матрице смежности граф.

Вывод

В ходе выполнения лабораторной работы была реализована программа, генерирующая случайную матрицу смежности для орграфа заданного размера. На основе полученной матрицы произведён анализ полустепеней захода всех вершин графа. Программа позволила определить количество вершин, имеющих заданную пользователем полустепень захода, а также отобразить множества входящих дуг для каждой из таких вершин.

Приложение A1. Исходный код

```
mod graphviz;

use rand::Rng;
use std::env;
use std::io;

fn main() {
    // Обработка аргументов командной строки
    let enable_visualization = {
        let args: Vec<String> = env::args().collect();
        args.contains(&"-viz".to_string()) || args.contains(&"--visualize".to_string())
    };

    let n = loop {
        println!("Введите размерность матрицы (4-10):");
        let mut input = String::new();
        io::stdin().read_line(&mut input).expect("Ошибка чтения");

        match input.trim().parse::<usize>() {
            Ok(num) if (4..=10).contains(&num) => break num,
            _ => println!("Некорректный ввод! Введите число от 4 до 10"),
        }
    };

    let mut matrix = vec![vec![0; n]; n];
    let mut rng = rand::rng();

    for i in 0..n {
        for j in 0..n {
            matrix[i][j] = if rng.random_bool(0.5) { 1 } else { 0 };
        }
    }
}
```

```

if enable_visualization {
    if let Err(e) = graphviz::save_dot(&matrix, n, "graph.dot") {
        eprintln!("Ошибка сохранения графа: {}", e);
        return;
    }
} else {
    println!("Визуализация отключена");
}

// Вывод матрицы
println!("\nСгенерированная матрица смежности:");
for row in &matrix {
    println!("{}", row);
}

let k = loop {
    println!("\nВведите искомую полустепень захода:");
    let mut input = String::new();
    io::stdin().read_line(&mut input).expect("Ошибка чтения");

    match input.trim().parse::<usize>() {
        Ok(num) => break num,
        _ => println!("Некорректный ввод! Введите целое число"),
    }
};

let mut results = Vec::new();

for j in 0..n {
    let in_degree: usize = matrix.iter().map(|row| row[j]).sum();

    if in_degree == k {
        let sources: Vec<usize> = matrix

```

```

        .iter()
        .enumerate()
        .filter(|(i, _)| matrix[*i][j] == 1)
        .map(|(i, _)| i + 1)
        .collect();

    results.push((j + 1, sources));
}
}

println!("\nРезультаты:");
println!("Найдено вершин: {}", results.len());

for (vertex, arcs) in results {
    println!("Множество дуг для вершины {}: {:?}", vertex, arcs);
}
}

```


Приложение А2. Исходный код

```
use std::fs::File;
use std::io::{self, Write};

pub fn save_dot(matrix: &Vec<Vec<usize>>, n: usize, filename: &str) -> io::Result<()> {
    let mut dot = String::new();
    dot.push_str("digraph G {\n");
    dot.push_str("    rankdir=LR;\n");
    dot.push_str("    node [shape=circle];\n");

    for i in 1..=n {
        dot.push_str(&format!("{}", i));
    }

    for i in 0..n {
        for j in 0..n {
            if matrix[i][j] == 1 {
                dot.push_str(&format!("{}", i + 1, j + 1));
            }
        }
    }
    dot.push_str("}\n");

    let mut file = File::create(filename)?;
    file.write_all(dot.as_bytes())?;

    println!("\nГраф сохранён в файле: {}", filename);
    println!("Для просмотра установите Graphviz и выполните:");
    println!("dot -Tpng {} -o graph.png && open graph.png", filename);

    Ok(())
}
```