

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

**Поиск путей в графах**  
Отчёт по лабораторной работе №6  
по дисциплине  
«Дискретная математика»  
Вариант 6

Выполнил студент гр. ИВТ6-1301-05-00	_____ /Макаров С.А./
Руководитель преподаватель	_____ /Пахарева И.В./

Киров 2025

## Цель

Цель лабораторной работы: изучение основ теории графов, методов обходов графов, поиск элементарных путей, разработка приложения на языке Паскаль или СИ согласно заданию.

## Задание

Орграф задан матрицей смежности в файле (вершин  $\geq 5$ , дуг  $\geq 5$ ). Найти возможные элементарные пути (вершины в пути не повторяются) из вершины, имя которой вводится с клавиатуры.

## Решение

Для решения задач подготовлен ориентированный граф, представленный на рисунке 1.

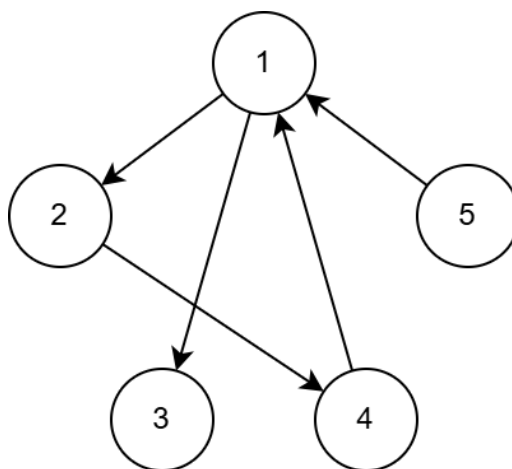


Рисунок 1 – Ориентированный граф

Перед разработкой составлены схемы алгоритмов для решения задач. На рисунке 2 представлены схемы алгоритмов подпрограмм ввода и вывода матрицы.

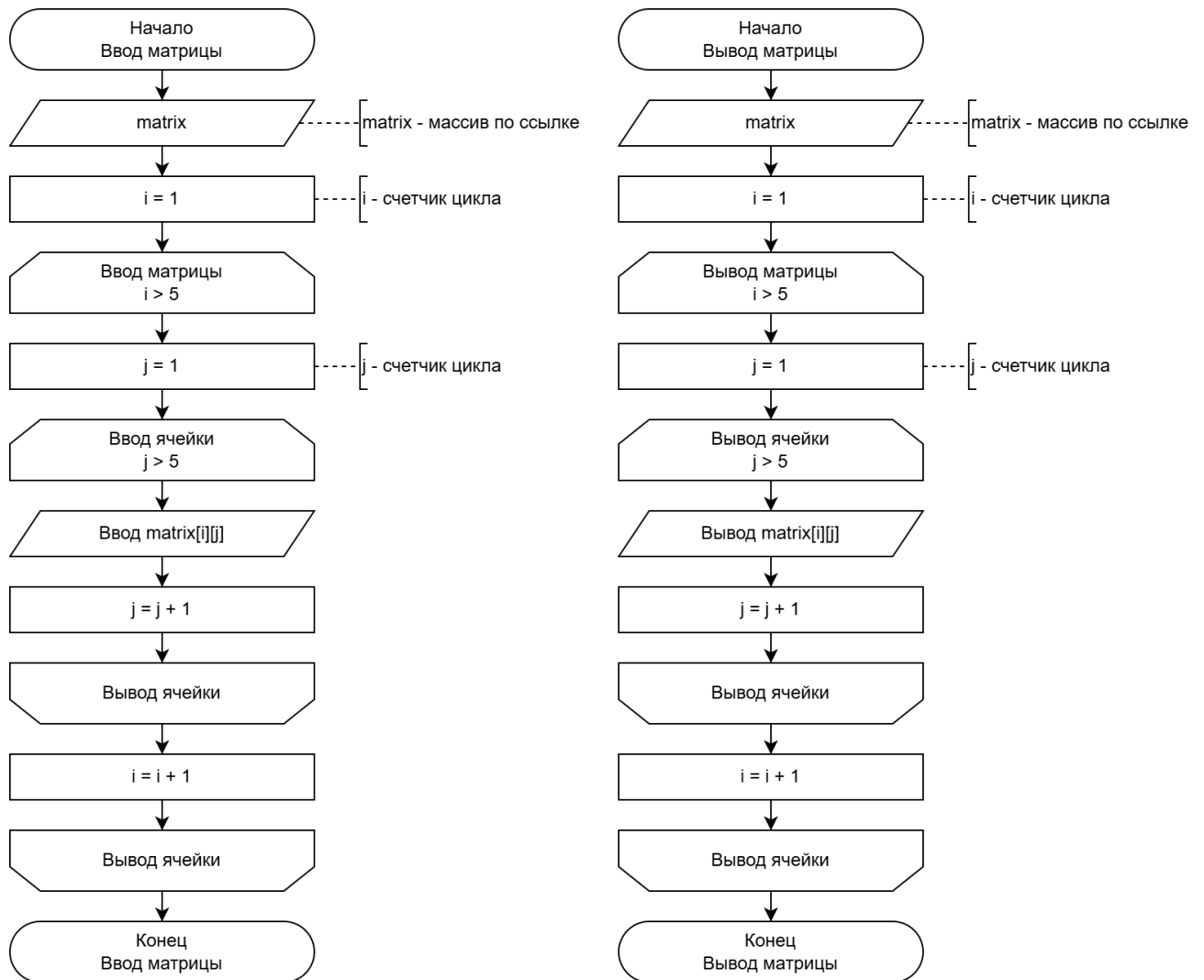


Рисунок 2 – Схемы алгоритмов ввода и вывода матрицы

На рисунке 3 представлены схемы алгоритмов подпрограмм вывода пути и обхода в глубину.

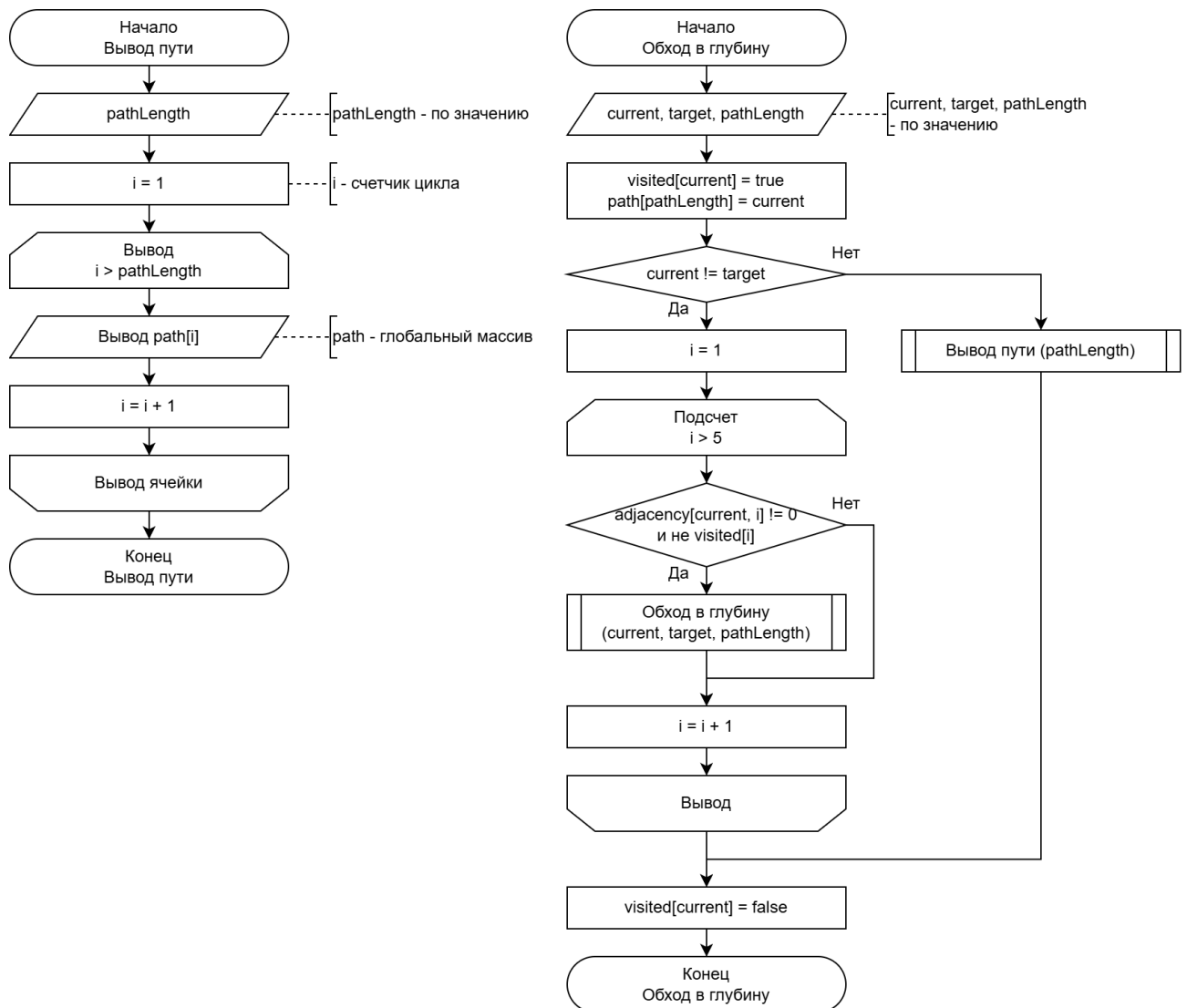


Рисунок 3 – Схемы алгоритмов вывода пути и обхода в глубину

На рисунке 4 представлены схема алгоритма подпрограммы поиска пути и основной программы.

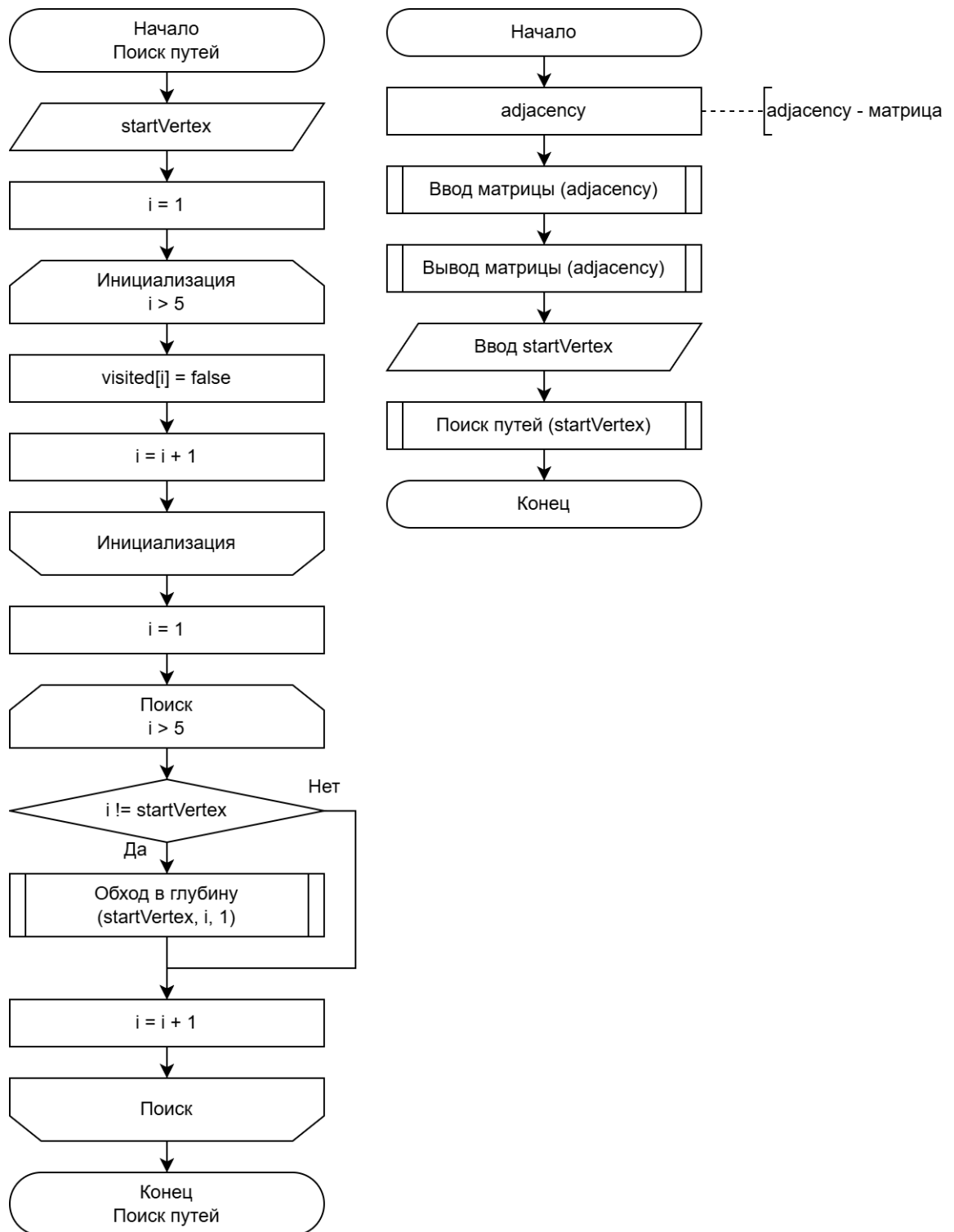


Рисунок 4 – Схемы алгоритмов поиска пути и основной программы

При разработке реализована программа, исходный код которой представлен ниже.

```
{ $codepage UTF8 }

uses
  SysUtils;

const
  MAX_SIZE = 5;

type
  TMatrix = array[1..MAX_SIZE, 1..MAX_SIZE] of integer;
  TVisited = array[1..MAX_SIZE] of boolean;
  TPath = array[1..MAX_SIZE] of integer;

var
  adjacency: TMatrix;
  visited: TVisited;
  path: TPath;
  startVertex: integer;

procedure ReadMatrix(var matrix: TMatrix);
var
  fileInput: text;
  fileLine: string;
  i, j, k: integer;
begin
  Assign(fileInput, 'input.txt');
  Reset(fileInput);

  i := 1;
  while not Eof(fileInput) do
    begin
      Readln(fileInput, fileLine);

      j := 1;
      for k := 1 to Length(fileLine) do
        begin
          if fileLine[k] <> ' ' then
            begin
              matrix[i, j] := StrToInt(fileLine[k]);
              j := j + 1;
            end
          else
            continue;
        end
      end
      i := i + 1;
    end
  end;
```

```

        end;
    end;
    i := i + 1;
end;

Close(fileInput);
end;

procedure PrintMatrix(var matrix: TMatrix);
var
    i, j: integer;
begin
    Write(' ');
    for i := 1 to MAX_SIZE do
        Write(' ', i, ' ');
        Writeln();

        for i := 1 to MAX_SIZE do
            begin
                Write(i, ' ');
                for j := 1 to MAX_SIZE do
                    Write(matrix[i, j], ' ');
                    Writeln();
                end;
            end;
        end;
    end;

procedure PrintPath(pathLength: integer);
var
    i: integer;
begin
    for i := 1 to pathLength - 1 do
        Write(path[i], ' -> ');
        Writeln(path[pathLength]);
    end;

procedure DFS(current, target, pathLength: integer);
var
    i: integer;
begin
    visited[current] := true;
    path[pathLength] := current;

    if current <> target then

```

```

begin
  for i := 1 to MAX_SIZE do
    if (adjacency[current, i] <> 0) and not visited[i] then
      DFS(i, target, pathLength + 1);
    end
  else
    PrintPath(pathLength);

    visited[current] := false;
end;

procedure FindAllPaths(startVertex: integer);
var
  i: integer;
begin
  for i := 1 to MAX_SIZE do
    visited[i] := false;

    for i := 1 to MAX_SIZE do
      if i <> startVertex then
        DFS(startVertex, i, 1);
      end;
    end;

begin

  ReadMatrix(adjacency);
  Writeln(#10, 'Матрица смежности', #10);
  PrintMatrix(adjacency);

  Write(#10, 'Введите начальную вершину: ');
  Readln(startVertex);

  Writeln('Список всех элементарных путей');
  FindAllPaths(startVertex);

  Readln;

end.

```



Экранная форма программы в виде консольного приложения представлена на рисунке 5.

```
Матрица смежности

  1  2  3  4  5
1  0  1  1  0  0
2  0  0  0  1  0
3  0  0  0  0  0
4  1  0  0  0  0
5  1  0  0  0  0

Введите начальную вершину: 1
Список всех элементарных путей
1 -> 2
1 -> 3
1 -> 2 -> 4
```

Рисунок 5 – Консольный интерфейс программы

## Вывод

В процессе выполнения лабораторной работы, при решении предложенных задач, реализована программа на языке Паскаль, которая находит возможные элементарные пути из вершины, согласно матрице смежности, используя обход в глубину.