Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №7**

«Реализация алгоритмов на графах, представленных с помощью

матриц»

Выполнил:

студент группы 213 Файтельсон А.А.

Проверил:

ассистент кафедры ПОиАИС Овсянников А.В.

Курск, 2025

**Цель работы:** Изучить основные алгоритмы теории графов

**Словесная постановка задачи:**

Определить, является ли граф связным. Граф задан матрицей смежности.

**Алгоритм решения задачи в текстуальном виде:**

Граф является связным, если для любых двух вершин существует путь между ними. Для проверки связности можно использовать алгоритм поиска в ширину (BFS) или поиск в глубину (DFS). Если из одной вершины можно достичь все остальные вершины, то граф связный. Если граф ориентированный, необходимо учитывать достижимость в обоих направлениях.

Создаем функцию isConnected, которая принимает матрицу смежности графа. Используем BFS для проверки достижимости всех вершин из первой вершины. Если хотя бы одна вершина недостижима, граф несвязный.

**Обоснование правильности выбора алгоритма**

Алгоритм BFS выбран потому, что он эффективно проверяет достижимость всех вершин из начальной вершины. Этот метод гарантирует, что каждая вершина будет посещена ровно один раз, что обеспечивает оптимальное использование ресурсов. BFS также подходит для работы с матрицей смежности, так как легко проверяет наличие ребер между вершинами.

**Листинг программы**

**package main**

**import (**

**"fmt"**

**)**

**// Функция для проверки связности графа**

**func isConnected(adjMatrix [][]int) bool {**

**n := len(adjMatrix) // Количество вершин**

**visited := make([]bool, n)**

**// BFS для проверки достижимости всех вершин**

**queue := []int{0} // Начинаем с первой вершины**

**visited[0] = true**

**for len(queue) > 0 {**

**current := queue[0]**

**queue = queue[1:]**

**for i := 0; i < n; i++ {**

**if adjMatrix[current][i] == 1 && !visited[i] {**

**visited[i] = true**

**queue = append(queue, i)**

**}**

**}**

**}**

**// Проверяем, все ли вершины были посещены**

**for \_, v := range visited {**

**if !v {**

**return false**

**}**

**}**

**return true**

**}**

**func main() {**

**// Пример матрицы смежности для неориентированного графа**

**adjMatrix := [][]int{**

**{0, 1, 0, 0},**

**{1, 0, 1, 1},**

**{0, 1, 0, 0},**

**{0, 1, 0, 0},**

**}**

**if isConnected(adjMatrix) {**

**fmt.Println("Граф связный")**

**} else {**

**fmt.Println("Граф несвязный")**

**}**

**}**