Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №6

по курсу «Логика и основы алгоритмизации

в инженерных задачах»

на тему «Унарные и бинарные операции над графами»

Выполнил: ст. гр. 24ВВВ1

Будников А.С.

Принял:

к.т.н, доцент Юрова О. В.

к.т.н. Деев М.В.

Пенза 2025

**Цель работы**: реализовать унарные и бинарные операции над неориентированными графами, представленными в виде матриц смежности и списков смежности.

**Лабораторное задание:**

**Задание 1:**

1. Сгенерировать две матрицы смежности неориентированных помеченных графов G₁, G₂
2. \*Преобразовать матрицы смежности в списки смежности

**Задание 2:**

1. Для матричной формы представления выполнить операции:  
   а) отождествления вершин  
   б) стягивания ребра  
   в) расщепления вершины
2. \*Выполнить те же операции для представления в виде списков смежности

**Задание 3:**

Для матричной формы представления выполнить операции:  
а) объединения G = G₁ ∪ G₂  
б) пересечения G = G₁ ∩ G₂  
в) кольцевой суммы G = G₁ ⊕ G₂

**Задание 4:**  
\*Выполнить операцию декартова произведения графов G = G₁ × G₂

**Теоретическая часть:**

Унарные операции:

* Отождествление вершин - операция слияния двух вершин в одну с объединением их окружений
* Стягивание ребра - частный случай отождествления смежных вершин
* Расщепление вершины - операция разделения вершины на две с распределением инцидентных ребер

Бинарные операции:

* Объединение - граф, содержащий все вершины и ребра исходных графов
* Пересечение - граф, содержащий только общие вершины и ребра исходных графов
* Кольцевая сумма - граф, содержащий ребра, присутствующие только в одном из исходных графов
* Декартово произведение - граф, вершины которого являются упорядоченными парами вершин исходных графов

**Практическая часть:**

**Задание 1:**

1. Для генерации графов реализована функция *create\_graph*, использующая линейный конгруэнтный генератор псевдослучайных чисел. Функция принимает параметры: количество вершин n, вероятность ребра *edgeProb*, вероятность петли *loopProb* и начальное значение генератора *seed*. Алгоритм последовательно обходит возможные пары вершин и на основе вероятностных параметров определяет наличие ребер и петель, одновременно заполняя матрицу смежности и список смежности.
2. Преобразование в списки смежности выполняется автоматически в процессе генерации графа. Функция *print\_list* обеспечивает наглядный вывод списков смежности в формате "вершина: соседние\_вершины".

**Задание 2:**

1. Отождествление вершин реализовано в функции *identify\_vertices*. Алгоритм включает:
   * Объединение окружений вершин u и v
   * Удаление вершины с большим номером
   * Корректную обработку петель и ребер между объединяемыми вершинами
   * Обновление номеров вершин во всех списках смежности
2. Стягивание ребра реализовано в функции *contract\_edge*. Алгоритм аналогичен отождествлению, но включает дополнительную проверку наличия ребра между вершинами и специальную обработку для исключения создания петли из стягиваемого ребра.
3. Расщепление вершины реализовано в функции *split\_vertex*. Алгоритм включает:
   * Добавление новой вершины в граф
   * Создание ребра между исходной и новой вершиной
   * Перераспределение инцидентных ребер согласно заданному списку
   * Корректную обработку петель при расщеплении

**Задание 3:**

1. Объединение графов реализовано в функции *graph\_union*. Алгоритм создает граф с максимальным количеством вершин из исходных графов. Для общих вершин ребро присутствует, если оно есть хотя бы в одном из исходных графов.
2. Пересечение графов реализовано в функции *graph\_intersection.* Алгоритм создает граф с минимальным количеством вершин. Ребро присутствует только если оно есть в обоих исходных графах.
3. Кольцевая сумма реализована в функции *ring\_sum*. Алгоритм использует операцию *XOR* для матриц смежности. Дополнительно выполняется удаление изолированных вершин, включая вершин только с петлями.

**Задание 4:**

Декартово произведение реализовано в функции graph\_cartesian\_product. Алгоритм:

* Создает граф с количеством вершин |V₁| × |V₂|
* Соединяет вершины (u₁,v₁) и (u₂,v₂) если:
  + u₁ = u₂ и v₁ смежна с v₂ в G₂, ИЛИ
  + v₁ = v₂ и u₁ смежна с u₂ в G₁
* Обеспечивает симметричность матрицы смежности для неориентированного графа

**Вывод:**

В результате выполнения работы была успешно разработана и реализована комплексная система для выполнения унарных и бинарных операций над неориентированными графами. Система поддерживает два представления графов - матрицы смежности и списки смежности, что обеспечивает гибкость в использовании. Все алгоритмы демонстрируют корректную работу, включая обработку краевых случаев таких как петли, изолированные вершины и графы разного размера. Особое внимание уделено эффективному преобразованию между различными представлениями графов и согласованности результатов операций. Практическая значимость работы подтверждена полной функциональной готовностью всех алгоритмов и их взаимной согласованностью.