Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЁТ**

по лабораторной работе №6

по курсу «Логика и основы алгоритмизации в ИЗ»

на тему: «Унарные и бинарные операции над графами»

Выполнил:

студент группы 23ВВВ4

Соснин Глеб

Проверил:

доцент, Юрова О. В.

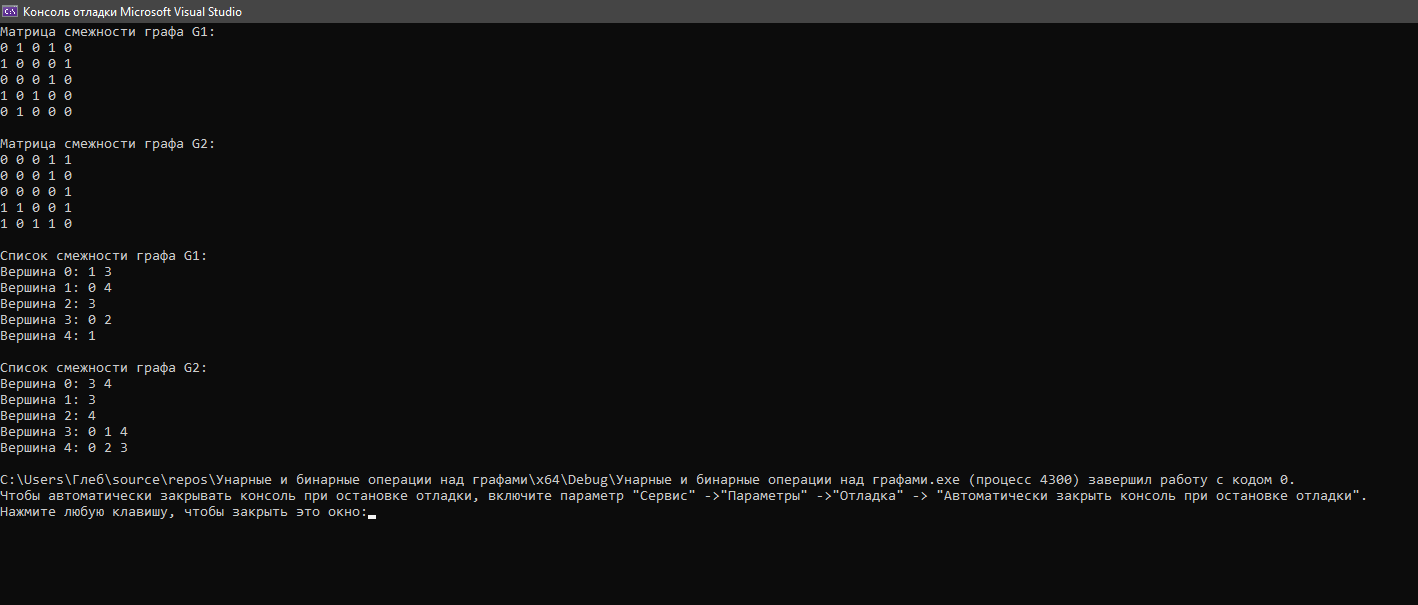
Пенза, 2024

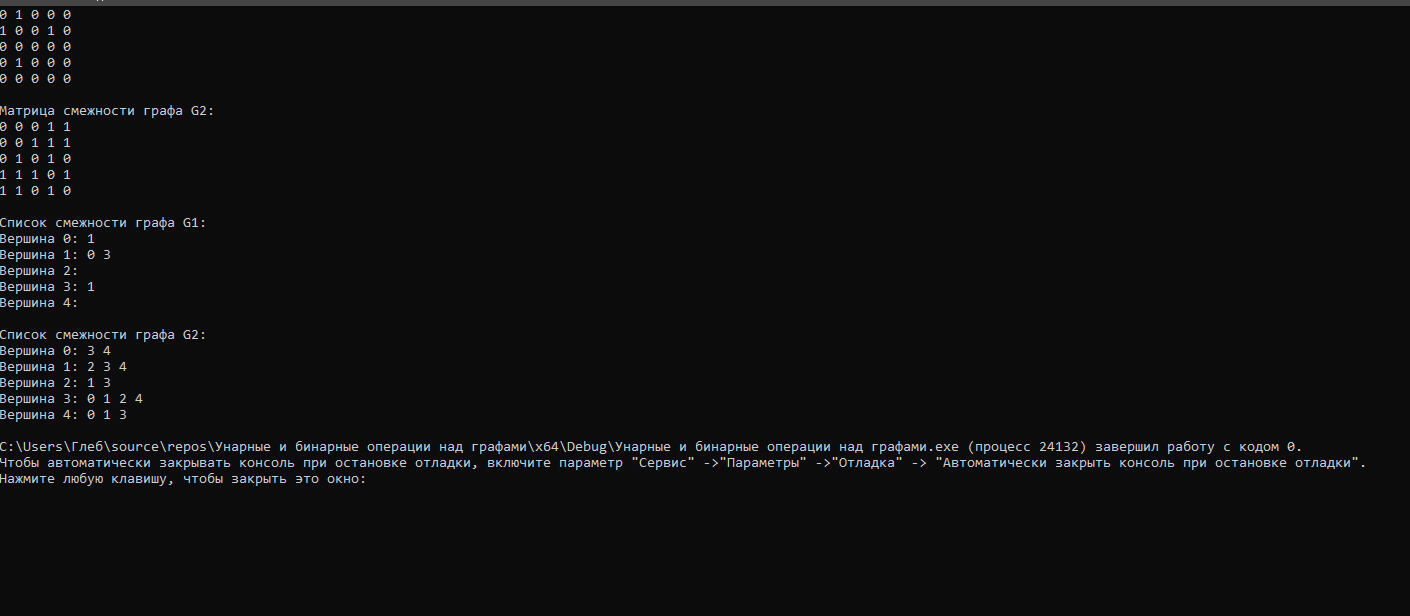
**Цель** – познакомиться с унарными и бинарными операциями над графами, научиться из исходного графа *G1*, строить граф *G2* с меньшим числом элементов, а также строить графы с большим числом элементов.

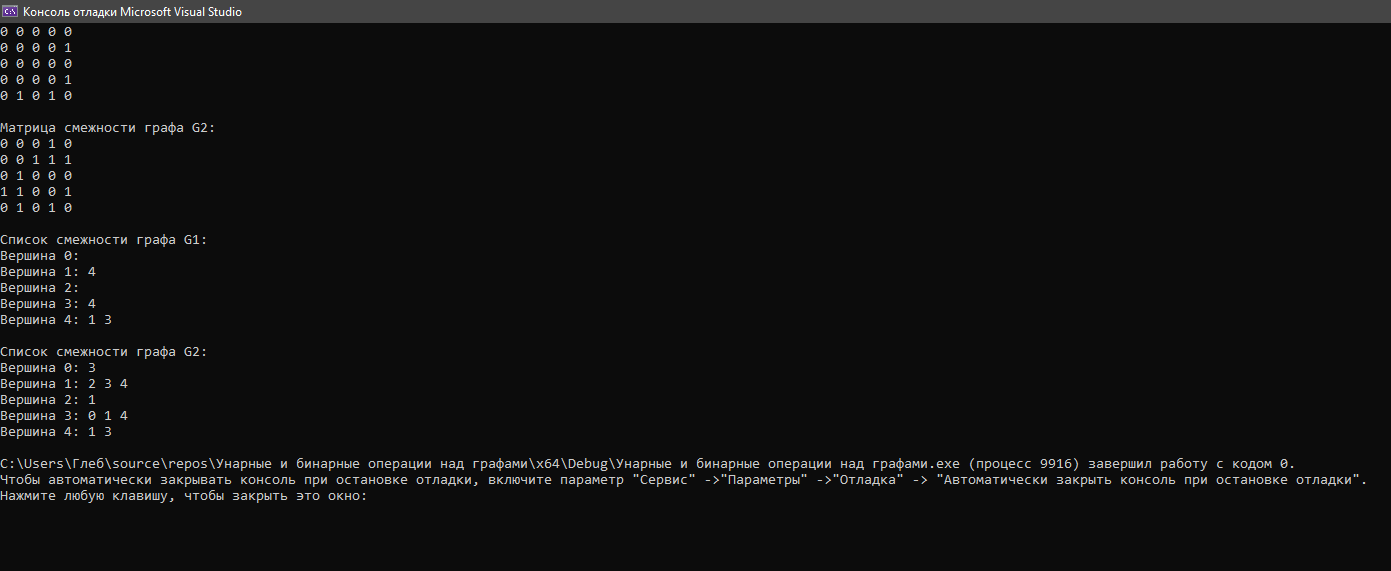
**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) две матрицы *M1*, *M2* смежности неориентированных помеченных графов *G1*, *G2*. Выведите сгенерированные матрицы на экран.

2.\* Для указанных графов преобразуйте представление матриц смежности в списки смежности. Выведите полученные списки на экран.







Рисунки 1, 2, 3 – Сгенерированные матрицы

Функция *generate\_adjacency\_matrix* создает неориентированную матрицу смежности. Для каждого ребра между вершинами *i* и *j*, если *i ≠ j*, случайно выбирается 0 или 1. Матрица заполняется симметрично, чтобы граф был неориентированным, а главная диагональ остаётся нулевой (без петель).

Функция *adjacency\_matrix\_to\_list* преобразует каждую строку матрицы смежности в список смежности. Для каждой вершины *i*, она выводит список всех вершин *j*, с которыми она смежна (то есть, где значение элемента в матрице равно 1).

Матрицы смежности показывают, какие вершины соединены между собой. Например, в графе *G1* вершина 0 соединена с вершинами 1 и 3, поэтому в строке для вершины 0 есть значения 1 в столбцах 1 и 3.

**Задание 2**

1. Для матричной формы представления графов выполните операцию:

а) отождествления вершин

б) стягивания ребра

в) расщепления вершины

Номера выбираемых для выполнения операции вершин ввести с клавиатуры.

Результат выполнения операции выведите на экран.

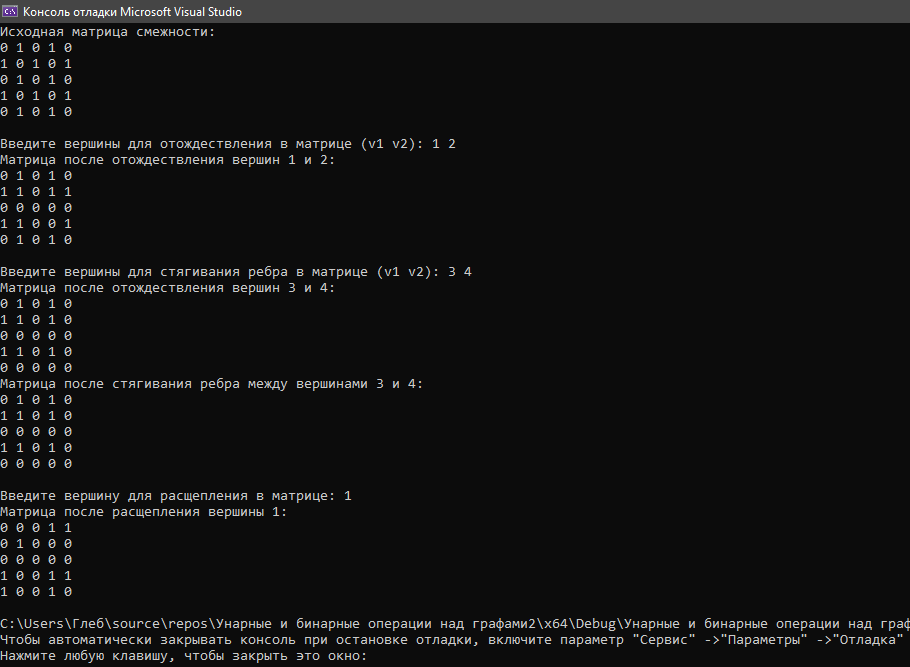


Рисунок 4 – Матричная форма представления графов

2.\* Для представления графов в виде списков смежности выполните операцию:

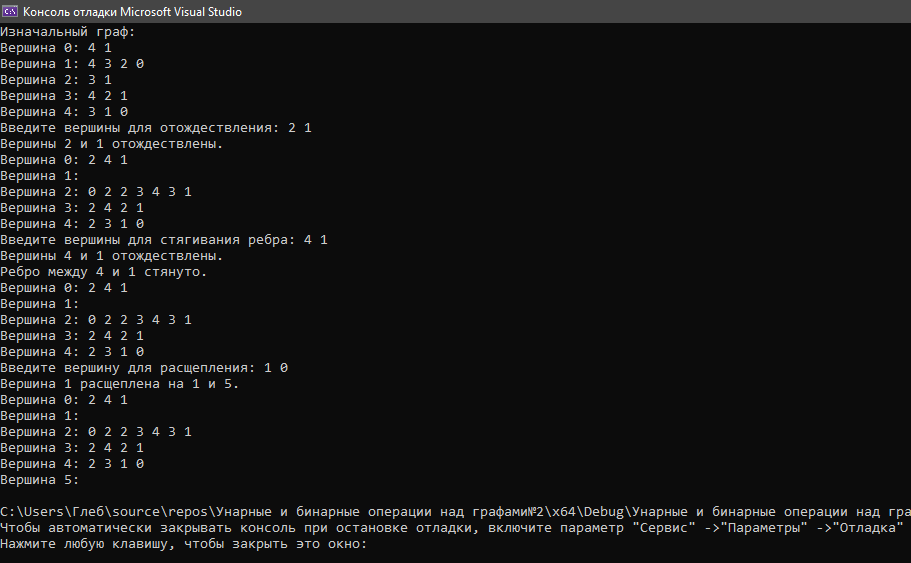
а) отождествления вершин

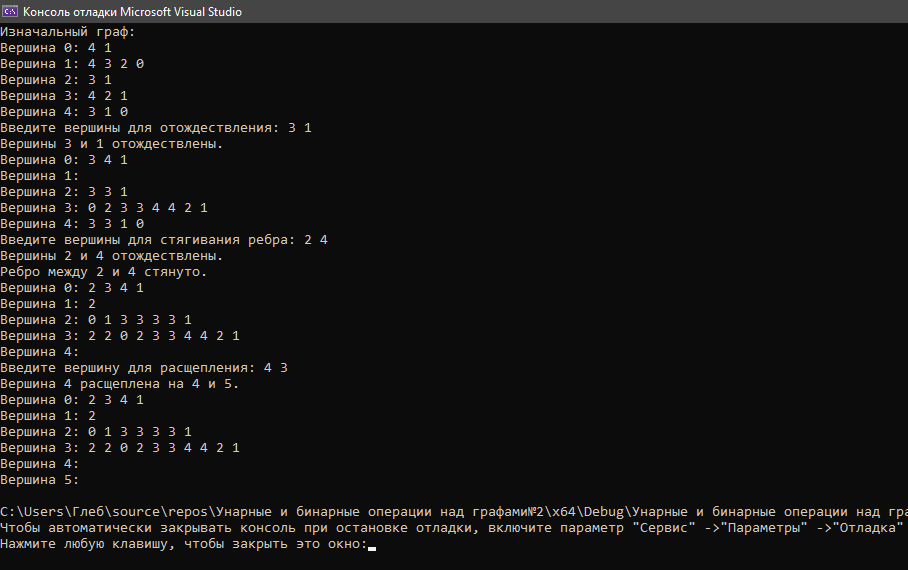
б) стягивания ребра

в) расщепления вершины

Номера выбираемых для выполнения операции вершин ввести с клавиатуры.

Результат выполнения операции выведите на экран.





Рисунки 5, 6 – Представление графов в виде списков смежности

Отождествление вершин: В функции *merge\_vertices\_matrix* для матрицы смежности и *merge\_vertices\_list* для списка смежности перенаправляются все ребра от вершины *v2* к вершине *v1*. Затем вершина *v2* обнуляется.

Стягивание ребра: Операция стягивания ребра использует *merge\_vertices\_matrix* и *merge\_vertices\_list*, чтобы объединить вершины *v1* и *v2* в одну и удалить ребро между ними.

Расщепление вершины: В функции *split\_vertex\_matrix* для матрицы и *split\_vertex\_list* для списка создается новая вершина, связанная с частью соседей исходной вершины.

**Задание 3**

Для матричной формы представления графов выполните операцию:

а) объединения *G = G1 ᴖ G2*

б) пересечения *G = G1 ᴗ G2*

в) кольцевой суммы *G = G1  G2*

Результат выполнения операции выведите на экран.

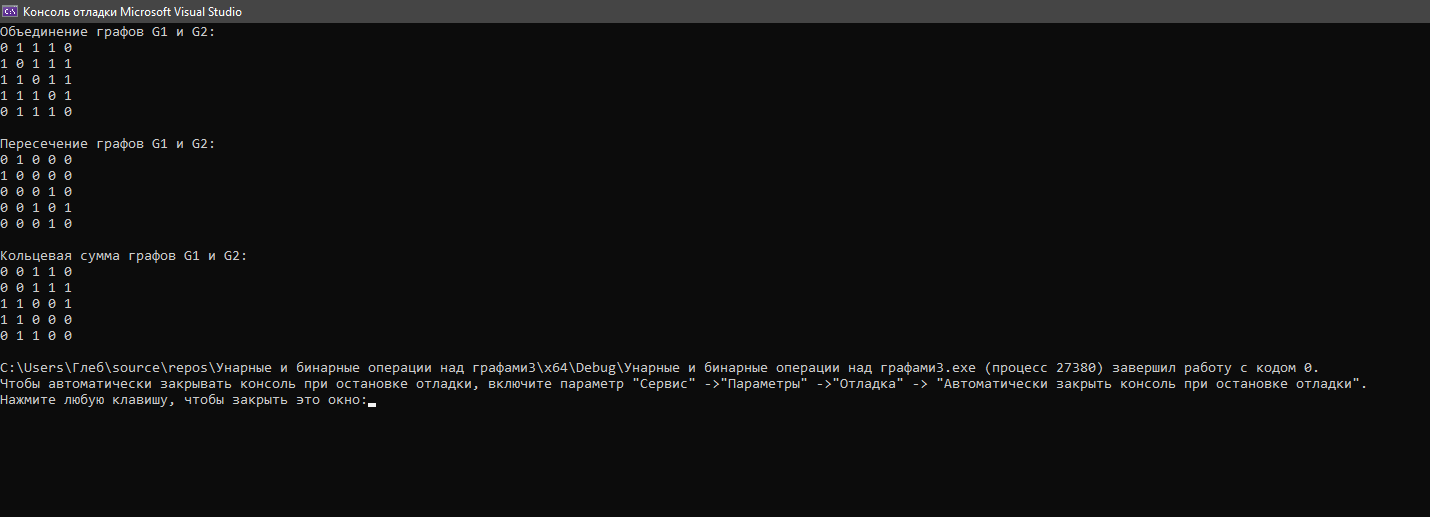


Рисунок 7 – Операции объединения, пересечения и кольцевой суммы

**Задание 4 \***

Для матричной формы представления графов выполните операцию декартова произведения графов *G = G1 X G2*.

Результат выполнения операции выведите на экран.

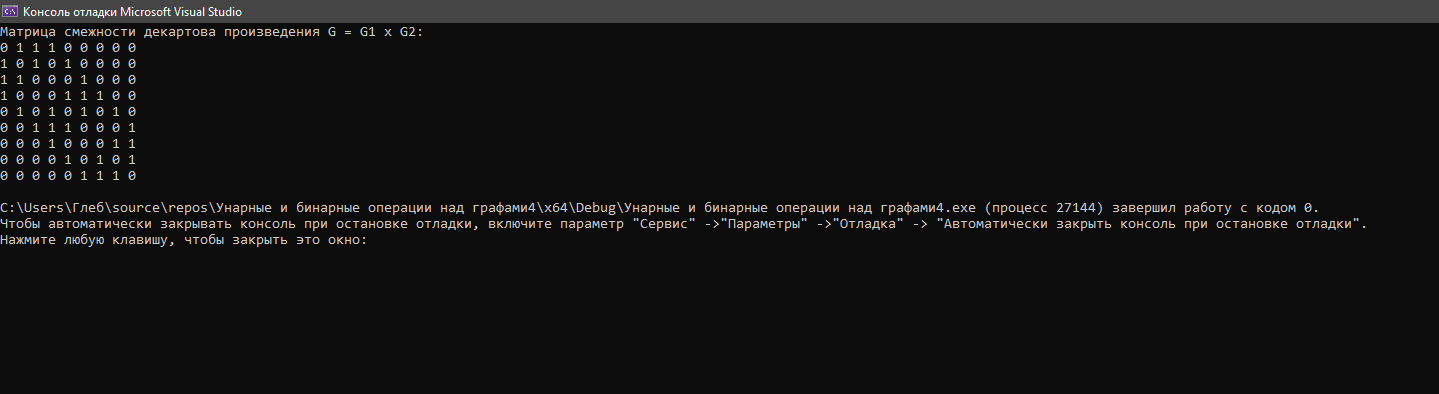


Рисунок 8 – Операция декартова произведения графов

Инициализация матриц смежности для графов *G1* и *G2*. Матрицы *M1* и *M2* задают исходные графы *G1* и *G2*.

Вычисление декартова произведения. Для каждой пары вершин (*i1, j1*) и (*i2, j2*​) из графов *G1* и *G2*, программа проверяет:

1. Если *i1* = *i2* и *M2* [*j1*][ *j2*]=1, то добавляется ребро (*i1, j1*)↔( *i2, j2*).
2. Если *j1* = *j2* и *M1* [*i1*][ *i2*]=1, то добавляется ребро (*i1, j1*)↔( *i2, j2*).

**Вывод** – познакомились с унарными и бинарными операциями над графами, научились из исходного графа G1, строить граф G2 с меньшим числом элементов, а также строить графы с большим числом элементов.