Задание 8

Перечисление контуров ориентированного графа методом латинской композиции.

Язык программирования – С++

Теория:

Исходя из предложенного алгоритма для решения задачи под понятием контура ориентированного графа будет подразумеваться Гамильтонов цикл, так как метод латинской композиции – метод, позволяющий легко найти все пути и циклы Гамильтона в графе.

**Гамильтоновым циклом** в графе называют цикл, проходящий через все вершины. Также можно определить понятие гамильтонова пути — незамкнутого пути, также проходящего через все вершины.

Описание метода латинской композиции:

Шаг 1:

Считав граф в таблицу смежности составляем матрицу по правилу:

Если в графе есть ребро вершины i к вершине j, а также I отличается от j, то в ячейке ij матрицы M печатаем ij, во всех остальных случаях элемент матрицы M будет равен нулю.

Шаг 2:

Введем операцию умножения матриц: пусть при умножении матриц A и B, подобных матрицe M мы получаем матрицу C такую, что каждый элемент состоит из списка, полученного путем объединения каждой строки с каждой, при этом опускается первый символ каждой строки, входящей во вторую матрицу.

Шаг 3:

Умножаем матрицу M на саму себя, получая матрицу .

Шаг 4:

Повторяя шаг 3 (n – 1) раз, где n = количеству вершин в графе получаем матрицу , где первая ячейка каждой строки содержит в себе все возможные гамильтоновы циклы в графе.

Таким образом, придерживаясь вышеприведенного алгоритма получаем функцию для решения данной задачи.

Описание программы:

Для решения поставленной задачи было написано вычислительное ядро – набор функций, объектов и методов, необходимых и достаточных для выполнения задания. Ядро представляет собой программу, принимающую на вход файл с таблицей смежности графа и возвращающую файл, содержащий все контуры данного графа.

Сложность:

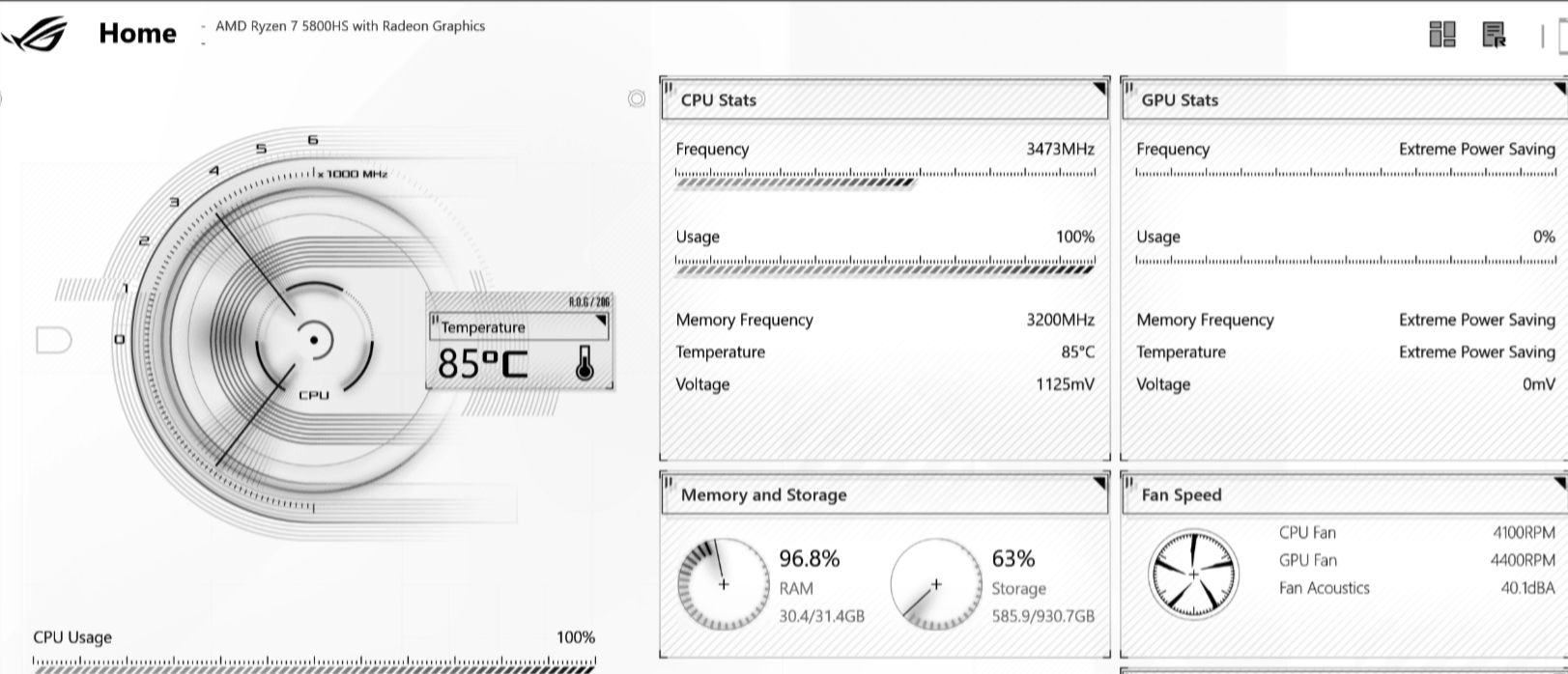
Так как временная сложность алгоритма в худшем случае составляет

О(N!), где N – количество вершин в графе было решено использовать многопоточность и язык С++. Благодаря многопоточности удалось улучшить результат в ~10 раз ((для процессора с 8 ядрами и 16 потоками) в зависимости от количества логических ядер процессора результат может меняться как в худшую, так и в лучшую сторону). Алгоритм метода латинской композиции подразумевает полный перебор всех возможных контуров в графе, поэтому перебор будет занимать N! времени.

Нецелесообразность использования графического интерфейса:

Программа генерирует большое количество выходных данных, измеряемых в гигабайтах. Для полного графа с 10 вершинами был создан выходной файл, объем которого превышает 1 Гбайт. Всего было найдено более 300 тысяч контуров, отображение и отрисовка которых невозможна. Входные данные задаются файлом, который генерируется случайным образом (от 5 до 15 вершин, от 0 до 225 ребер). Отталкиваясь от всех вышеизложенных факторов, я считаю нецелесообразным создание графического интерфейса к данному приложению. Написанное ядро отлично подойдет как база для более конкретных задач.

Также стоит отметить невероятное потребление ресурсов компьютера для нахождения всех контуров графа:

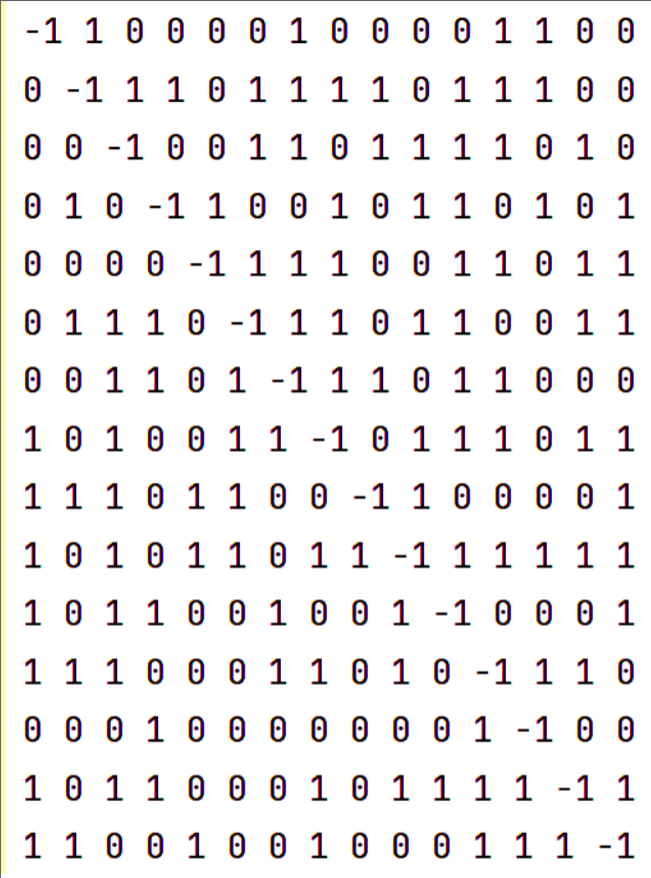


Инструкция к запуску:

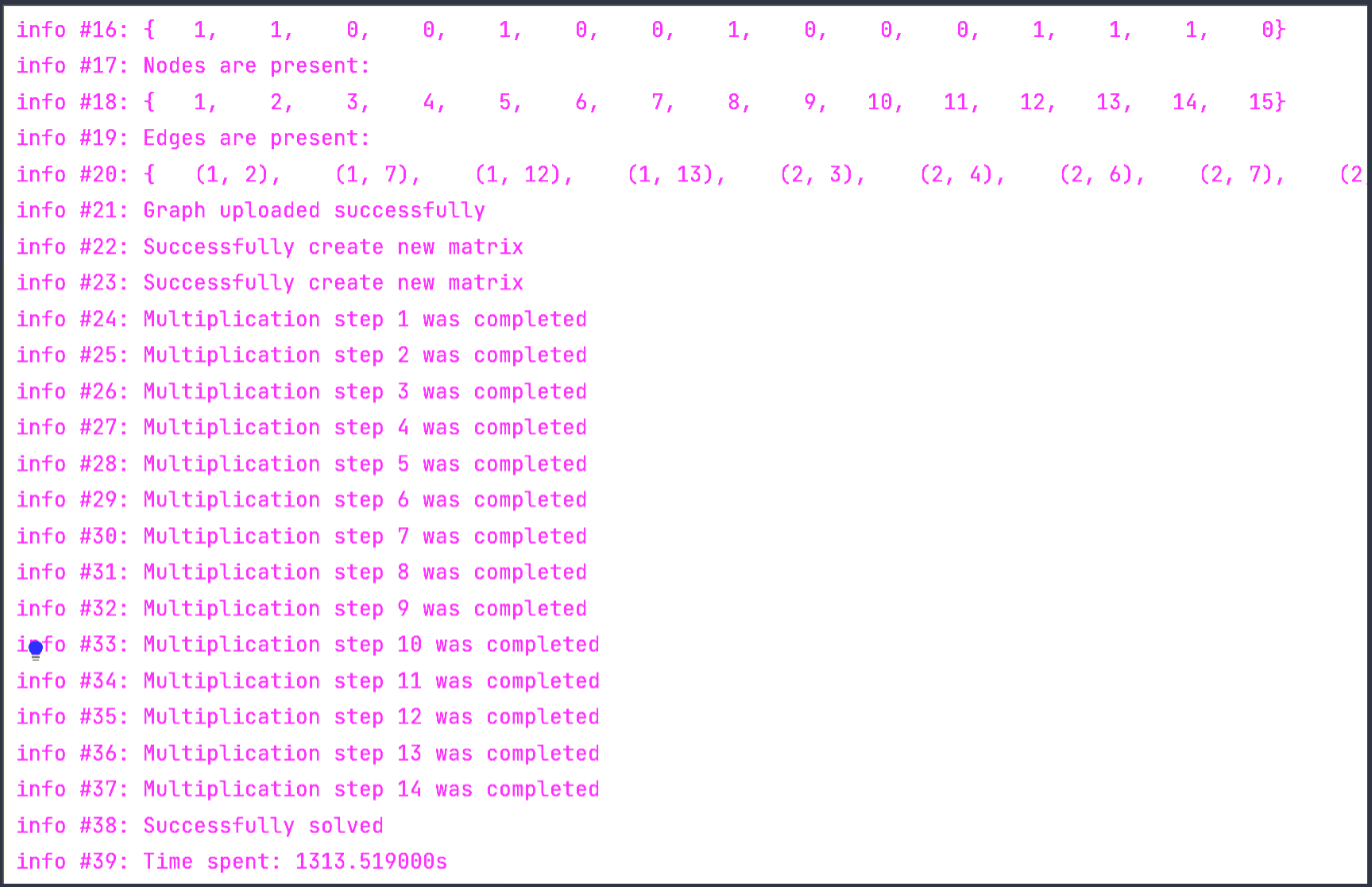
Запустить .exe файл и ждать завершения вычислений.

Пример работы программы:

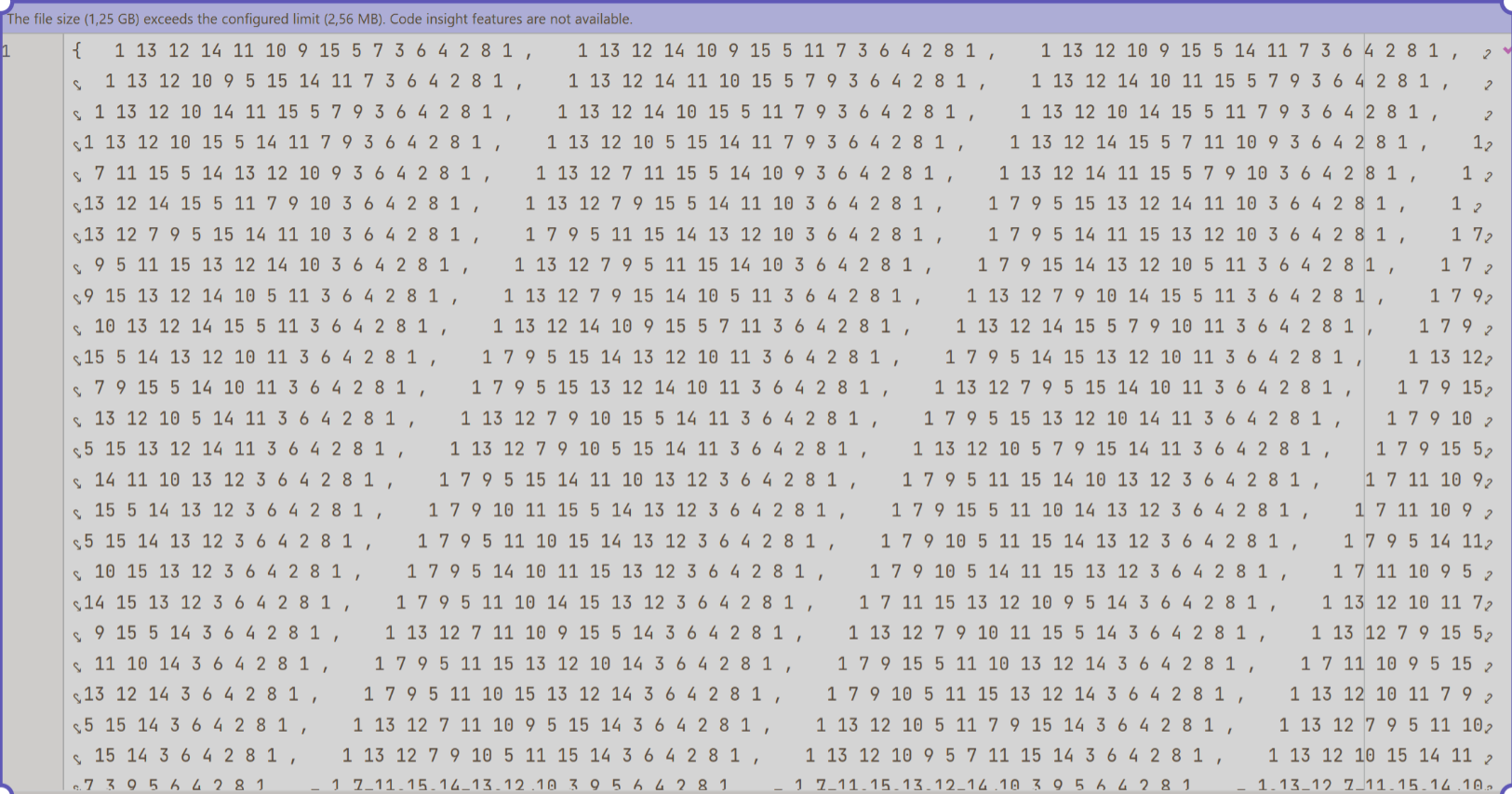
Входной файл graph.txt (таблица смежности):



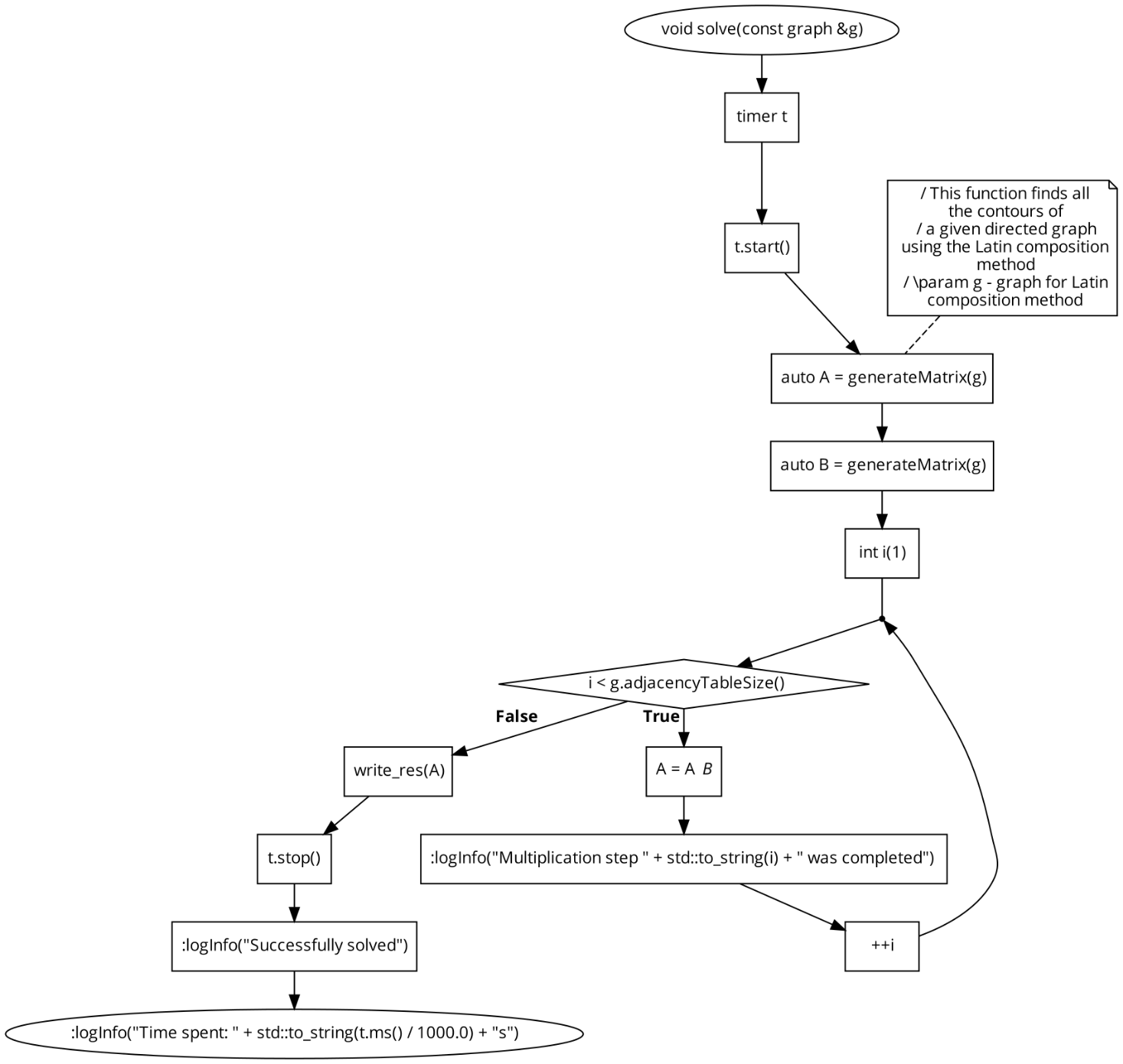
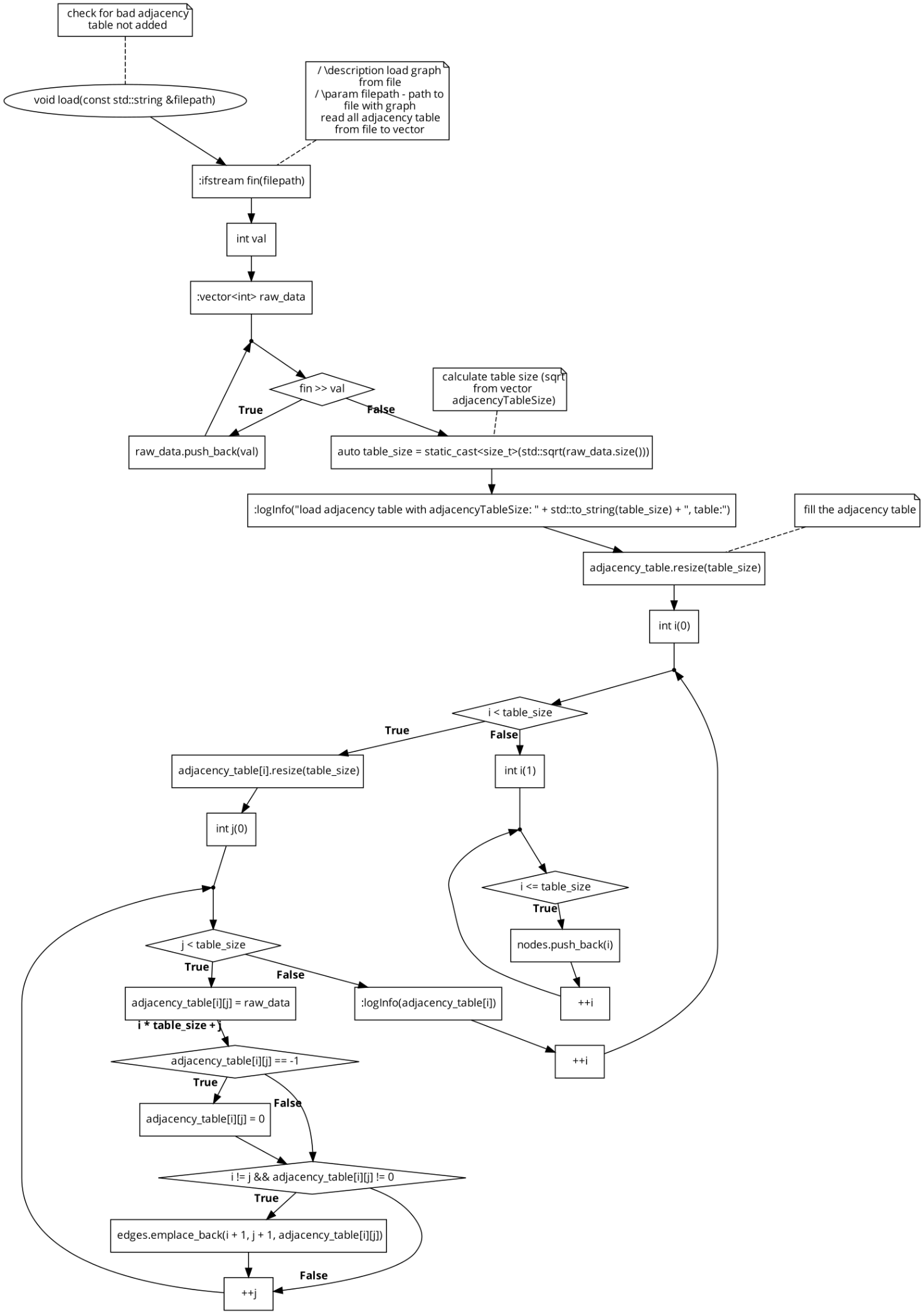
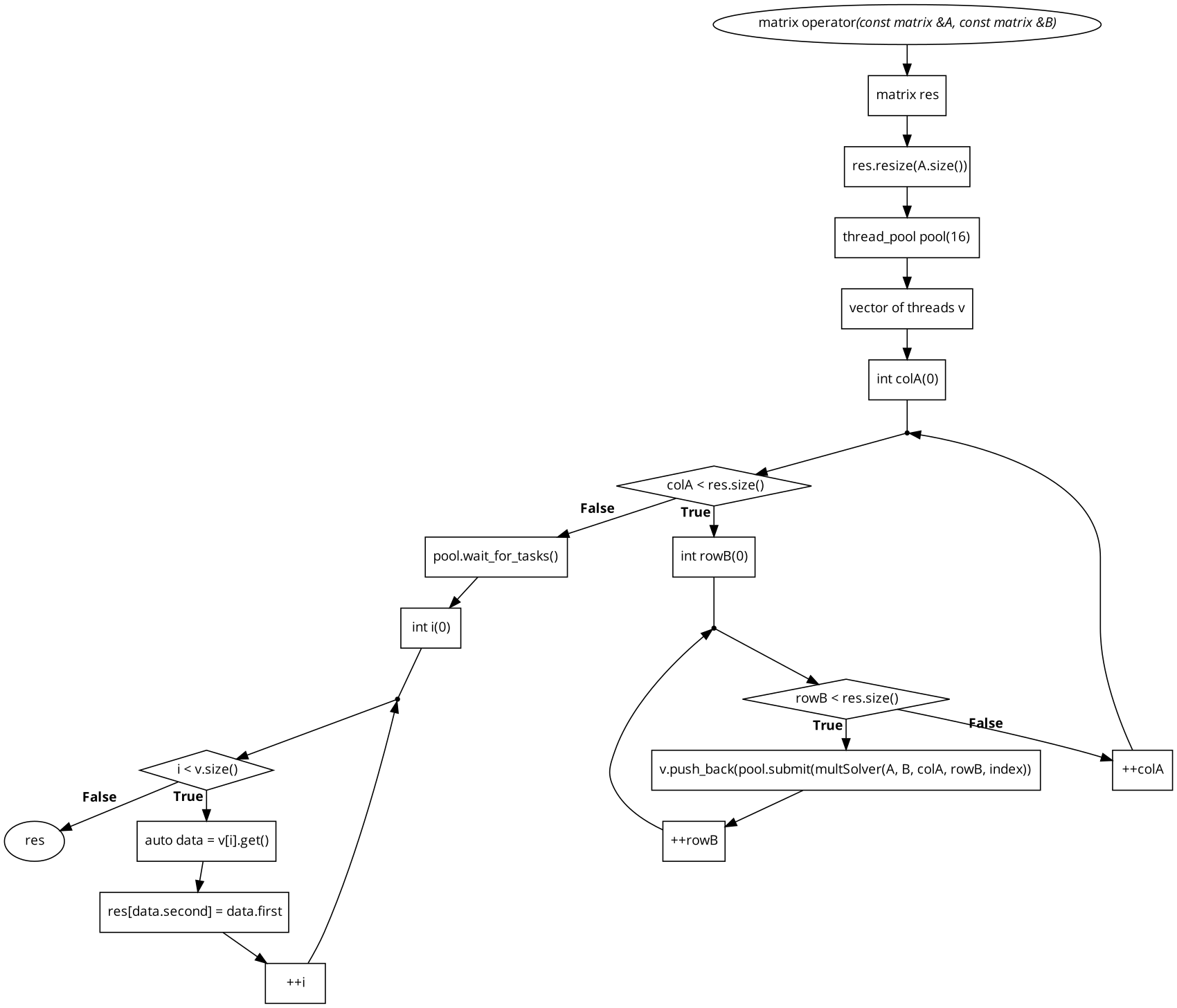
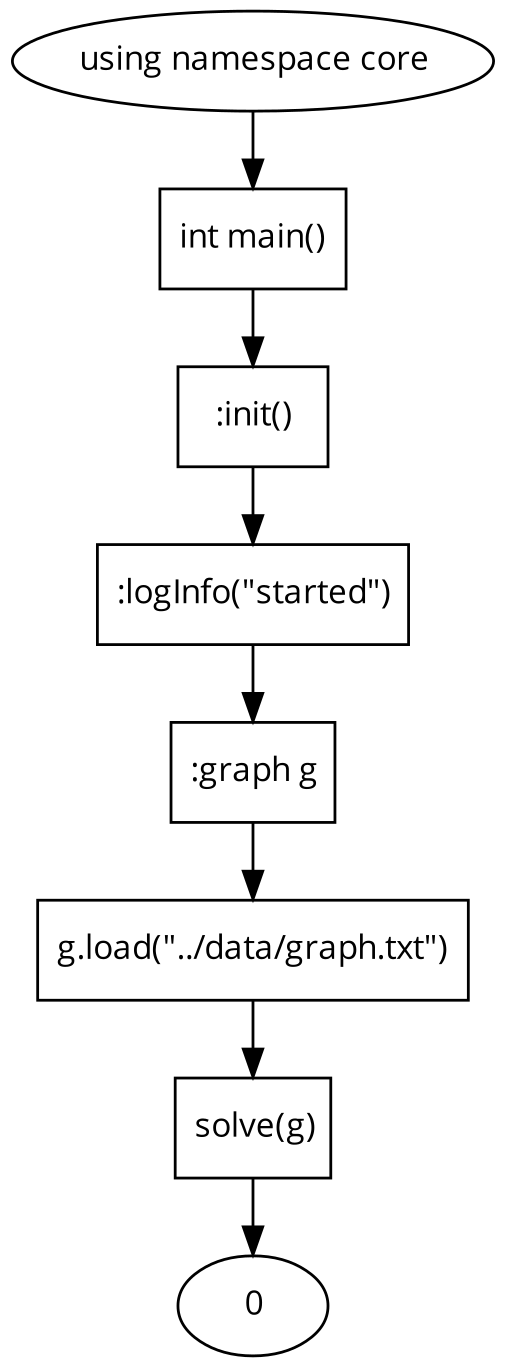
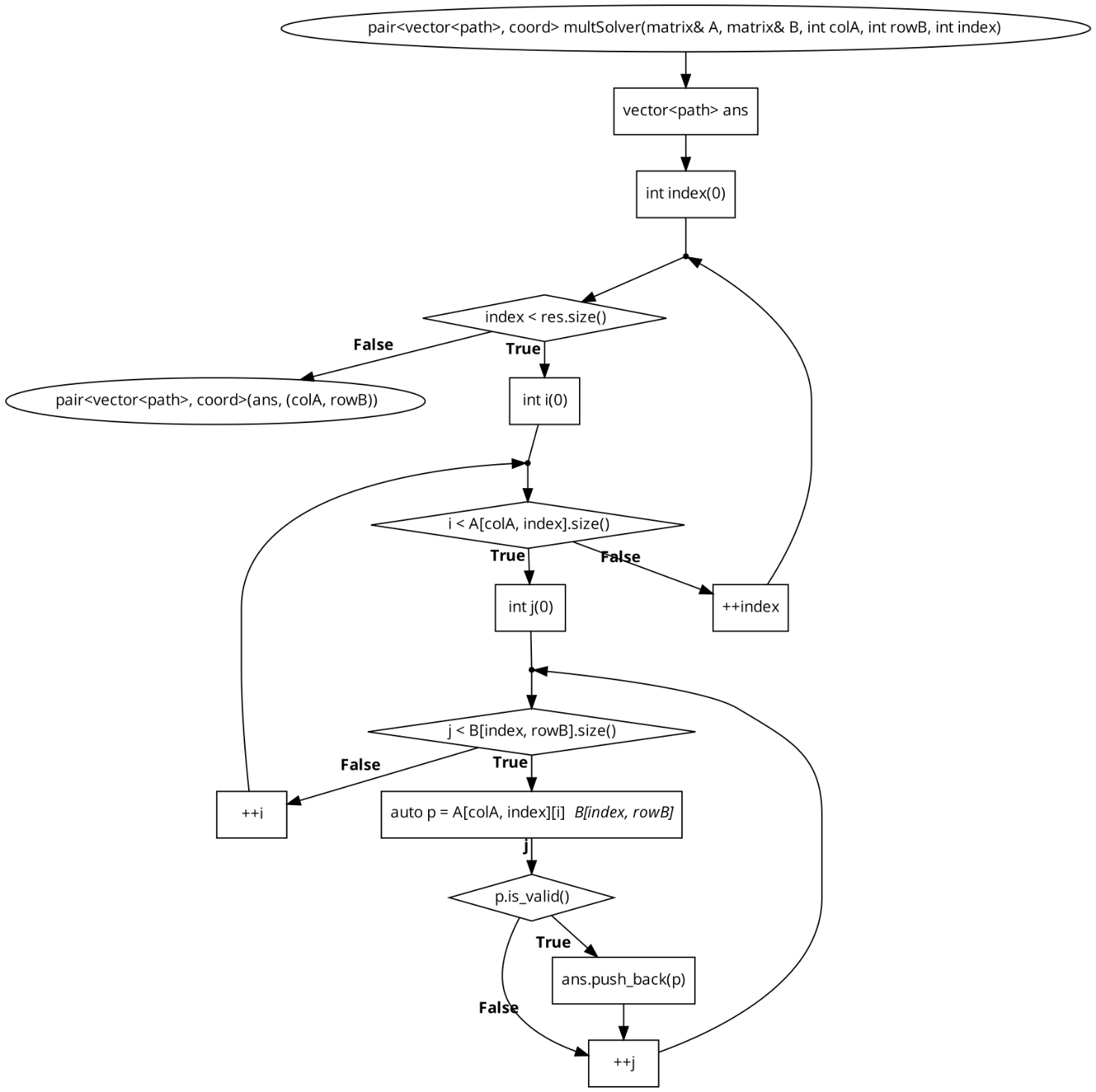
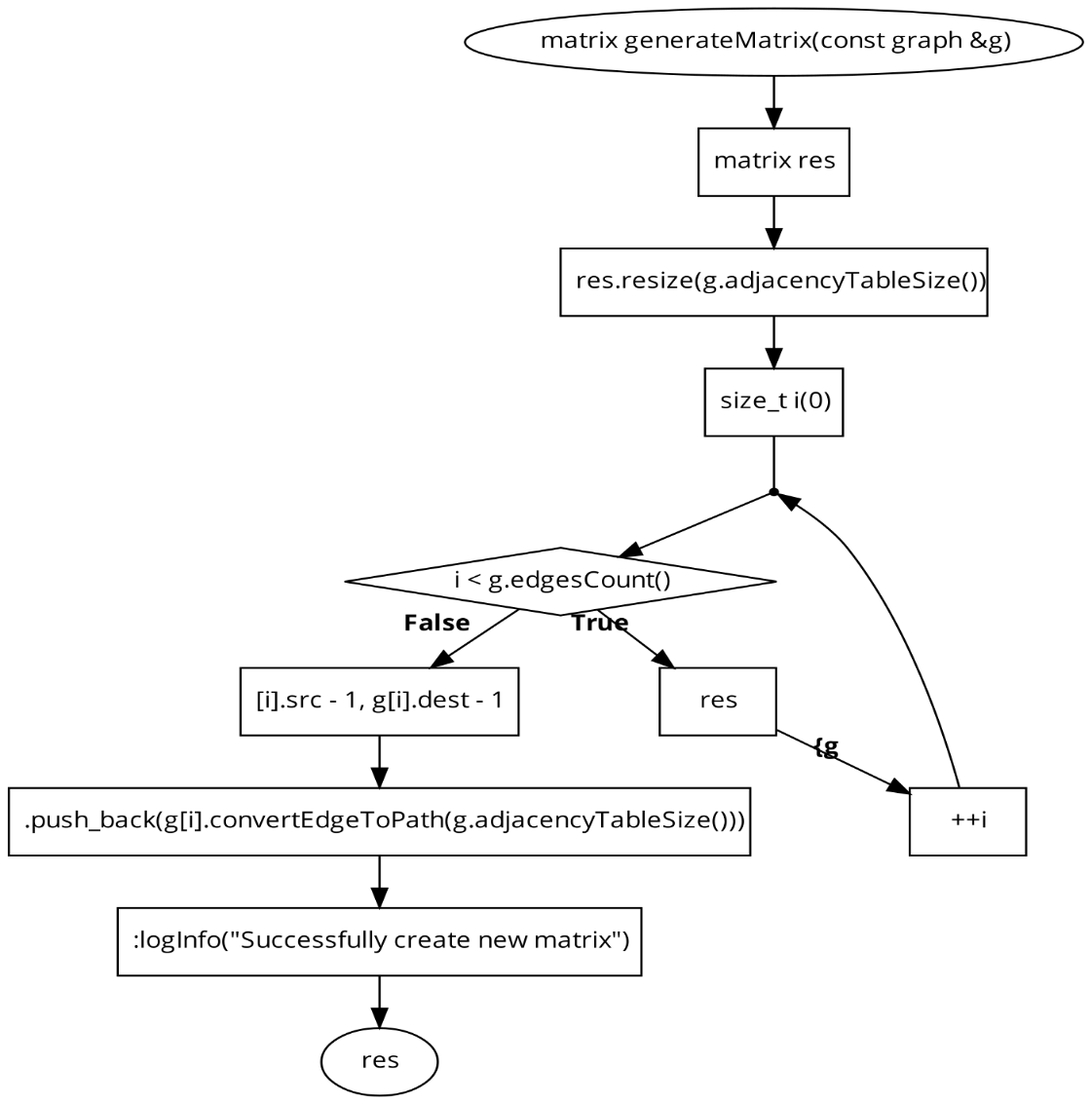
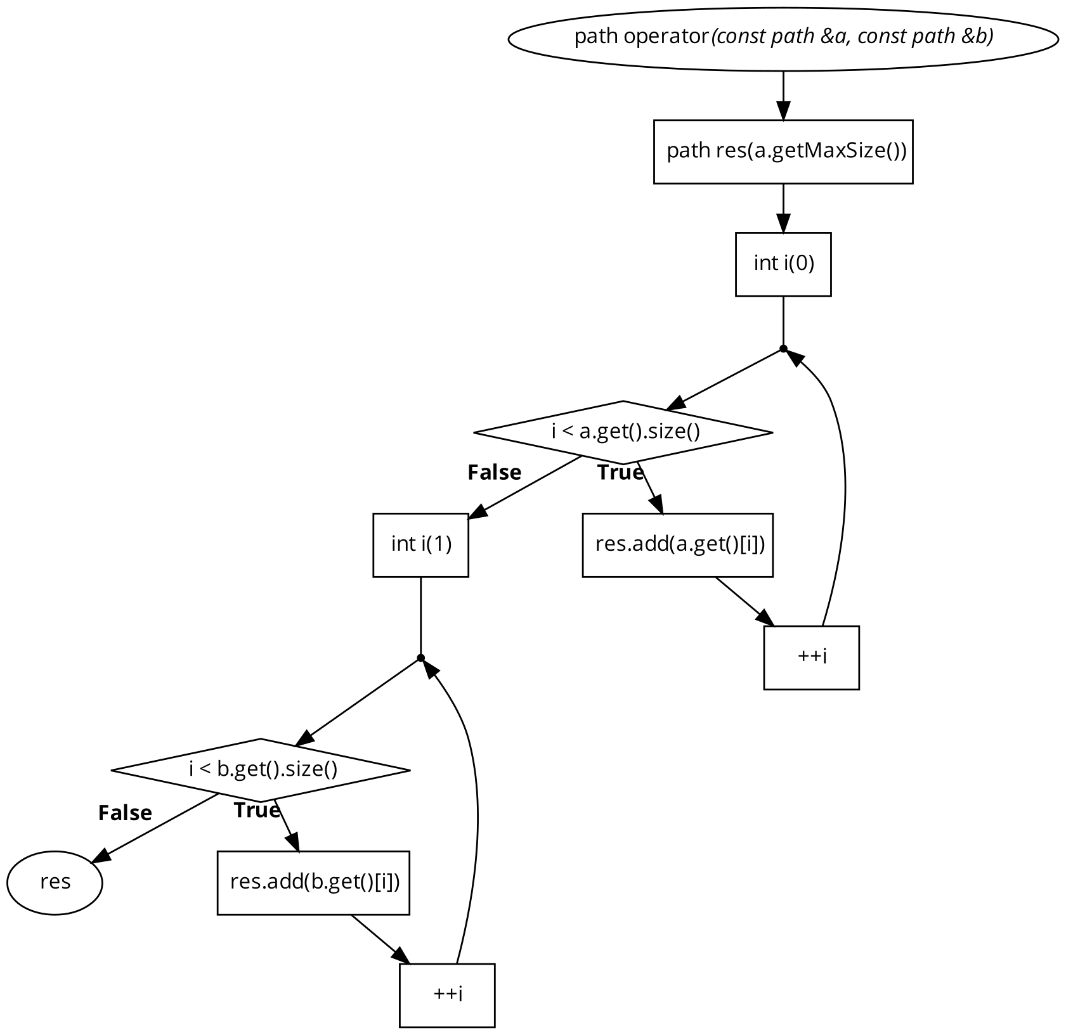
Лог выполнения программы:



Выходной файл res.txt:



Блок-схема:



Прикладная задача:

Пусть дан граф столиц стран мира и всевозможные пути авиасообщения между ними. Сколько всего существует возможных маршрутов для посещения всех столиц и возвращения обратно в столицу отправления, не побывав в какой-либо столице дважды кроме начальной и не летая одним и тем же маршрутом дважды?