Лабораторная работа № 8 по курсу дискретного анализа: жадные алгоритмы

Выполнил студент группы 08-307 МАИ *Боев Савелий*.

Условие:

Заданы N объектов с ограничениями на расположение вида «A должен находиться перед B». Необходимо найти такой порядок расположения объектов, что все ограничения будут выполняться.

Метод решения

Программа начинается с запроса у пользователя количества объектов (N) и количества ограничений (M), которые вводятся через стандартный поток ввода.

Создаётся список смежности **adj** для представления графа, который состоит из N + 1 векторов (для учета того, что номера объектов начинаются с 1), и вектор **in_degree** для отслеживания входящих степеней каждой вершины.

Для каждого ограничения, введенного пользователем в форме пары чисел (A, B), в список смежности добавляется направленное ребро от A к B, увеличивая при этом входящую степень вершины B.

Функция **greedyTopologicalSort** реализует жадный метод топологической сортировки. Она использует множество **zero_in_degree** для отслеживания всех вершин с нулевой входящей степенью и добавляет их в вектор **result**, сохраняя тем самым порядок, в котором вершины могут быть расположены.

На каждом шаге алгоритма выбирается вершина с нулевой входящей степенью из множества **zero_in_degree**, которая затем удаляется из множества, и добавляется в результат.

После добавления вершины в результат, алгоритм уменьшает входящие степени всех смежных вершин. Если входящая степень какой-либо вершины становится нулевой, эта вершина добавляется в множество **zero_in_degree**.

Если после процесса сортировки какие-либо вершины остаются с ненулевой входящей степенью, это означает, что в графе есть цикл, и топологическая сортировка невозможна. В этом случае функция возвращает **false**.

Если функция **greedyTopologicalSort** возвращает **true**, то программа выводит последовательность пар вершин, указывая возможный порядок их расположения. Если возвращается **false**, выводится "-1", сигнализируя о невозможности выполнить сортировку из-за наличия циклов в графе.

Описание программы

Инициализация графа: Программа начинает работу с объявления структур данных, необходимых для представления направленного графа. Она использует список смежности для отображения связей между вершинами и массив для хранения входящих степеней каждой вершины.

Построение графа: Пользователь вводит количество вершин и ограничений, после чего вводит пары чисел, обозначающих направленные связи между вершинами (ограничения). Программа обрабатывает каждую пару, добавляя соответствующие рёбра в граф и обновляя входящие степени вершин.

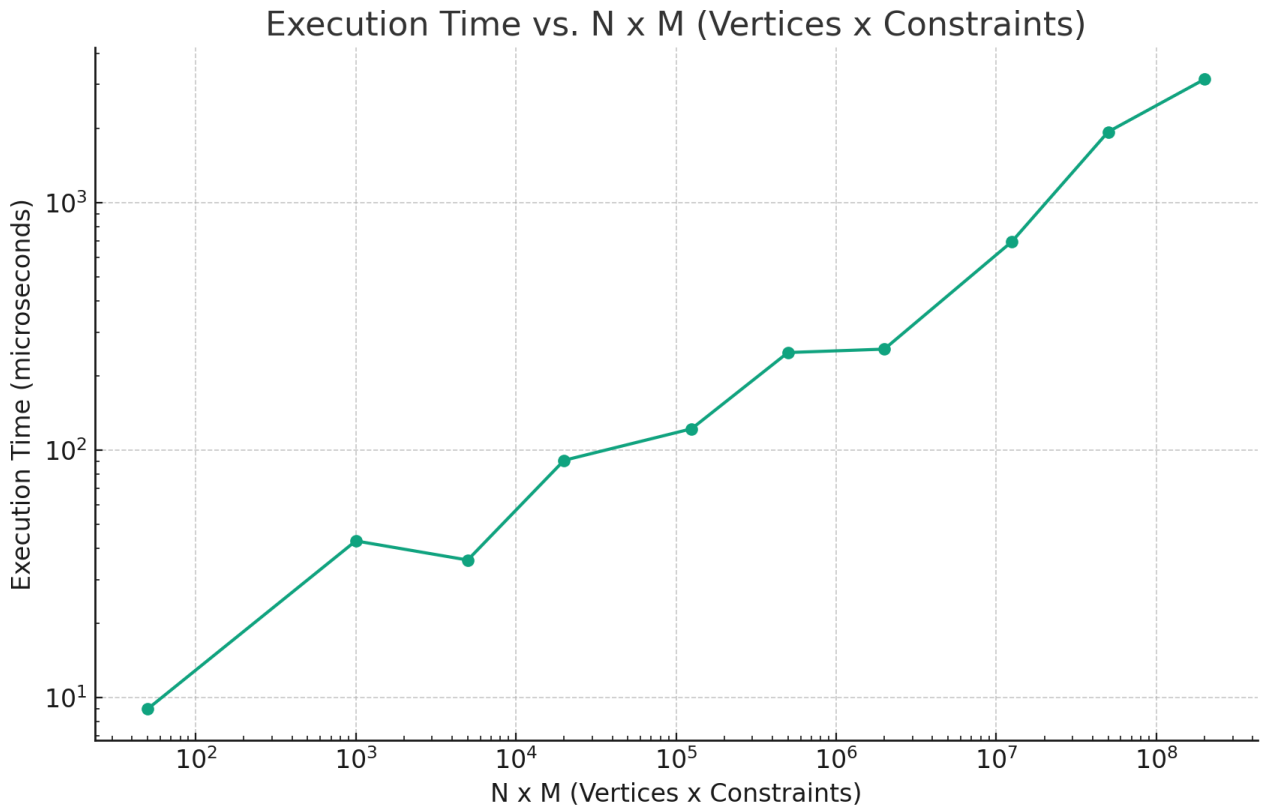
Топологическая сортировка: После построения графа программа пытается выполнить топологическую сортировку с помощью функции greedyTopologicalSort. Алгоритм ищет вершины без входящих рёбер (с нулевой входящей степенью), добавляет их в результат и удаляет все исходящие рёбра, обновляя входящие степени связанных вершин.

Обработка циклов: Если в процессе сортировки обнаруживается, что удалить все рёбра невозможно (присутствуют циклы), алгоритм определяет, что топологическая сортировка невыполнима, и возвращает false.

Вывод результатов: В зависимости от результата топологической сортировки, программа либо выводит упорядоченный список вершин, либо сообщает о невозможности выполнения сортировки, выводя "-1". Успешный вывод означает, что был найден порядок, удовлетворяющий всем ограничениям. В противном случае, наличие циклов указывает на конфликт

ограничений, и программа информирует пользователя о невозможности их удовлетворить.

Тест производительности



Выводы

В ходе выполнения лабораторной работы была решена задача жадным методом. То есть на каждом этапе алгоритма мы выбирали единственный локально наилучший вариант. На практике такое не всегда приводит к оптимальному ответу, но зато такие алгоритмы быстрее как раз за счет отсутствия альтернатив для перебора.