Лабораторная работа №7 по дисциплине «Типы и структуры данных»

Графы

Выполнил студент группы ИУ7-32Б

Чалый Андрей

Цель работы

Реализовать алгоритмы обработки графовых структур: поиск различных путей, проверку связности, построение остовых деревьев минимальной стоимости.

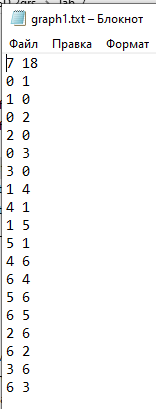
Задание

Обработать графовую структуру в соответствии с указанным вариантом задания. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Ввод данных – на усмотрение программиста. Результат выдать в графической форме.

Задана система двусторонних дорог. Определить, можно ли, закрыв какие-нибудь три дороги, добиться того, чтобы из города А нельзя было попасть в город В

Входные данные

Программа считывает данные о графе из текстового файла. В первой строке файла содержится число вершин и число ребер. В остальных строках файла указываются две вершины.Хранение графа в файле:



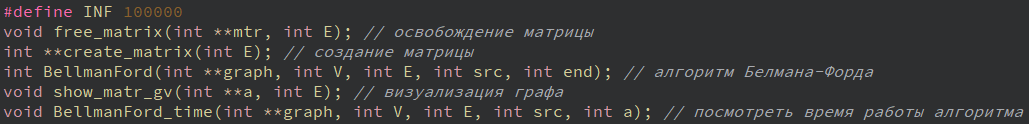
Выходные данные

Визуализированный граф. Начальная и конецная вершины.

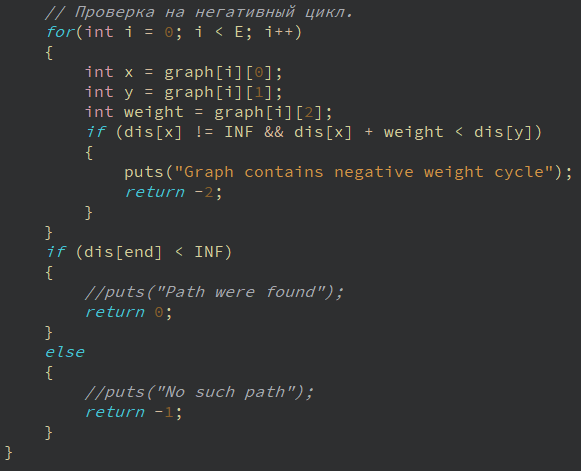
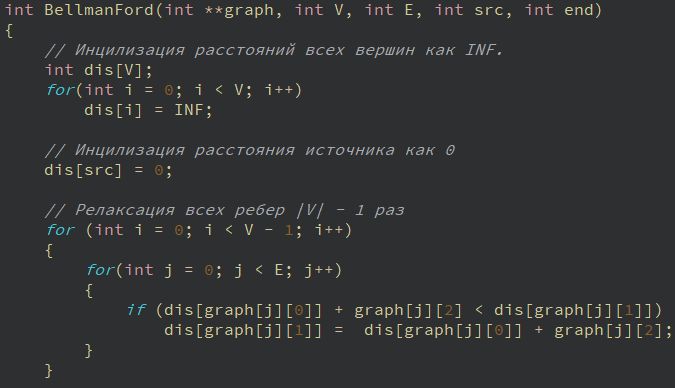
Структура для хранения графа

Для хранения графа используется ассоциативный массив смежности, элементом которого является массив, состоящий из двух вершин и веса ребра, связывающий вершины. Ассоциативный маccив смежности является более удобным способом хранения данных при обработке и заполнении.

Функции



Алгоритм Беллмана-Форда



Аварийные ситуации:

1. Проверка на наличие файла.

При попытке открыть файл, которого не существует, программа выдаст сообщение “FILE ERROR”

2. Проверка на корректность файла

При некорректном файле, программа выдаст сообщение “INCORRECT DATA”

3. Проверка на корректность введенной вершины

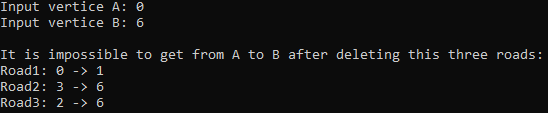
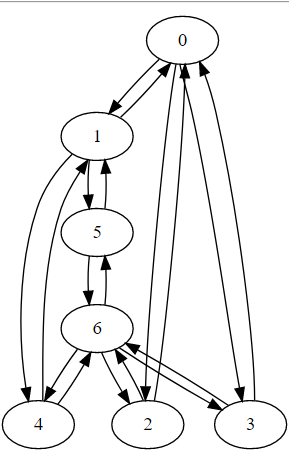
При некорректной вершине, программа выдаст сообщение “INCORRECT VERTICE”

4. Проверка на корректность введенного пути

При некорректном пути, программа выдаст сообщение “INCORRECT PATH”

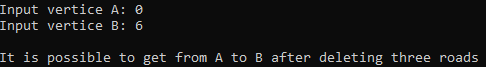
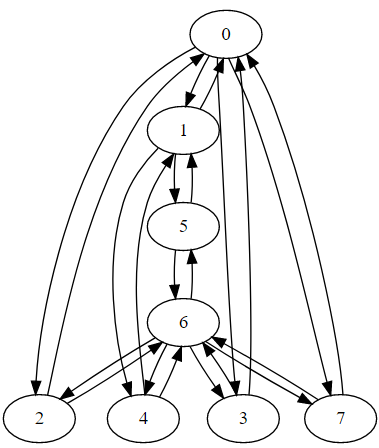
Тесты

1.



Из вершины 0 в вершину 6

2. Исход когда после удаления любых 3 дорог путь все равно есть



Оценка алгоритма:

Алгоритм Беллмана-Форда – алгоритм поиска кратчайшего пути во взвешенном графе. За время O(|V|x|E|) алгоритм находит кратчайшие пути от одной вершины графа до другой. В отличие от алгоритма Дейкстры, алгоритм Беллмана – Форда допускает ребра с отрицательным весом. Также по сравнению с алгоритмом Дейкстра прост в реализации и требует в несколько раз меньше памяти.

Вывод

Для реализации задачи был выбран алгоритм Беллмана-Форда, так как этот алгоритм очень удобен. Для реализация этого алгоритма было необходимо использовать ассоциативный массив смежности.

Контрольные вопросы

1. Что такое граф?

Граф – конечное множество вершин и соединяющих их ребер; G = <V, E>. V – вершины графа, E – ребра. Если пары Е (ребра) имеют направление, то граф называется ориентированным; если ребро имеет вес, то граф называется взвешенным. В моем задании мы имеем дело с ориентированным взвешенным графом.

1. Как представляются графы в памяти?

Существуют различные методы представления графов в программе.

* + - * 1. Матрица смежности B(n \* n) – элемент b[i, j] = вес ребра, если существует ребро, связывающее вершины i и j, и = 0, если ребра не существует.
        2. Список смежностей – содержит для каждой вершины из множества вершин V список тех вершин, которые непосредственно связаны с ней. Входы в списки смежностей могут храниться в отдельной таблице (в массиве), либо же каждая вершина может хранить свой список смежностей.

1. Какие операции возможны над графами?

Основные операции над графами: обход вершин и поиск различных путей: кратчайшего пути от вершины к вершине; кратчайшего пути от вершины ко всем остальным; кратчайших путей от каждой вершины к каждой; поиск эйлерова пути и гамильтонова пути, если таковые есть в графе.

1. Какие способы обхода графов существуют?

Один из основных методов проектирования графовых алгоритмов – это поиск (или обход графа) в глубину (depth first search, DFS). При поиске в глубину, начиная с произвольной вершины v0, ищется ближайшая смежная вершина v, для которой осуществляется поиск в глубину (т.е. снова ищется ближайшая смежная с ней вершина) до тех пор, пока не встретится ранее просмотренная вершина, или не закончится список смежности вершины v (то есть вершина полностью обработана). Если нет новых вершин, смежных с v, то вершина v считается использованной, идет возврат в вершину, из которой попали в вершину v, и процесс продолжается до тех пор, пока не получим v = v0. Иными словами, поиск в глубину из вершины v основан на поиске в глубину из всех новых вершин, смежных с вершиной v. Путь, полученный методом поиска в глубину, в общем случае не является кратчайшим путем из вершины v в вершину u. Это общий недостаток поиска в глубину.

Указанного недостатка лишен другой метод обхода графа – поиск в ширину (breadth first search, BFS). Обработка вершины v осуществляется путем просмотра сразу всех новых соседей этой вершины. При этом полученный путь является кратчайшим путем из одной вершины в другую.

1. Где используются графовые структуры?

Графовые структуры могут использоваться в задачах, в которых между элементами могут быть установлены произвольные связи, необязательно иерархические. Наиболее распространенным является использование графов при решении различных задач о путях, будь то построение коммуникационных линий между городами или прокладка маршрута на игровом поле.

1. Какие пути в графе Вы знаете?

Путь в графе, проходящий через каждое ребро ровно один раз, называется эйлеровым путём; путь может проходить по некоторым вершинам несколько раз – в этом случае он является непростым.

Путь, проходящий через каждую вершину ровно один раз, называется гамильтоновым путем. Как эйлеров, так и гамильтонов путь могут не существовать в некоторых графах.

Условие существования эйлерова пути доподлинно известно, гамильтонова – нет.

1. Что такое каркасы графа?

Каркас графа – дерево, в которое входят все вершины графа, и некоторые (не обязательно все) его рёбра. Для построения каркасов графа используются алгоритмы Крускала и Прима.