|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04** Программная инженерия

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 8 |

**Название:**

Графы

**Дисциплина:** Типы и структуры данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-36Б |  | | А.А. Жаворонкова |
|  | (Группа) | |  | (И.О. Фамилия) |
|  |  | |  |  |
| Преподаватель: | Никульшина Т. А. | |  |  |

Москва, 2022

Описание условия задачи

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, где для любой пары городов есть соединяющий их путь. Найти город с минимальной суммой расстояний до остальных городов.

Описание ТЗ

*Описание исходных данных и результатов*

*Исходные данные:* количество вершин графа, расстояние от каждой вершины до остальных.

*Результаты:* вершина с минимальной суммой расстояний до остальных.

*Способ обращения к программе*

Вызов программы происходит через терминал (main.exe).

*Описание возможных ошибок пользователя*

* Ввод некорректного числа вершин
* Ввод некорректного расстояния
* Ввод таких расстояний, при которых не каждая вершина достижима

Описание внутренних структур данных

­­Граф представлен матрицей меток дуг, в которой петли недопустимы, то есть расстояние из выбранной вершины в нее же считается равным нулю. При этом, если пользователь вводит расстояние равное нулю, то в матрицу оно записывается как INT\_MAX.

Форма представления графа объясняется возможностью на основе исходной матрицы меток дуг, быстро построить матрицу кратчайших расстояний.

Также для сравнения граф представлен следующими структурами:

typedef struct node

{

    int v;

    int weight;

    struct node \*next;

} node\_t;

v – номер вершины  
weight – вес дуги, ведущей в вершину “v”  
next – указатель на следующий элемент

typedef struct graph

{

    node\_t \*\*arr;

    int len;

} graph\_t;

arr – массив списков  
len – длина массива

Описание алгоритма

Для решения данной задачи использовался алгоритм Флойда-Уоршалла. Согласно этому алгоритму, сначала ищется кратчайший путь от одной вершины ко всем вершинам, доступным из нее, затем проводятся те же действия, но пытаясь пройти от этой вершины ко всем доступным из нее, проходя каждый раз через новую вершину (сначала через первую, затем – через вторую и т.д.). Таким образом обрабатываются все вершины.

Временная эффективность и затраты памяти

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Количество вершин | Количество ненулевых путей | Матрица | | Список | |
| Время, нс | Память, б | Время, нс | Память, б |
| 5 | 19 | 1 642 | 100 | 6 426 | 320 |
| 10 | 90 | 9 498 | 400 | 48 543 | 1 456 |
| 25 | 592 | 86 969 | 2 500 | 1 236 910 | 9 488 |
| 50 | 2 419 | 1 234 950 | 10 000 | 24 000 942 | 38 720 |

Из таблицы видно, что матричное представление графа имеет выигрыш по времени и памяти при большой степени заполненности и при малом количестве вершин графа.

Преимущество матрицы по времени объясняется быстротой обращения к нужному элементу: элемент матрицы [i][j] обозначает длину пути из i-ой вершины в j-ую. При работе со списком, чтобы найти длину такого же пути, необходимо просматривать весь список смежности данной вершины.

Выигрыш списка по памяти при малом количестве ненулевых путей объясняется тем, что в списке смежности выбранной вершины не хранится информация о вершинах, к которым нет пути. В то время как в матрице признаком отсутствия пути является INT\_MAX. Также в списках не хранится информация о петлях (в матрице этот путь считается равным нулю).

Вывод

Для представления графа можно использовать матрицу и список смежности. Если приоритетнее получить выигрыш по времени, лучше использовать матричное представление. Если необходим выигрыш по памяти, представление в виде списка окажется в несколько раз выгоднее.

Ответы на вопросы

1. Что такое граф?  
   Граф – это конечное множество вершин и ребер, соединяющих их.
2. Как представляются графы в памяти?  
   Графы представляются в виде матрицы смежности или в виде списка смежности.
3. Какие операции возможны над графами?  
   Поиск кратчайшего пути от одной вершины к другой, поиск кратчайшего пути от одной вершины ко всем другим, поиск кратчайших путей между всеми вершинами, поиск эйлерова пути, поиск гамильтонова пути.
4. Какие способы обхода графов существуют?  
   Обход графа в глубину – ищется ближайшая смешная вершина, из которой возобновляется обход в глубину. Обход графа в ширину – просматривается весь список смежности.
5. Где используются графовые структуры?  
   Графовые структуры используются при исследовании коммуникационных сетей, систем информатики, химических и генетических структур, электрических цепей и других систем сетевой структуры.
6. Какие пути в графе Вы знаете?  
   Эйлеров путь – произвольный путь в графе, проходящий через каждое ребро графа точно один раз.  
   Гамильтонов путь – путь в графе, проходящий в точности один раз через каждую вершину графа.
7. Что такое каркасы графа?  
   Каркасы графа – подграфы, получающиеся при обходе графа в глубину или ширину.