Лабораторная работа №8 по дисциплине «Типы и структуры данных»

Графы

Цель работы :

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

Задание (Вариант 15):

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.

Найти минимальное (по количеству ребер) подмножество ребер, удаление которых превращает заданный связный граф в несвязный.

Исходные данные:

Программа считывает данные о графе из консоли или из текстового файла.
Граф задается количеством вершин и списком ребер.
Ребро задается парой чисел.
Количество вершин должно быть меньше 100.

Тесты

При вводе некорректного ребра пользователь увидит следующий текст:

"Неверное ребро. Попробуйте еще раз."

При вводе некорректного числа вершин или пункта меню пользователь увидит следующий текст: **"Error"**

Интерфейс программы:

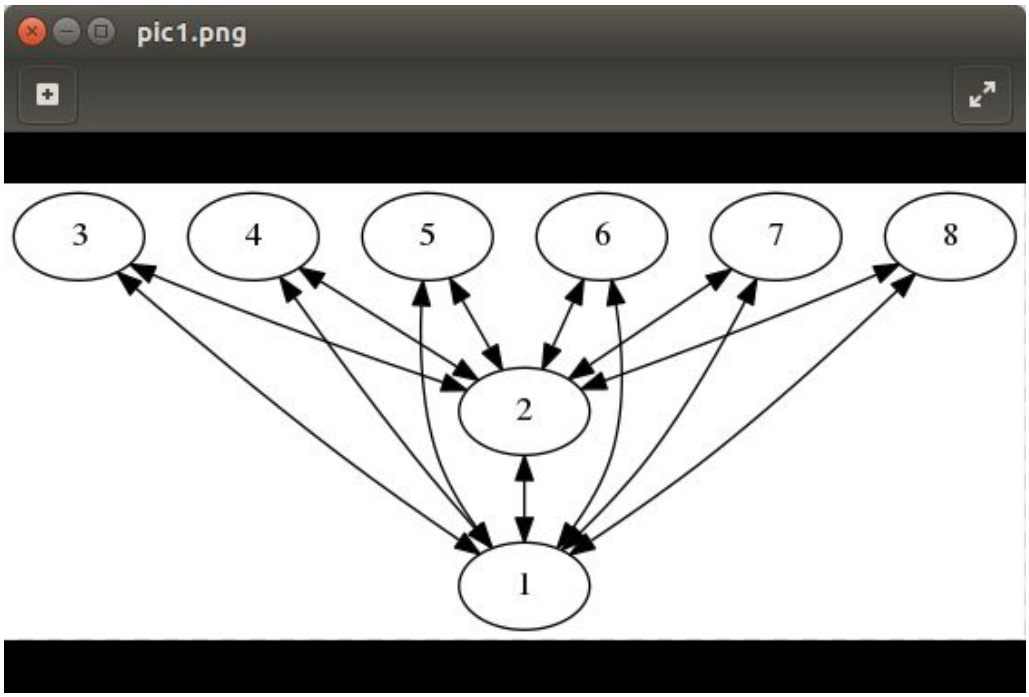
1)Главное меню

```
**************************************************************
Выберите одно из следующих действий:
0: Загрузить граф из файла
1: Задать граф
2: Закончить работу
```

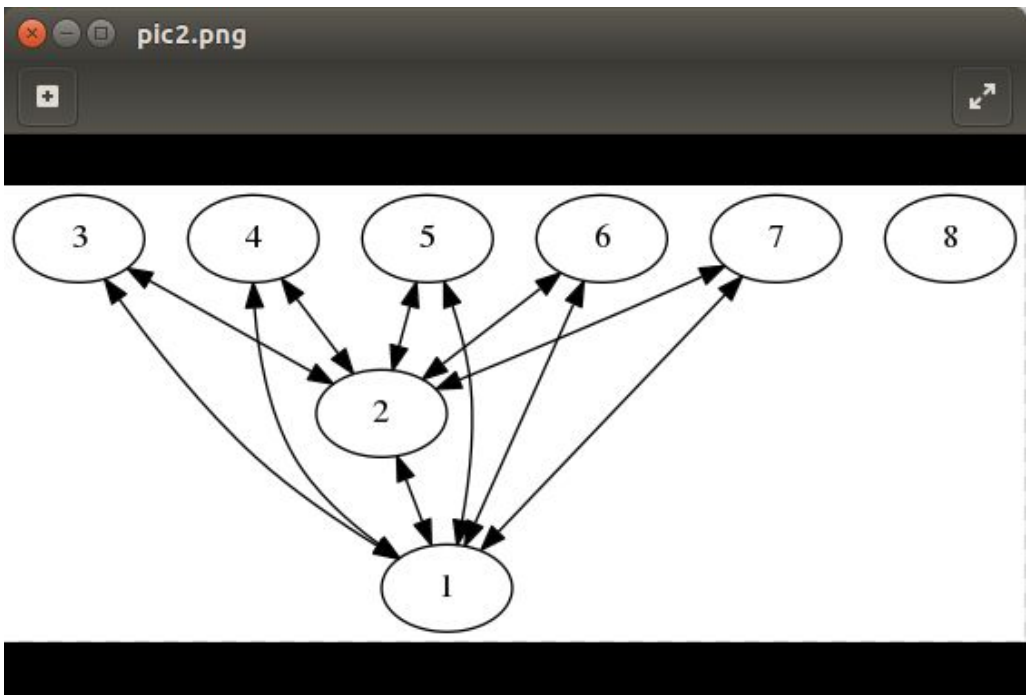
2) Загрузка преднастроенных графов

```
Выберите один из графов
0. Три треугольника
1. Два треугольника
2. Два квадрата
3. Ромбы в ромбе
4. Пятиугольник
5. Квадрат с диагональю
```

3) Отображение исходного графа



4) Отображение несвязного графа



5) Список и количество удаленных ребер

```
Ребро для удаления 8-1
Ребро для удаления 8-2
Всего удалено 2 ребер(а)
```

Особенности реализации:

Был реализован алгоритм, идея которого описывается следующим образом: Будем итеративно повторять следующий процесс: находить минимальный разрез между какой-нибудь парой вершин s и t, а затем объединять эти две вершины в одну (соединяя списки смежности). В конце концов, после n-1 итерации, граф сожмётся в единственную вершину и процесс остановится. После этого ответом будет являться минимальный среди всех n-1 найденных разрезов.

Внутренняя структура данных:

Граф представлен в виде матрицы смежности размера n*n, где в (i,j) ячейке хранится 0, если дуги между вершинами нет, и 1 в противном случае.
Матрица смежности является более удобным способом хранения данных при обработке. Недостатком выбранной реализации является большое количество требуемой памяти.

Вывод

Для реализации данной задачи применен алгоритм Штор-Вагнера с выводом удаляемых вершин
Для данного алгоритма не нужно использовать стек или очередь, но побочным эффектом данной реализации является порча оригинального графа, при этом данный алгоритм легко модернизировать для графа со взвешенными ребрами и вершинами.

Контрольные вопросы

1.Что такое граф?

Граф – конечное множество вершин и соединяющих их ребер; G = <V, E>. Если пары E (ребра) имеют направление, то граф называется направленным; если ребро имеет вес, то граф называется взвешенным.

2.Как представляются графы в памяти?

Существуют различные методы представления графов в программе. Матрица смежности B(n*n) – элемент b[i,j]=1, если существует ребро, связывающее вершины i и j, и =0, если ребра не существует.

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, либо же каждая вершина может хранить свой список смежностей.

3.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.

4.Какие способы обхода графов существуют?

Один из основных методов проектирования графовых алгоритмов – поиск в глубину. Начиная с некоторой вершины v0, ищется ближайшая смежная ей вершина v, для которой в свою очередь осуществляется поиск в глубину 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. При просмотре используется стек.

Поиск в ширину – обработка вершины V осуществляется путём просмотра сразу всех «новых» соседей этой вершины, которые последоватеьно заносятся в очередь просмотра.

5.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.

6.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.

7.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.