

Домашнее задание 1.

Изоморфизм графов

Этап II. Написание программы

В рамках данного этапа требуется реализовать программу, решающую задачу изоморфизма графов согласно алгоритму, предложенному в рамках первого этапа.

Формат входных данных.

Описание графов считывается программой либо со стандартного потока ввода (cin), либо из файла. При этом программа должна корректно обрабатывать следующие два основных формата описания графов:

1. матрицы смежности (M);
2. списки ребер (L).

Каждая строка в указанных описаниях непосредственно относится к описанию формата или является комментарием в стиле языка C (например, "//строка с комментарием")

Весами ребер могут быть целые и вещественные числа, а также символьные последовательности. В случае, если веса являются вещественными числами, то равенство весов устанавливается с точностью 0.0001.

Матрица смежности (формат M)

Первая строка содержит символ, указывающий на тип графа:

1. «D» - ориентированный граф;
2. «U» - неориентированный граф.

Вторая строка содержит информацию о том, является ли граф взвешенным:

1. «N» - граф не содержит весов;
2. «E» - граф содержит веса, приписанные ребрам;
3. «V» - граф содержит веса, приписанные вершинам;
4. «A» - граф содержит веса, приписанные и ребрам, и вершинам.

Если граф содержит веса на вершинах или на ребрах, то после указания символа во второй строке через пробел указывается тип весов. Сначала указывается тип для вершин, потом для ребер. При этом используются следующие обозначения: «int» - целочисленные веса, «float» - вещественные веса, «string» - строковые веса.

Третья строка содержит число N – число вершин графа (по умолчанию, считается, что все вершины пронумерованы от 0 до N-1). Последующие N строк содержат описание матрицы смежности графа, причем элементы матрицы в каждой строке разделены пробелами. Если граф содержит веса, приписанные вершинам графа, то они указываются на диагонали матрицы.

Примеры корректного входного описания графа в формате M:

```
//Vertex and edge weighted 3-vertex directed cycle
D
A float float
3
```

```

2.03 1.3 0
0 1.2 3.4
2.1 0 3.2

//3-vertex graph with 1 edge
U
N
3
0 1 0
1 0 0
0 0 0

//Vertex weighted 3-vertex undirected cycle
U
V float
3
1.2 1 1
1 2.3 1
1 1 3.1

//Symbol labelled 3-vertex directed cycle
D
E string
3
0 'v(1)' 0
0 0 'v(2)'
'v(3)' 0 0

```

Список ребер (формат L)

Как и в случае формата М, первые три строки характеризуют информацию о типе графа, наличии весов, приписанных вершинам или ребрам, и числе вершин графа N (по умолчанию, считается, что все вершины пронумерованы от 0 до N-1). Четвертая строка содержит число K – число ребер графа. Последующие K строк содержат описание ребер графа. При этом в строке через пробел указываются номера вершин и вес, ребра, если они заданы. В случае, если граф имеет веса, приписанные вершинам, то последние N строк содержат информацию об этих весах.

Примеры корректного входного описания графа в формате L:

```

//Vertex and edge weighted 3-vertex directed cycle
D
A float float
3
3
0 1 1.3
1 2 3.4
2 0 2.1
2.03
1.2
3.2

//3-vertex graph with 1 edge
U

```

```

N
3
1
0 1

//Vertex weighted 3-vertex undirected cycle
U
V float
3
3
0 1
1 2
2 0
1.2
2.3
3.1

//Symbol labelled 3-vertex directed cycle
D
E string
3
3
0 1 'v(1)'
1 2 'v(2)'
2 0 'v(3)'

```

Формат выходных данных.

В результате своей работы программа в стандартный поток вывода (cout) или в файл выдает информацию о том изоморфны ли полученные на вход графы или нет. В случае, если графы изоморфны, то программа выдает информацию о найденном изоморфизме.

Если графы изоморфны, то программа выдает «YES», если нет, то «NO». Далее, в первом случае, программа последовательно выдает N строк, где N – число вершин в графе, которые задают вектор-столбец найденного изоморфизма.

Кроме того, при завершении своей работы программа выдает «1», если графы изоморфны, «0», если графы не изоморфны, и «2», если программа завершила свою работу и при этом не могла установить являются ли графы изоморфными или нет.

Примеры корректного выхода программы

```

//Isomorphism 0->2 1->1 2->0
YES
2
1
0

//No isomorphism
NO

```

Параметры командной строки.

Программа должна поддерживать следующие параметры командной строки:

1. `--help, -h` – при передаче этого параметра программа должна вывести краткую справку о работе с программой, которая включает краткое описание программы, а также все параметры командной строки и их назначение;
2. `--input (M|L), -i (M|L)` – параметр определяет тип входного формата (в скобках указаны возможные типы входного формата: M – графы задаются в виде матриц смежности, L – графы задаются списками своих ребер);
3. `--input_file_1 "filename1.txt" -input_file_2 "filename2.txt", -1 "filename1.txt" -2 "filename2.txt"` – если указаны эти параметры, то программа считывает вход из файлов с именем "filename1.txt" и "filename2.txt", если параметр не указан, то считывание происходит со стандартного потока ввода (cin);
4. `--output_file "filename.txt", -O "filename.txt", -O` – если указан этот параметр, то программа записывает результат работы в файл с именем "filename.txt", если параметр не указан, то запись происходит в стандартный поток вывода (cout);

Примеры корректного задания параметров командной строки

```
isomorphism.exe --help /*вывод краткой справки о работе программы*/
```

```
isomorphism.exe -i M /*программа считывает со стандартного потока ввода графы, заданные в виде матриц смежности, и выдает результат своей работы в стандартный поток вывода.*/
```

```
isomorphism.exe --input L -1 test1.txt -2 test2.txt -O result.txt /*программа считывает графы, заданные списками своих ребер, из файла test1.txt и test2.txt и записывает результат своей работы в файл result.txt.*/
```

Примеры некорректного задания параметров командной строки

```
isomorphism.exe -h -i M /*несогласованные параметры командной строки*/
```

```
isomorphism.exe -i /*пропущен обязательный параметр при "-i"*/
```

```
isomorphism.exe -1 test.txt /*не задан входной формат представления графов и задан только один из двух входных файлов.*/
```