

Расчетная работа

Мацукевич Захар 421703

Цель:

Продемонстрировать работу программы решения теоретико-графовой задачи по построению дерева кратчайших путей заданного графа.

Ключевые понятия:

Граф – совокупность двух множеств – множества самих объектов (вершин) и множества их парных связей (ребер).

Ориентированный граф (кратко орграф) – граф, рёбрам которого присвоено направление.

Путь в графе – это последовательность рёбер, в которой конец каждого ребра (кроме последнего) совпадает с началом следующего.

Список смежности – один из способов представления графа в виде коллекции списков вершин. Каждой вершине графа соответствует список, состоящий из «соседей» этой вершины.

Матрица смежности – это вид представления графа в виде матрицы, когда пересечение столбцов и строк задаёт дуги.

Неориентированный граф – граф, в котором рёбра не имеют направления. На рисунке выше показан как раз неориентированный граф. В таком неориентированном графе можно перемещаться вдоль ребра в любом из двух направлений.

Взвешанный граф – граф, ребра которого имеют некое числовое значение.

Дерево кратчайших путей – подграф исходного графа, одна из вершин которого является вершиной, кратчайшие пути из которой хранит граф, в котором каждый путь от корня до любой другой вершины является кратчайшим в исходном графе.

Демонстрация работы программы в семантической памяти на примере невзвешенного неориентированного графа:

Для лучшего восприятия картинок ненужные на соответствующем этапе эл-ты SCg будут скрыты. Каждому пункту соответствует рисунок с соответствующим номером.

1. Задаем сам граф, его принадлежность к множеству графов и непринадлежность к множествам ориентированных и взвешенных графов, а также для всех его вершин

и ребер указываем принадлежность к соответствующим множествам. Указываем начальную вершину, пусть это будет вершина “1”. Отметим её же посещенной, указав принадлежность к соответствующему множеству.

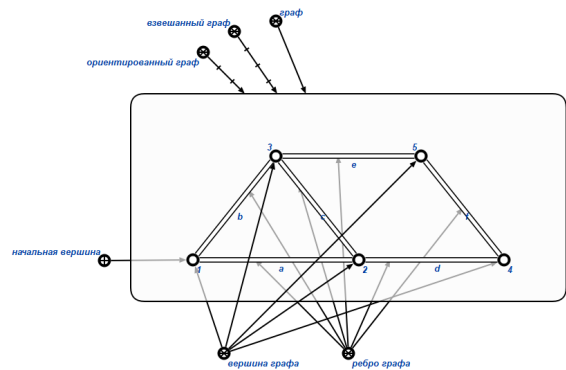


Рис.1

2. Начать следует с первой вершины. Зададим итерацию “it1” и присвоим ей все происходящее. Отметим вершину “1” как текущую вершину. Путь в не имеет ребер, т.к. изначально мы находимся в конце пути. Значит задаем элемент “путь 1-1” и указываем, что это объединение из вершины “1” и это граф. Также связываем вершину “1” с “путь 1-1” ролевым отношением “путь сюда*”.

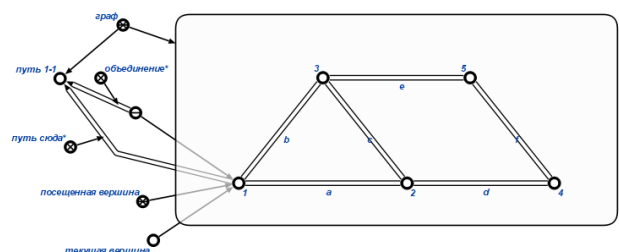


Рис.2

3. Теперь нужно объединить выходящие из текущей вершины ребра и узлы, в которые они приходят, в структуру “it<номер итерации>” какое ребро к какому узлу ведет

и связать структуры отношением “следующая”. Указываем принадлежность вершины “1” к множеству пройденных вершин. Таким образом получаем очередность выполнения операций, зная ребро, по которому попали в каждый узел.

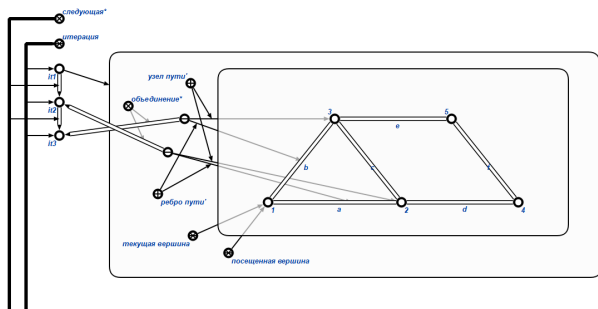


Рис.3

4. Закончив первую итерацию идем к следующей. Помечаем вершину пути как текущую. Если она помечена как пройденная, то ничего не делаем и идем к следующей итерации. Иначе объединяем текущую вершину, ребро пути и “путь сюда*” для вершины, с которой еще связано ребро пути, в данном случае - это вершина “2”, ребро “а” и “путь 1-1”. Результат объединения - “путь 1-2”. Связываем “путь 1-2” с вершиной “2” отношением “путь сюда*”. “Путь 1-2” также граф.

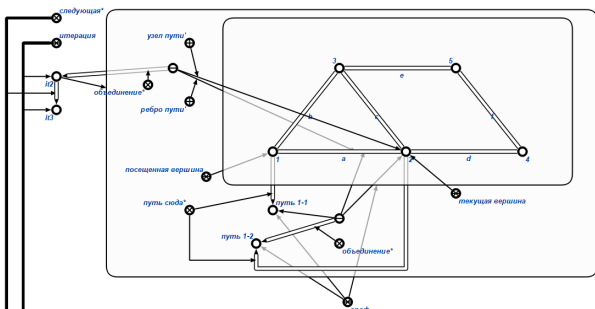


Рис.4

5. Теперь определим следующие итерации. Объединяем ребра “с” и “d” с вершинами “3” и “4” соответственно в итерации “it4” и “it5”, обозначая ребра и вершины пути для каждой итерации. Помечаем “2” пройденной вершиной.

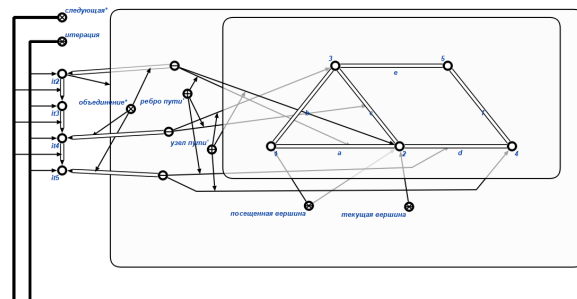


Рис.5

6. На третьей итерации делаем все те же действия, что на итерации “it2”: строим “путь 1-3” объединяя текущую вершину “3”, ребро пути "b" и “путь 1-1”.

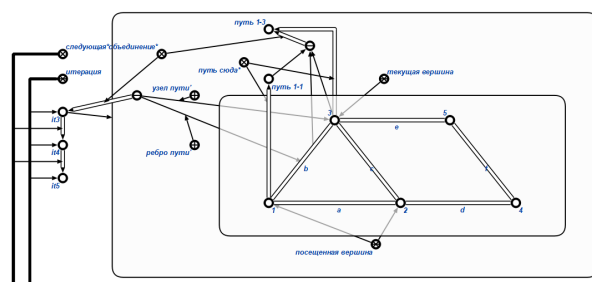


Рис.6

7. Определяем следующие итерации. Указываем узел “3” как посещенную вершину.

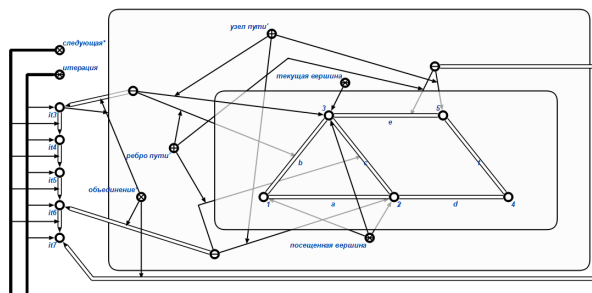


Рис.7

8. Итерация “it4”- ребро “с” и вершина “3”. Отмечаем её как текущую. Видим, что она уже принадлежит множеству посещенных вершин, так что никаких операций производить не нужно, кратчайший путь в эту вершину уже записан как “путь 1-3”. Переходим к “it5”.

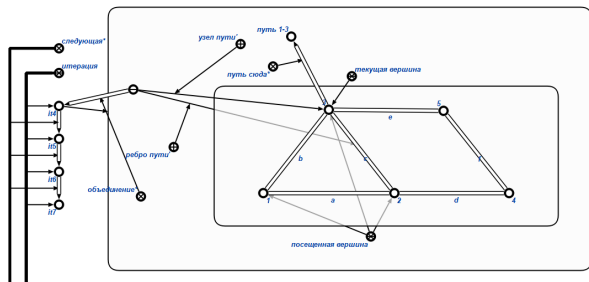


Рис.8

9. На пятой итерации рассматривается вершина "4" и ребро "d". Вершина "4" не была посещена, так что помечаем ее как текущую, объединяем вершину и ребро пути пятой итерации с путем "путь 1-2" в "путь 1-4" и указываем, что это "путь сюда*" для узла "4".

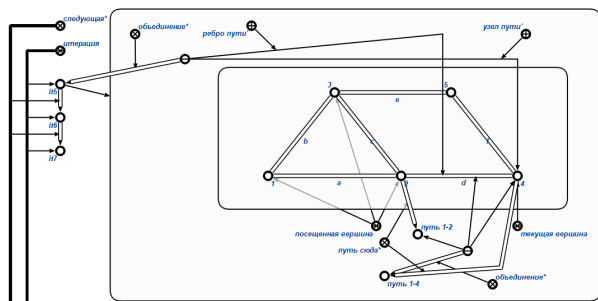


Рис.9

10. Объединяем ребро "f" и вершину "5" в следующую итерацию "ит8". Помечаем "4" как посещенную вершину и идем дальше в очереди.

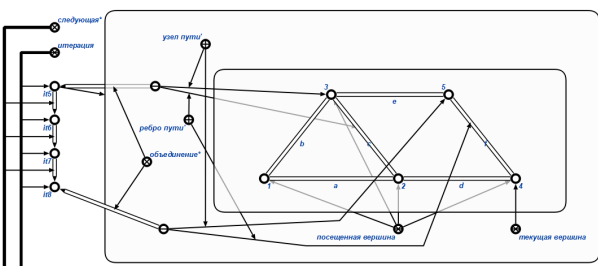


Рис.10

11. Итерация "ит6" имеет вершину пути "2" и ребро пути "c", отмечаем вершину как текущую. Замечаем ее принадлежность множеству посещенных вершин и наличие связи "путь сюда*", значит игнорируем эту итерацию и идем дальше.

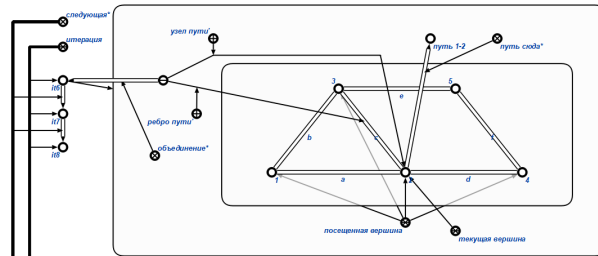


Рис.11

12. На седьмой итерации помечаем "5" как текущую вершину. Объединяем ребро пути "e", текущую вершину и "путь 1-3" в "путь 1-5". Объединяем ребро "f" и вершину "4" в "ит9". Помечаем вершину "5" как посещенную и движемся дальше.

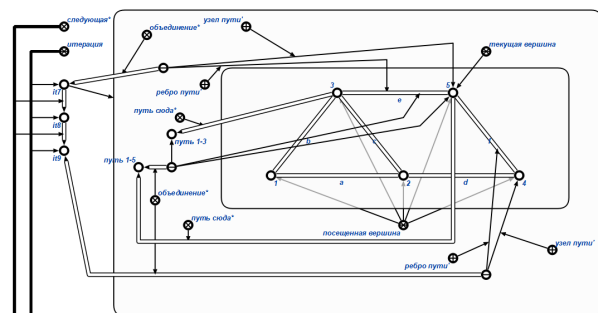


Рис.12

13. Переходим к "ит8". Ребро пути - "f", вершина пути - "5". Эта вершина принадлежит множеству посещенных и путь сюда уже построен, значит итерация игнорируется. Идем дальше.

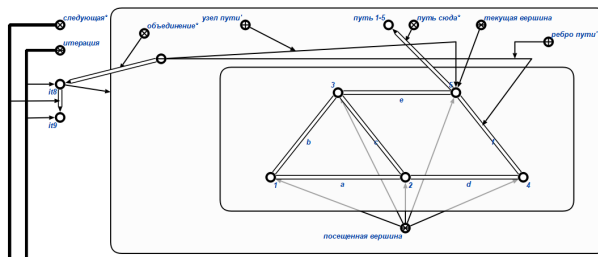


Рис.13

14. У девятой итерации вершина пути также отмечена посещенной, так что и она тоже игнорируется.

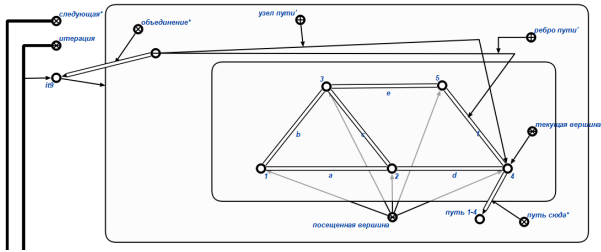


Рис.14

15. Итак, у нас появилось некоторое количество новых графов “путь 1-п”, где п - номер крайней вершины графа, то есть каждый такой граф - путь из первой вершины в п-ую. Объединение всех этих путей - есть дерево кратчайших путей исходного графа.

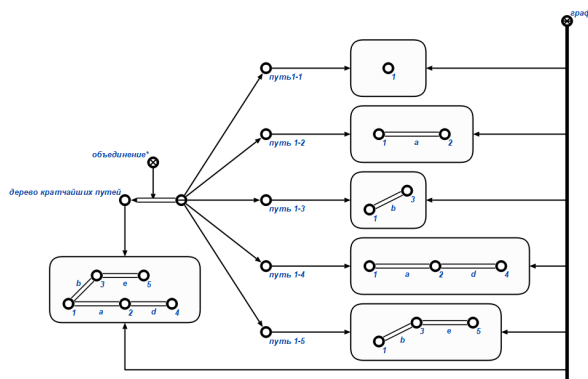


Рис.15

ВВОД:
 0 1 1 0 0
 0 0 0 1 0
 0 1 0 0 1
 0 0 0 0 1
 0 0 0 0 0

ВЫВОД:
 0 1 1 0 0
 0 0 0 1 0
 0 0 0 0 1
 0 0 0 0 0
 0 0 0 0 0

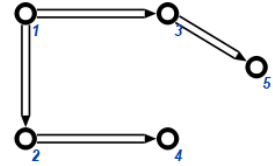
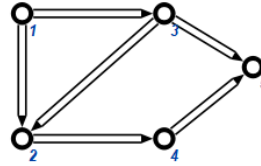


Рис.17

ВВОД:
 0 3 2 0 0
 3 0 1 4 0
 2 1 0 0 5
 0 4 0 0 2
 0 0 5 2 0

ВЫВОД:
 0 3 2 0 0
 0 0 0 4 0
 0 0 0 0 5
 0 0 0 0 0
 0 0 0 0 0

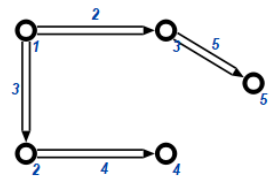
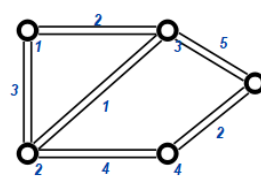


Рис.18

На рисунках 16-19 представлен вывод работы программы для неориентированного графа, ориентированного графа, взвешенного неориентированного графа, взвешенного ориентированного графа в соответствующем порядке.

ВВОД:
 0 1 1 0
 1 0 1 1
 1 1 0 0
 0 1 0 0

ВЫВОД:
 0 1 1 0
 0 0 0 1
 0 0 0 0
 0 0 0 0

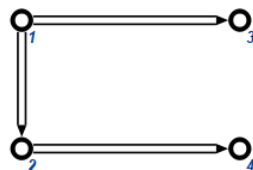
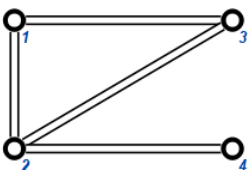


Рис.16

ВВОД:
 0 6 3 0
 0 0 0 1
 0 2 0 0
 0 0 0 0

ВЫВОД:
 0 0 3 0
 0 0 0 1
 0 2 0 0
 0 0 0 0

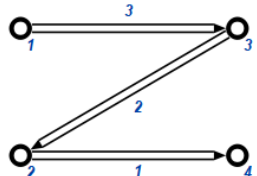
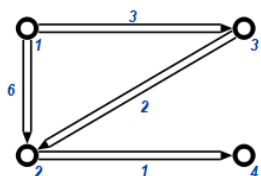


Рис.19

Выводы:

В результате выполнения данной расчётной работы был формализован алгоритм нахождения дерева кратчайших путей невзвешенных неориентированных и ориентированных графах, были изучены:

- Основы теории графов
- Способы представления графов
- Базовые алгоритмы для работы с графами
- Основы SC-кода и SC-алфавита

Источники:

- Оре О. Теория графов. – 2-е изд.. – М.: Наука, 1980. – С. 336.
- Кормен Т. Х. и др. Часть VI. Алгоритмы для работы с графами // Алгоритмы: построение и анализ = Introduction to Algorithms. – 2-е изд.. – М.: Вильямс, 2006. – С. 1296.
- Харари, Ф. Теория графов / Ф. Харари / Пер. с англ. и предисл. В.П. Козырева. Под ред. Г.П. Гаврилова. Изд. 2-е. – М.: Едиториал УРСС, 2003. – 269 с.