**Глава II. Основы теории графов.**

**п.1. Основные понятия и определения**

***Граф*** представляет собой непустое конечное множество вершин *S* и множество рёбер *U,* оба конца которых принадлежат множеству *S*. Обозначать граф будем *G = (S,U)*.

Пусть ,  ‑ вершины,  ‑ соединяющее их ребро. Тогда вершина  и ребро  ‑ ***инцидентны*.**

Два ребра, инцидентные одной вершине, называются ***смежными****.*

Две вершины, инцидентные одному ребру, также называют ***смежными*.**



Обычно графы изображают диаграммой: вершины – точками, рёбра ‑ линиями.

На рис.1  и  ‑ инцидентны; , ,  ‑ смежные (общая вершина ); ,  ‑ смежные (инцидентны ).

***Степенью вершины*** называется число рёбер графа, которым принадлежит эта вершина. Обозначают .

Вершина графа, для которой , называется *изолированной*.

Если  нечётное число, то вершина называется *нечётной*, если  чётное число, то вершина называется *чётной*.

На рис.1 вершины ,  ‑ нечётные, т.к. , ; вершины ,  ‑ чётные, т.к. , ; вершина  ‑ изолированная, т.к. .

***Ориентированный граф*** или ***орграф*** представляет собой пару *G = (S,U)*, где *S* – конечное множество вершин, а *U* ‑ множество ориентированных рёбер (называемых дугами), соединяющих пары вершин.



Если ‑ дуга орграфа, то ‑ *начало* дуги, а ‑ *конец* дуги. На рис.2 показан граф *G = (S,U)*, у которого , 

Одна и та же вершина ориентированного графа может служить началом для одних рёбер и концом для других, поэтому различают две степени вершины: степень входа и степень выхода.

*Степень входа* ‑ число входящих в  дуг. *Степень выхода* ‑ число выходящих из  дуг.

На рис.2 , ; , ; , ; , .

**п.2. Маршруты, цепи, циклы**

***Маршрутом*** в графе называется чередующаяся последовательность вершин и рёбер, в которой любые два соседних элемента инцидентны:  Если  то маршрут *замкнут*, в противном случае *открыт*.

Маршрут называется ***цепью***, если все его рёбра различны, и ***простой цепью***, если все его вершины различны.

Замкнутая цепь называется ***циклом***; замкнутая простая цепь называется ***простым циклом***.

Граф без циклов называется ***ациклическим***.

Важным понятием теории графов является *связность*. Граф называется ***связным***, если любые две его несовпадающие вершины соединены маршрутом. Например, на рис.3*а* изображён несвязный граф, нет маршрута из  в ; из  в ; из  в ; из  в . Связный граф показан на рис.3б, где ‑ простая цепь, а ‑ простой цикл.











*а)*

*Рис. 3*





















*б)*

Для орграфа существует понятие сильной связности.

***Путём*** в графе называется ориентированный маршрут.

Орграф называется ***сильно связным***, если для любых двух вершин  найдётся путь с началом в , и концом в .

На рис.4,*а* показан сильно связный граф, а на рис.4,*б* связный граф (нет пути из  в ).











*а)*

*Рис. 4*

















*б)*















Замкнутый путь в орграфе называется ***ориентированным циклом*** или ***контуром***. Например, в графе на рис.4,*а* путь  является ориентированным циклом, а в графе на рис.4,*б* контуров нет.

**п.3. Способы задания графов**

В подавляющем большинстве случаев граф задаётся матрицей. Для расчётов на ЭВМ это единственный способ.

Наиболее часто граф задают с помощью матриц смежности и инциденций.

***Матрица смежности вершин*** – квадратная матрица порядка , где  ‑ число вершин.



*Рис. 5*

*В*

*D*

*C*

*E*

*А*



















Дуга  на рис.5 называется ***петлёй***.



Для ориентированного графа на рис.5 матрица смежности имеет вид:



Если предположить, что граф на рис.5 неориентированный, то матрица смежности будет следующей:



***Матрица инцидентности рёбер*** – матрица размера , где  ‑ число вершин,  ‑ число рёбер.



*Рис. 6*

*В*

*С*

*D*

*А*











Для ориентированного графа на рис.6 матрица инцидентности имеет вид:



Если предположить, что граф на рис.6 неориентированный, то матрица инцидентности будет следующей:





**п.4. Упорядочивание дуг и вершин орграфа**

Пусть  связный орграф без контуров.

*Определение.* Под упорядочиванием вершин графа  понимается разбиение вершин на группы, при котором:

1) вершины первой группы не имеют предшествующих вершин, а вершины последней группы последующих;

1-я группа

последняя группа

2) вершины любой другой группы не имеют предшествующих в следующей группе;

*i*-я группа

(*i+1*)-я группа

3) вершины одной и той же группы дугами не соединяются.

Такое разбиение всегда возможно. В результате подобного упорядочивания получается граф, изоморфный исходному.

*Рис. 7*

*В*

*С*

*D*

*А*

*Е*

***Графический способ упорядочивания вершин графа (алгоритм Фалкерсона****)*

**1**. Найти вершины графа, в которые не входит ни одна дуга. Они образуют 1-ю группу. Пронумеровать вершины группы в произвольном порядке.

**2**. Вычеркнуть все пронумерованные вершины и дуги, из них исходящие. В получившемся графе найдётся, по крайней мере 1 вершина, в которую не входит ни одна дуга. Этой вершине, входящей во 2-ю группу присвоить очередной номер и т.д. второй шаг повторять до тех пор, пока не будут упорядочены все вершины.

*С*

*D*

*А*

*Е*

*Рис. 8*

*Пример*. Упорядочить вершины орграфа рис.7.

**1**. Вершина *В*-не содержит входящих дуг, следовательно, отнесём её к первой группе.

**2**. Вычёркиваем вершину *В* и все дуги, исходящие из *В*. Получим граф на рис.8.

В полученном графе опять находим вершины, в которые не заходит ни одна дуга. Это вершина *D*. Отнесём её ко второй группе.

*С*

*А*

*Е*

*Рис. 9*

**3**. Вычёркиваем вершину *D* и все дуги, исходящие из *D*. Получим граф на рис.9.

В полученном графе опять находим вершины, в которые не заходит ни одна дуга. Это вершина *Е*. Отнесём её ко третьей группе.

**4**. Вычёркиваем вершину *Е* и все дуги, исходящие из *Е*. Получим граф на рис.10.

В полученном графе опять находим вершины, в которые не заходит ни одна дуга. Это вершина *А*. Отнесём её ко четвёртой группе.

*С*

*А*

*Рис. 10*

*С*

*Рис. 11*

**5**. Вычёркиваем вершину *А* и все дуги, исходящие из *А*. Получим граф на рис.11.

Находим вершины, в которые не заходит ни одна дуга. Это вершина С. Отнесём её к пятой группе.

Изоморфный граф с упорядоченными вершинами представлен на рис.12.

*Рис. 12*

*А*

*В*

*С*

*D*

*E*

*Матричный способ упорядочивания вершин графа.*

Рассмотрим матрицу смежности вершин графа:



Определим полустепени входа вершин графа

,

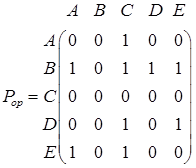
полустепени выхода вершин графа

.

Вычислим полустепени входа для графа, заданного матрицей смежности:

, , , , .

Так как , то в вершину *В* не заходит ни одна дуга и вершина *В* образует первую группу. Вычёркиваем из матрицы *Р* строку, соответствующую вершине *В*. Этим мы исключим из рассмотрения вершину *В* и дуги из неё исходящие.



Получим матрицу



у которой , , , .

Так как , то в вершину *D* не заходит ни одна дуга и вершина *D* образует вторую группу. Вычёркиваем из матрицы  строку, соответствующую вершине *D*. Получим матрицу :



у которой , , .

Так как , то в вершину *Е* не заходит ни одна дуга и вершина *Е* образует третью группу. Вычёркиваем из матрицы  строку, соответствующую вершине *Е*. Получим матрицу :



у которой , .

Так как , то в вершину *А* не заходит ни одна дуга и вершина *А* образует четвёртую группу. Вычёркиваем из матрицы  строку, соответствующую вершине *А*. Получим матрицу :



у которой .

Так как , то в вершину *С* не заходит ни одна дуга и вершина *С* образует пятую группу.

**п.5. Нахождение кратчайших путей. Алгоритм Дейкстры**

Едсгер Дейкстра (1930-2002) – нидерландский математик.

В различных приложениях теории графов дугам (рёбрам) графов, моделирующих реальные процессы, обычно сопоставляются какие-либо числовые характеристики. Например, если дугами изображаются транспортные магистрали, то числовой характеристикой дуги может быть пропускная способность соответствующей магистрали и т.п. В таких случаях говорят, что дугам графа приписаны определённые веса.

Пусть  ‑ орграф. Если каждой дуге  поставлено в соответствие число , то граф называется графом со взвешенными дугами или ***сетью***. При этом вершины графа называют ***узлами сети***. Число  называется **весом дуги** . **Весом пути µ** (длиной, стоимостью и т.д. в зависимости от контекста) сети  называется число .

Понятие сети и веса маршрута для неориентированного графа определяется аналогично.

*Рис. 13*

*А*

*В*

*С*

*F*

*D*

*E*

**2**

**3**

**4**

**2**

**1**

**1**

**5**

Сеть на рис.13 может представлять, например, длины дорог в километрах между шестью деревнями. Т.к. количество вершин в этом графе невелико, то для нахождения наиболее короткого пути между деревнями, нужно перебрать все возможные пути. Это возможно, т.к. количество вершин в этом графе невелико. В реальных задачах число вершин настолько велико, что такой подход к поиску кратчайшего пути слишком неэффективен.

Существует множество алгоритмов поиска кратчайшего пути, но мы познакомимся только с одним, алгоритмом Дейкстры.

Кратчайший путь – это путь минимального общего веса соединяющий выбранные вершины. Общий вес кратчайшего пути, ведущего из вершины  в вершину , называют *расстоянием* от  до .

Определим весовую матрицу Ωчьи элементы  задаются формулой



Например, для графа на рис.13 матрица Ω имеет вид:



Матрица Ω является простым обобщением матрицы смежности вершин.

Пусть  ‑ ориентированный граф со взвешенными дугами. Обозначим через *s* ‑ вершину – начало пути, а через *t* ‑ вершину – конец пути.

В процессе работы алгоритма Дейкстры вершинам графа  приписываются числа (метки) , которые служат оценкой длины (веса) кратчайшего пути от вершины *s* к вершине . Если вершина  получила на некотором шаге метку , это означает, что в графе *G* существует путь из *s* в , имеющий вес .

Метки могут находиться в двух состояниях – быть временными или постоянными. Превращение метки в постоянную означает, что кратчайшее расстояние от вершины *s* до соответствующей вершины найдено.

***Ограничения алгоритма Дейкстры*** – веса должны быть положительными.

Алгоритм состоит из 2-х этапов:

**I этап** – нахождение длины кратчайшего пути;

**II этап** – построение кратчайшего пути от вершины *s* к вершине *t*.

***Алгоритм Дейкстры***

**I этап. Нахождение длины кратчайшего пути.**

***Шаг 1****.* Присвоение вершинам начальных меток.

Полагаем  и считаем эту метку постоянной (постоянные метки помечаются сверху звёздочкой). Для остальных вершин ,  полагаем  и считаем эти метки временными. Пусть , где  ‑ обозначение текущей вершины.

***Шаг 2****.* Изменение меток.

Для каждой вершины  с временной меткой, непосредственно следующей за вершиной , меняем её метку в соответствии со следующим правилом: .

***Шаг 3****.* Превращение метки из временной в постоянную.

Из всех вершин с временными метками выбираем вершину  с наименьшим значением метки

.

***Шаг 4****.* Проверка на завершение первого этапа.

Если , то ‑ длина кратчайшего пути от *s* до *t*. В противном случае происходит возвращение ко второму шагу.

**II этап. Построение кратчайшего пути.**

***Шаг 5****.* Последовательный поиск дуг кратчайшего пути.

Среди вершин, непосредственно предшествующих вершине  с постоянными метками, находим вершину , удовлетворяющую соотношению

.

Включаем дугу  в искомый путь и полагаем .

***Шаг 6****.* Проверка на завершение второго этапа.

Если , то кратчайший путь найден – его образует последовательность дуг, полученных на пятом шаге и выстроенных в обратном порядке. В противном случае возвращаемся к пятому шагу.

*Пример.* Задана весовая матрица Ω сети *G*. Найти минимальный путь из вершины  в вершину  по алгоритму Дейкстры.



По данной матрице весов граф изображён на рис.14:

*Рис. 14*



**

**



**



**6**

**11**

**5**

**4**

**8**

**7**

**9**

**9**

**6**

**6**

**6**

В данном графе есть цикл . Поэтому вершины графа нельзя упорядочить по алгоритму Фалкерсона.

Распишем подробно работу алгоритма Дейкстры.

**I этап. Нахождение длины кратчайшего пути.**

***Шаг 1****.* , 

, , , , .

***Первая итерация.***

***Шаг 2****.* Множество вершин непосредственно следующих за  с временными метками . Пересчитываем временные метки этих вершин:

;

;

.

***Шаг 3****.* Одна из временных меток превращается в постоянную:





Следовательно, .

***Шаг 4****.* , происходит возвращение на второй шаг.

***Вторая итерация.***

***Шаг 2****.* Множество вершин непосредственно следующих за  с временными метками . Пересчитываем временные метки этих вершин:

;

;

.

***Шаг 3****.* Одна из временных меток превращается в постоянную: 



Следовательно, .

***Шаг 4****.* , происходит возвращение на второй шаг.

***Третья итерация.***

***Шаг 2****.* Множество вершин непосредственно следующих за  с временными метками . Пересчитываем временные метки этих вершин:

;

***Шаг 3****.* Одна из временных меток превращается в постоянную: 



Следовательно, .

***Шаг 4****.* , происходит возвращение на второй шаг.

***Четвёртая итерация.***

***Шаг 2****.* Множество вершин непосредственно следующих за  с временными метками . Пересчитываем временные метки этих вершин:

;

***Шаг 3****.* Одна из временных меток превращается в постоянную: 



Следовательно, .

***Шаг 4****.* , происходит возвращение на второй шаг.

***Пятая итерация.***

***Шаг 2****.* Множество вершин непосредственно следующих за  с временными метками . Пересчитываем временные метки этих вершин:

;

***Шаг 3****.* Одна из временных меток превращается в постоянную: 



Следовательно, .

***Шаг 4****.* , конец первого этапа.

**II этап. Построение кратчайшего пути.**

***Первая итерация.***

***Шаг 5****.* Составим множество вершин непосредственно предшествующих вершине  с постоянными метками . Проверим для этой вершины выполнение

;

.

Включаем дугу  в искомый путь и полагаем .

***Шаг 6****.* , следовательно. возвращаемся к пятому шагу.

***Вторая итерация.***

***Шаг 5****.* Составим множество вершин непосредственно предшествующих вершине  с постоянными метками . Проверим для этой вершины выполнение

;

;

.

Включаем дугу  в искомый путь и полагаем .

***Шаг 6****.* , завершение второго этапа.

Кратчайший путь образует последовательность дуг .

**п.6. Алгоритм нахождение максимального пути**

Для нахождения максимального пути граф G (сети) должен быть ациклическим, ибо в противном случае может оказаться, что длины некоторых путей не ограничены сверху. Если G – ациклический граф, то для любых двух его вершин  выполняется одно из трёх условий:

1.  предшествует , .

2.  следует за , .

3. Нет пути между  и .

Из-за требуемой ацикличности графа первое и второе условия одновременно не выполнимы.

Перед вычислением максимального пути в орграфе необходимо упорядочить вершины графа по алгоритму Фалкерсона. Сам алгоритм вычисления максимального пути чисто перечислительный. Он перебирает все возможные пути от текущей вершины до всех последующих, достижимых из текущей вершины.

Пусть  ‑ длина максимального пути от вершины  до вершины , тогда величина  удовлетворяет следующим рекуррентным соотношениям:

 (1)

Соотношения (1) позволяют легко вычислить длины максимальных путей от  до вершин, достижимых из вершины *s*. Сами пути могут быть построены методом последовательного возвращения (второй этап в алгоритме Дейкстры).

*Пример.* Граф G задан матрицей весов Ω. Найти длину максимального пути из вершины  в  и сам этот путь.



По данной матрице весов построен граф на рис.15. Этот граф ациклический, поэтому возможно упорядочение его вершин по алгоритму Фалкерсона. Сделаем это графическим способом.

*Рис. 15*



**

**



**



**4**

**3**

**5**

**8**

**8**

**9**

**6**

**7**

**7**

**6**

***Графический способ упорядочивания вершин графа (алгоритм Фалкерсона****)*

**1**. Вершина -не содержит входящих дуг, следовательно, отнесём её к первой группе.

**2**. Вычёркиваем вершину  и все дуги, исходящие из . Получим граф на рис.16.

*Рис. 16*

**

**



**



**3**

**5**

**8**

**8**

**9**

**7**

**7**

**6**

В полученном графе опять находим вершины, в которые не заходит ни одна дуга. Это вершина . Отнесём её ко второй группе.

**3**. Вычёркиваем вершину  и все дуги, исходящие из . Получим граф на рис.17.

*Рис. 17*

**



**



**3**

**5**

**8**

**9**

**7**

В полученном графе опять находим вершины, в которые не заходит ни одна дуга. Это вершина . Отнесём её к третьей группе.

**4**. Вычёркиваем вершину  и все дуги, исходящие из . Получим граф на рис.18.

*Рис. 18*



**



**3**

**5**

**7**

В полученном графе опять находим вершины, в которые не заходит ни одна дуга. Это вершина . Отнесём её к четвёртой группе.

*Рис. 19*





**3**

**5**. Вычёркиваем вершину  и все дуги, исходящие из . Получим граф на рис.19.

Находим вершины, в которые не заходит ни одна дуга. Это вершина . Отнесём её к пятой группе.

*Рис. 20*



**6**. Вычёркиваем вершину  и все дуги, исходящие из . Получим граф на рис.20.

Находим вершины, в которые не заходит ни одна дуга. Это вершина . Отнесём её к шестой группе

Изоморфный граф с упорядоченными вершинами представлен на рис.21.

*Рис. 21*













**6**

**4**

**8**

**8**

**7**

**3**

**9**

**5**

**7**

**6**

**Этап I**

,

,

,

,

,

.

Итак, длина максимального пути из  в  равна 30.

**Этап II**

***Для*** : ;

.

Включаем дугу  в максимальный путь.

***Для*** : ;

;

.

Включаем дугу  в максимальный путь.

***Для*** : ;

.

Включаем дугу  в максимальный путь.

***Для*** : ;

.

Включаем дугу  в максимальный путь.

***Для*** : .

Включаем дугу  в максимальный путь.

Искомый путь таков:

