7 Графы

Графическая интерпретация различных моделей графов дана на рис. 1. В виде ориентированных графов можно представить логическую или функционально-логическую схему. На графовых моделях можно, например, оценить быстродействие схемы. Блок- схема алгоритма может быть представлена вероятностным графом для оценки временных характеристик алгоритма, затрат процессорного времени. Графом типа "дерево" можно отобразить практически любую структуру организации или предприятия.

Широкое применение теория графов получила при исследовании проблемы оптимизации, возникающей при конструировании больших как технических, так и программных систем, например, таких, как компиляторы.

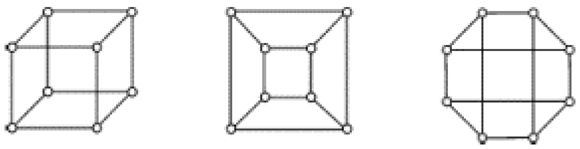


Рисунок 1 – Примеры моделей графа

Определение графа. Для описания строения различных систем, состоящих из связанных между собой элементов, часто используют графические схемы, изображая элементы точками (кружками, прямоугольниками и т.д.), а связи между ними – ориентированными или неориентированными линиями, соединяющими элементы. При этом получаются диаграммы, подобные представленной на рис 2.

На таких диаграммах важно лишь то, какие именно пары элементов соединены линиями, а не способ изображения элементов, форма или длина линий. Эту же структуру можно описать, не прибегая к графическому изображению, а просто перечислив пары связанных между собой элементов.

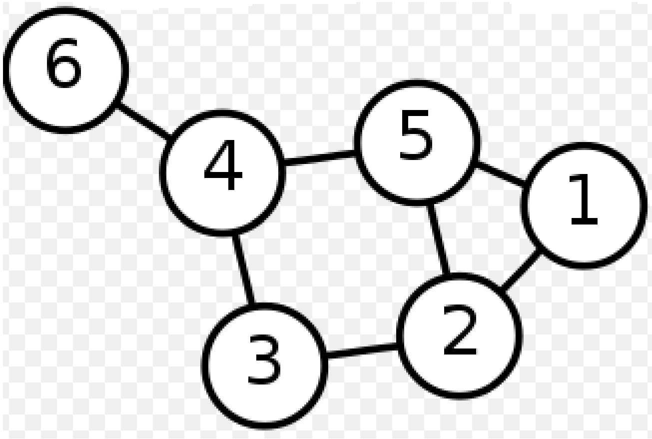


Рисунок 2 – Пример представления графа

Определение. Обыкновенным графом называется пара G = (V,E), где V - конечное множество, E - множество неупорядоченных пар различных элементов из V. Элементы множества V называются вершинами графа, элементы множества E - его ребрами.

Слегка модифицируя это определение, можно получить определения других типов графов без **кратных** ребер: если заменить при описании элементов множества E слово "неупорядоченных" словом "упорядоченных", получится определение ориентированного графа без петель, если убрать слово "различных", получится определение графа с петлями. Ориентированный граф часто называют орграфом.

В дальнейшем термин "граф" мы будем употреблять в смысле "обыкновенный граф", а рассматривая другие типы графов, будем специально это оговаривать.

Множество вершин графа G будем обозначать через VG, множество ребер - EG, числовершин - n(G), число ребер – m(G).

Термин "граф" неоднозначен, это легко заметить, сравнивая приводимые в разных книгах определения. Однако во всех этих определениях есть кое-что общее. В любом случае граф состоит из двух множеств - множества вершин и множества ребер, причем для каждого ребра указана пара вершин, которые это ребро соединяет. Вершины и ребра называются элементами графа. Будем рассматривать только конечные графы, то есть такие, у которых множества вершин и ребер конечны. Чтобы получить законченное определение графа того или иного типа, необходимо уточнить еще три момента.

Рассмотрим виды графов.

Ориентированный граф (кратко орграф) – (мульти) граф – это граф, рёбрам которого присвоено направление. Направленные рёбра именуются также дугами, а в некоторых источниках и просто рёбрами.

Ориентированный или неориентированный? Зададим ребро парой конечных вершин ребра. Прежде всего, договоримся, считаем ли мы пары (a, b) и (b, a) различными. Если да, то говорят, что рассматриваются упорядоченные пары (порядок элементов в паре важен), если нет - неупорядоченные. Если ребро соединяет вершину a с вершиной b и пара (a, b) считается упорядоченной, то это ребро называется ориентированным, вершина a - его начало, вершина b - конец. Если же эта пара считается неупорядоченной, то ребро называется неориентированным, а обе вершины – его концами. Чаще всего рассматривают графы, в которых все ребра имеют один тип – либо ориентированные, либо неориентированные. Соответственно и весь граф называют ориентированным или неориентированным. На рисунках ориентацию ребра (направление от начала к концу) указывают стрелкой.

Связный граф – граф, содержащий ровно одну компоненту связности. Это означает, что между любой парой вершин этого графа существует как минимум один путь.

Граф называется односвязным (связным), если:

1. У него одна компонента связности.
2. Существует путь из любой вершины в любую другую вершину
3. Существует путь из заданной вершины в любую другую вершину
4. Содержит связный подграф, включающий все вершины исходного графа
5. Содержит в качестве подграфа дерево, включающее все вершины исходного графа(такое дерево называется остовным)
6. При произвольном делении его вершин на 2 группы всегда

существует хотя бы 1 ребро, соединяющее пару вершин из разных групп

Орграф называется сильно связным, или сильным, если для двух любых различных его вершин xi и xj существует, по крайней мере, один путь, соединяющий эти вершины. Это определение означает также, что любые две вершины сильно связного графа взаимодостижимы. Пример данного графа показан на рис. 3а.

Орграф называется односторонне связным, или односторонним, если для любых двух различных его вершин xi и xj существует, по крайней мере, один путь из xi в xj или из xj в xi или оба пути существуют одновременно. Пример односторонне связного графа приведен на рис. 3б.

Орграф называется слабо связным, или слабым, если для любых двух различных вершин графа существует по крайней мере один маршрут, соединяющий их. Пример слабосвязного графа – рис. 3в.

Орграф называется несвязным, если для некоторой пары вершин орграфа не существуетмаршрута, соединяющего их (рис. 3г).

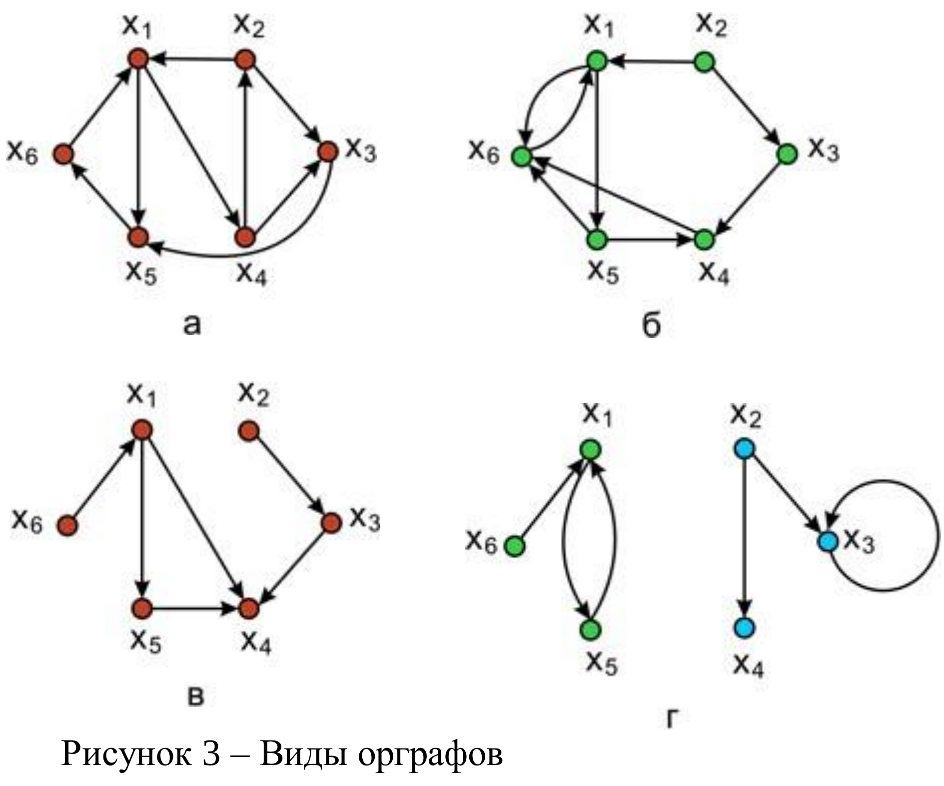


Рисунок 3 – Виды орграфов

Связный граф, не имеющий циклов, либо граф, в котором каждая пара вершин соединенаодной и только одной простой цепью, называется **деревом.**

Степень вершины x (*degree)* в теории графов — количество рёбер графа, которым принадлежит (инцидентна) вершина x.

Полустепень захода в орграфе для **вершины** x – число дуг, входящих в вершину. **Полустепень исхода** в орграфе для вершины x– число дуг, исходящих из вершины.

Взвешенные графы

Если графы используются для **моделирования** реальных систем, их вершинам, или ребрам, или и тем, и другим приписываются некоторые числа. Природа этих чисел может быть разнообразна. Например, если граф представляет собой модель железнодорожной сети, то число, приписанное ребру, может указывать длину перегона между двумя станциями, или наибольший вес состава, который допустим для этого участка пути, или среднее число поездов, проходящих через этот участок в течение суток и т.п. Сложилась традиция называть эти числа весами, а граф с заданными весами вершин и /или ребер - взвешенным графом.

Иногда дугам графа сопоставляют числа ai -> ci , называемые весом или длиной, или стоимостью или ценой. В каждом конкретном случае выбирается то слово, которое ближеподходит по смыслу задачи.

Граф G, описываемый тройкой вида G = (X, A, C),

где X = { xi }, i =1, 2, 3, ..., n – множество вершин,

A = { ai }, i = 1, 2, 3, ..., m – множество дуг,

C = {Ci}, i = 1, 2, 3, ..., m – множество характеристик дуг, называется графом со взвешенными дугами.

Пример такого графа приведен на рис. 4.

Граф со взвешенными вершинами – это граф, описываемый тройкой G = (X, A, V),

где X = { xi }, i = 1, 2, ..., n – множество вершин графа;

A = { ai }, i = 1, 2, ..., m – множество дуг графа;

V ={ vi }, i = 1, 2, ..., n – множество характеристик вершин.

В качестве характеристик вершин могут выступать "стоимость", "мощность", "вес" и т. п. Для графа со взвешенными вершинами в случае представления пути последовательностью вершин весом пути является сумма весов, входящих в этот путь вершин.

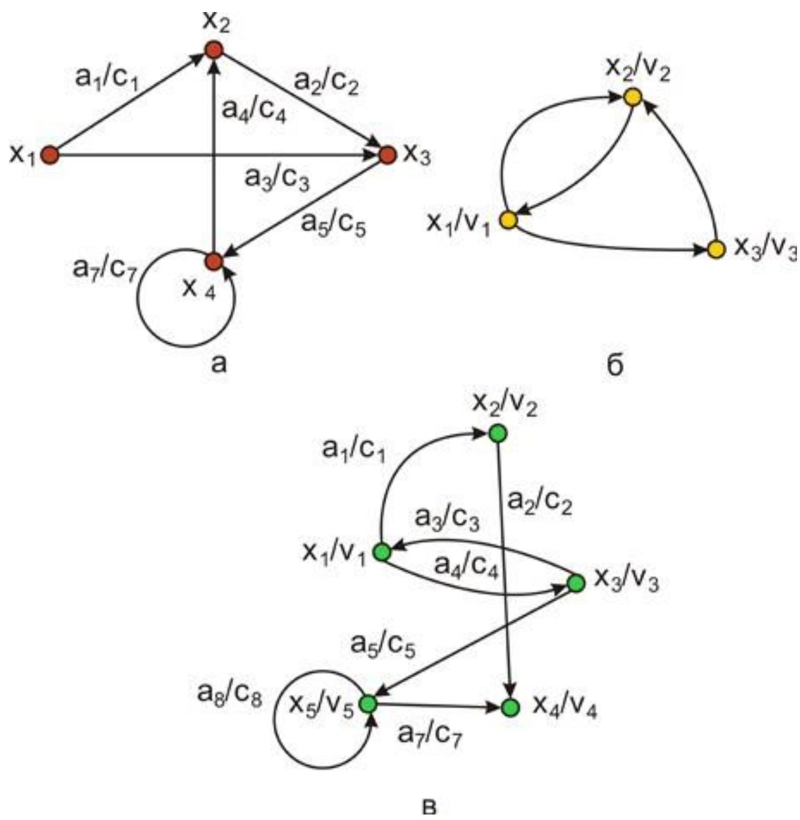


Рисунок 4 – Взвешенные графы: а – граф со взвешенными дугами; б – граф со взвешенными вершинами; в – взвешенный граф

И наконец, взвешенный граф определяется четверкой вида G = (X, A, V, C), т. е. и дуги, и вершины этого графа имеют некоторые характеристики.

Областями применения взвешенных графов в качестве моделей являются транспортные задачи, задачи оптимизации сети связи и системы перевозок и др. Одной из известнейших оптимизационных задач является нахождение кратчайших путей в графе со взвешенными дугами.

Понятие пути в графе. Путем в орграфе называется последовательность дуг, в которой конечная вершина всякой дуги, кроме последней, является начальной вершиной следующей дуги.

Например, для графа на рис 5 последовательности дуг

M1: a6, a5, a9, a8, a4 ,

M2: a1, a6, a5, a9, a7 ,

M3: a1, a6, a5, a9, a10, a6, a4

являются путями. Пути могут быть различными.

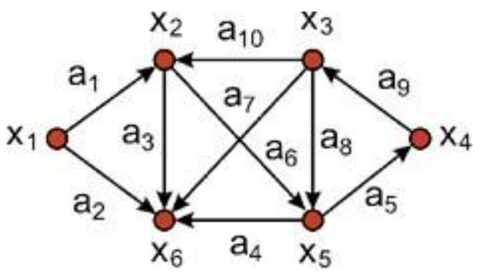


Рисунок 5 – Орграф

Маршрутом в орграфе называют чередующуюся последовательность вершин и *дуг,* вида $v0\{v0,v1\}v1\{v1,v2\}v2...vn$ (вершины могут повторяться). **Длина маршрута** — количество дуг в нем.

Путь - это *маршрут* в орграфе без повторяющихся дуг, **простой путь** — без повторяющихся вершин. Если **существует** путь из одной вершины в другую, то вторая вершина **достижима** из первой.

Контур есть замкнутый *путь*.

Для **полумаршрута снимается** ограничение на направление дуг, аналогично определяются **полупуть** и **полуконтур**.

Вес и длина пути. Если дугам графа сопоставлены числа ai -> ci , называемые весом или длиной, или стоимостью или ценой, то при рассмотрении пути M, представленного последовательностью дуг (a1, a2, ..., aq), за его вес (или длину, или стоимость) принимается число L(M), равное сумме весов всех дуг, входящих в путь, т. е. L(M) = Σci для всех ci ϵ_ M.

Длиной (или мощностью) пути называется число дуг, входящих в него. Чаще всего термин "длина" употребляется, когда все дуги, входящие в путь, имеют веса, равные 1, т.е. когда вес пути совпадает с его длиной (мощностью).

Каркасы. Пусть G - обыкновенный граф. Его каркасом называется остовный подграф, в котором нет циклов, а области связности совпадают с областями связности графа G. Таким образом, каркас связного графа - дерево, а в общем случае - лес.

У любого графа есть хотя бы один каркас. Действительно, если в G нет циклов, то он сам является собственным каркасом. Если же циклы есть, то можно удалить из графа любое ребро, принадлежащее какому-нибудь циклу. Такое ребро не является **мостом** (мост – это ребро графа, удаление которого увеличивает число компонент связности), поэтому при его удалении области связности не изменятся. Продолжая действовать таким образом, после удаления некоторого количества ребер получим остовный подграф, в котором циклов уже нет, а области связности - те же, что у исходного графа, то есть этот подграф и будет каркасом. Можно сказать, сколько ребер необходимо удалить для получения каркаса. Если в графе n вершин, m ребер и k компонент связности, то в каркасе будет тоже n вершин и k компонент связности. Но в любом лесе с n вершинами и k компонентами связности имеется ровно n-k ребер. Значит, удалено будет m-n+k ребер. Это число называется цикломатическим числом графа и обозначается через v(G).

Способы задания графов. Рассмотрим наиболее часто использующиеся способы заданияграфов.

Теоретико-множественное **представление** ***графов.*** Граф описывается перечислением множества вершин и дуг. Пример описания для орграфа приведен на рис 6: ***G4 = (X, A),*** где ***X =*** {xi}, i = 1, 2, 3, 4 – множество вершин; A = {ai }, i = 1, 2, ..., 6 – множество дуг, причем A = {(x1, x2), (x4, x2), (x2, x4), (x2, x3), (x3, x3), (x4 , x1)}.

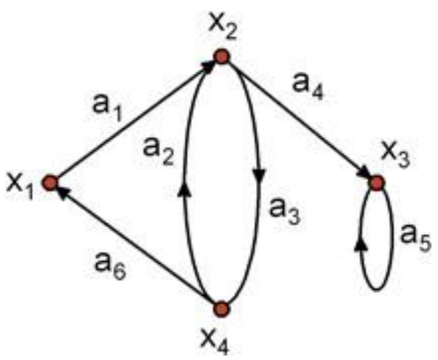


Рисунок 6 – Орграф

Задание графов соответствием**.** Описание графов состоит в задании множества вершин X и соответствия Г, которое показывает, как между собой связаны вершины**.**

Соответствием Г называется отображение множества X в X, а граф в этом случаеобозначается ***парой*** $G = (X, Г)$.

Отображением вершины xi — Г(xi) является множество вершин, в котором существуютдуги из вершины xi, т. е. Г(xi)={xj :_дуга (xi ,xj) ϵ A}.

Так для орграфа на рис. 6 описание заданием множества вершин и соответствия выглядитследующим образом:

G4=(X, Г),

где X = {xi}, I = 1, 2, ..., 4 – множество вершин,

Г(x1) = { x2 }, Г(x2) = { x3, x4 }, Г(x3) = {x3 }, Г(x4) = { x1, x2 } – отображения.

Для неориентированного или смешанного графов предполагается, что соответствие Г задает такой эквивалентный ориентированный граф, который получается из исходного графа заменой каждого неориентированного ребра двумя противоположно направленными дугами, соединяющими те же самые вершины.

Матричное представление графов. Для компьютерной обработки графы удобнопредставлять в виде матриц смежности и инциденций.

Матрица смежности – это квадратная матрица размерностью n x n, (где n – число вершинграфа), однозначно представляющая его структуру.

A = {aij}, i, j = 1, 2, ..., n, а каждый элемент матрицы определяется следующим образом:

aij = 1, если существует дуга (xi, xj),

aij = 0, если нет дуги (xi, xj).

Матрица инциденций представляет собой прямоугольную матрицу размером n x m, где n – количество вершин графа, а m – количество дуг графа. Обозначается матрица инциденций B = {bij}, i = 1, 2, ..., n, j = 1, 2, ..., m .

Каждый элемент матрицы определяется следующим образом:

$b_{ij} = 1$, если x_i является начальной вершиной дуги a_j,

$b_{ij} = –1$, если x_i является конечной вершиной дуги a_j,

$b_{ij} = 0$, если x_i не является концевой вершиной дуги a_j или если a_j является петлей.

На рис. 7 (а, б) приведен граф и его матрица смежности, по которой можно найти характеристики вершин. Так сумма элементов i -ой строки матрицы дает полустепень исхода вершины x_i, а сумма элементов i -го столбца дает полустепень захода вершины x_i. По матрице смежности можно найти прямые и обратные отображения. Рассмотрим i -ю строку матрицы. Если элемент $a_{ij}=1$, то элемент графа x_j входит в отображение $Г(x_i)$.

Например, во 2-й строке матрицы А (рис 7,б) единицы стоят в 2-м и 5-м столбцах, следовательно, $Г(x_2) = \{x_2,x_5\}$.

Для графа на рис. 7а матрица инциденций приведена на рис. 7в. Поскольку каждая дуга инцидентна двум различным вершинам, за исключением того случая, когда дуга образует петлю, то каждый столбец либо содержит один элемент равный 1 и один – равный – 1, либо все элементы столбца равны 0.

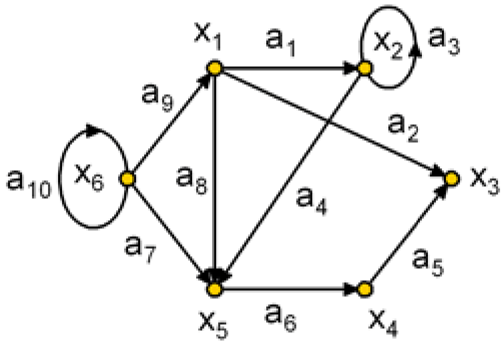
Для неориентированного графа, матрица инциденций определяется так же, за исключением того, что все элементы, равные –1, заменяются на 1.

Обходы графа. Существует множество алгоритмов на графах, в основе которых лежит такой систематический перебор вершин графа, что каждая вершина просматривается (посещается) в точности один раз.

Под обходом графов (поиском на графах) понимается процесс систематического просмотра всех ребер или вершин графа с целью отыскания ребер или вершин, удовлетворяющих некоторому условию.

При решении многих задач, использующих графы, необходимы методы регулярного обхода вершин и ребер графов. К стандартным и наиболее распространенным методамотносятся:

- поиск в глубину (Depth First Search, DFS);
- поиск в ширину (Breadth First Search, BFS).



а

A=	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	1	1	0	1	0
x_2	0	1	0	0	1	0
x_3	0	0	0	0	0	0
x_4	0	0	1	0	0	0
x_5	0	0	0	1	0	0
x_6	1	0	0	0	1	1

б

A=	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
x_1	1	1	0	0	0	0	0	1	-1	0
x_2	-1	0	0	1	0	0	0	0	0	0
x_3	0	-1	0	0	-1	0	0	0	0	0
x_4	0	0	0	0	1	-1	0	0	0	0
x_5	0	0	0	-1	0	1	-1	-1	0	0
x_6	0	0	0	0	0	0	1	0	1	0

в

Рисунок 7 – Орграф и его матричное представление: а – орграф; б – матрица смежности; в – матрица инциденций

Эти методы чаще всего рассматриваются на ориентированных графах, но они применимы и для неориентированных, ребра которых будем считать двунаправленными. Алгоритмы обхода в глубину и в ширину используются при решении различных задач обработки графов, например, построения остовного леса, проверки связности, ацикличности, вычисления расстояний между вершинами и других.

Поиск в глубину. При поиске в глубину посещается первая вершина, затем выполняется переход по ребрам графа, до тупика. Вершина графа является тупиковой, если все смежные с ней вершины уже посещены. После попадания в тупиковую вершину выполняется возврат по пройденному пути, пока не будет обнаружена вершина, у которой есть соседние, еще не посещенные вершины, а затем выполняется движение в этом новом направлении. Процесс оказывается завершенным при таком возвращении в начальную вершину, при котором все смежные с ней вершины уже посещены.

Алгоритм обхода в глубину состоит из трех основных шагов.

Шаг 1. Всем вершинам графа присваивается значение не посещенная. Выбирается первая вершина и помечается как посещенная.

Шаг 2. Для последней помеченной как посещенная вершины выбирается смежная вершина, являющаяся первой помеченной как не посещенная, и ей присваивается значение посещенная. Если таких вершин нет, то берется предыдущая помеченнаявершина.

Шаг 3. Повторять шаг 2 до тех пор, пока все вершины не будут помечены как посещенные(рис. 8).

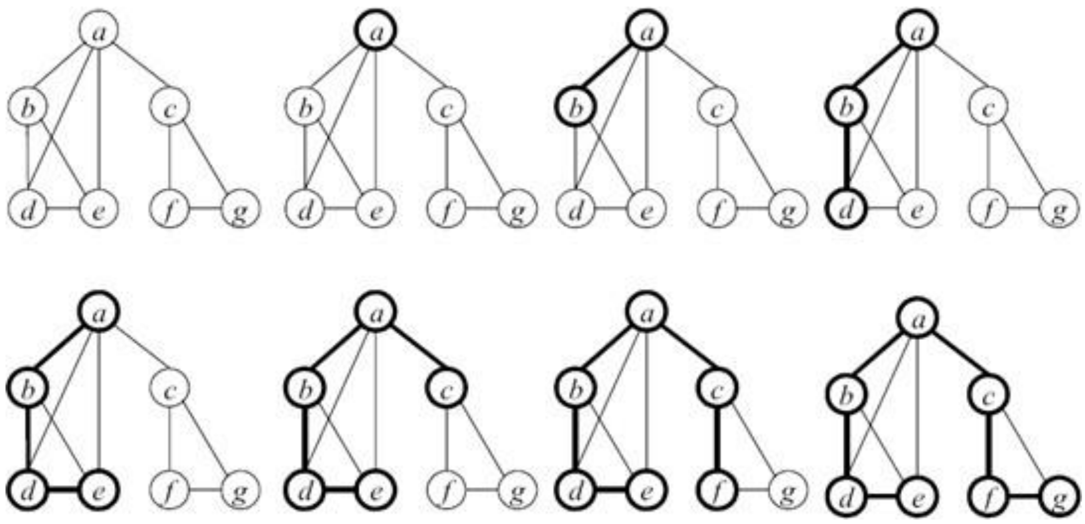


Рисунок 8 – Иллюстрация схемы «обход в глубину»

Временная сложность зависит от представления графа. Если используется матрица смежности, то временная сложность равна $O(n^2)$, при нематричном представлении графа сложность –O(n+m), рассматриваются все вершины и все ребра.

Обход в ширину. При обходе в **ширину**, после посещения первой вершины, посещаются все соседние с ней вершины. Потом посещаются все вершины, находящиеся на расстоянии двух ребер от начальной. На каждом новом шаге посещаются вершины, расстояние от которых до начальной на единицу больше предыдущего. Чтобы предотвратить повторное посещение вершин, необходимо вести список посещенных вершин. Для хранения временных данных, необходимых для работы алгоритма, используется очередь – упорядоченная последовательность элементов, в которой новые элементы добавляются в конец, а старые удаляются из начала.

При использовании этого алгоритма обхода для каждой вершины сразу находится длинакратчайшего маршрута от начальной вершины.

Алгоритм обхода в ширину также состоит из трех основных шагов.

Шаг 1. Всем вершинам графа **присваивается** значение не посещенная. Выбирается первая вершина и помечается как посещенная (и заносится в очередь).

Шаг 2. Посещается первая вершина из очереди (если она не помечена как посещенная). Все соседние с ней вершины заносятся в очередь. После этого она удаляется из очереди.

Шаг 3. Повторяется шаг 2 до тех пор, пока очередь не пуста (рис. 9).

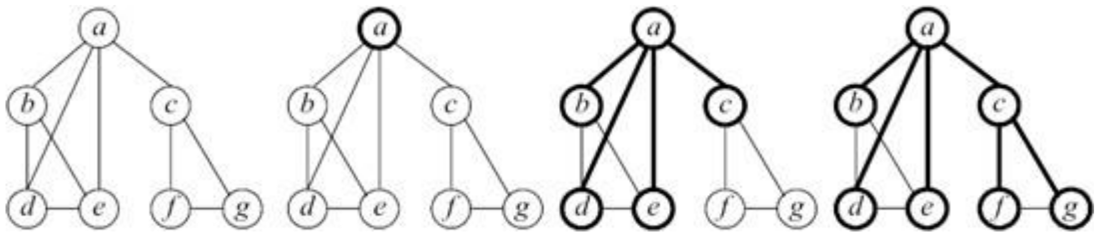


Рисунок 9 – Иллюстрация схемы «обход в ширину»

Не трудно обосновать, что сложность поиска в ширину при нематричном представлениияграфа равна O(n+m), так как рассматриваются все n вершин и m ребер. Если граф задан матрицей смежности сложность поиска оценивается как O(n^2).