Тема 3.2. Операции над графами. Маршруты. Количество маршрутов

Аннотация: Матричное представление графов. Матрица смежности. Матрица инцидентности. Связность графа. Матрица связности. Выделение компонент связности. Операции над графами. Маршруты. Связность. Поиск в ширину и глубину. Теорема о числе маршрутов заданной длины.

Операции над графами

Граф $H=\langle V',E'\rangle$ называется ***подграфом*** графа $G=\langle V,E\rangle$ если $V'\subseteq V$ и $E'=E\cap\left(V'\right)^2$.

Рассмотрим некоторые основные операции, производимые над графами.

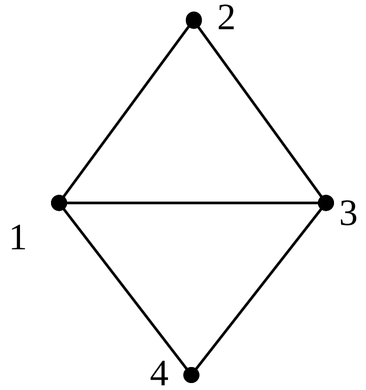
Операцией ***добавления*** к графу $G=\langle V,E\rangle$ ***вершины*** a образуется граф $\langle V\cup\{a\},E\rangle$. Операция ***добавления дуги*** (a,b) к графу *G* состоит в образовании графа $\langle V\cup\{a,b\},E\cup\{(a,b)\}\rangle$.

Под операцией ***удаления дуги*** (a,b) из графа *G* понимается операция, заключающаяся в удалении пары (a,b) из множества дуг *E*, в результате получается граф. Операция ***удаления вершины*** *a* из графа *G* заключается в удалении вершины а вместе с инцидентными ей дугами/

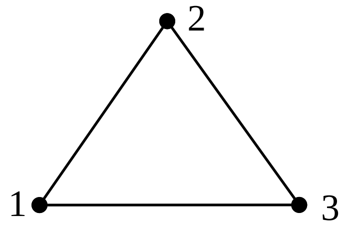
Отождествление (замыкание) вершин – при замыкании двух вершин, эти вершины удаляются из графа и заменяются одной новой, при этом ребра, инцидентные исходным вершинам, теперь будут инцидентны новой вершине. В случае, когда отождествляемые вершины *a* и *b* соединены дугой, операцию отождествления называют ***стягиванием дуги*** (a,b).

Примеры:

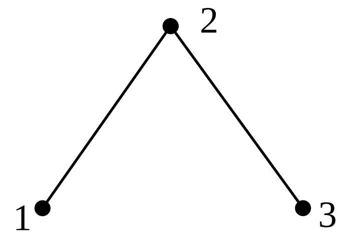
Рассмотрим графы:



G_1



G_2



G_3

- G_2 – подграф $G_{1.}$
- G_2 получается из G_1 удалением вершины 4.
- G_3 получается из G_2 удалением ребра (1,3).

Пусть $G_1 = \langle V_1, E_1 \rangle$, $G_2 = \langle V_2, E_2 \rangle$ - графы. Объединением $G_1 \cup G_2$ графов G_1 и G_2 называется граф $\langle V_1 \cup V_2, E_1 \cup E_2 \rangle$.

Если $V_1 \cap V_2 \neq \varnothing$, то пересечением $G_1 \cup G_2$ графов G_1 и G_2 называется граф $\langle V_1 \cap V_2, E_1 \cap E_2 \rangle$.

Маршруты в графах

Маршрутом от вершины v_H к вершине v_K в графе G называется всякая последовательность вида $v_H = v_{0'}, e_1, v_{1'}, e_2, \ldots, e_n, v_{n'} = v_K$ (в дальнейшем, не теряя общности, штрихи в номерах будем опускать), где e_k – ребро, соединяющее вершины v_{k-1} и v_k, $k = 1, 2, \ldots, n$. В случае орграфа v_{k-1} – начало дуги e_k, а v_k – ее конец. При этом вершину v_H называют ***началом маршрута***, а вершину v_K – его ***концом.*** В маршруте некоторые вершины и ребра могут совпадать.

Путь или маршрут часто указывают, перечисляя его вершины:

$$v_H = v_0 \to v_1 \to v_2 \to \ldots \to v_{s-1} \to v_s \to v_K .$$

Также маршрут можно задавать последовательностью ребер $e_1, e_2, \ldots, e_n$.

Число ребер в маршруте называется его ***длиной***.

Тривиальным называется маршрут длины 0.

В дальнейшем будем обозначать маршрут от v_H к v_K следующим образом: (v_H, v_K), иногда также будем указывать промежуточные вершины.

Маршрут называется ***цепью***, если в нем нет совпадающих ребер, и ***простой цепью*** – если дополнительно нет совпадающих вершин, кроме, может быть, начала и конца цепи.

Если начало цепи (простой цепи) совпадает с ее концом, то такая цепь называется ***циклом*** **(*простым циклом*)**.

Граф без циклов называется *ациклическим*.

Теорема

Пусть $G = \langle V,\ E \rangle$ - граф . Если существует маршрут из вершины v_j в вершину v_j, тогда существует и простая соединяющая их цепь.

Доказательство:

➢ Воспользуемся методом от противного. Пусть маршрут из v_j в v_j не является простой цепью. Тогда существует по крайней мере одна вершина v_m, встречающаяся в нем не менее двух раз, и маршрут имеет вид

$$v_i \to v_{i_1} \to \ldots \to v_m \to v_{m+1} \to \ldots \to v_m \to \ldots \to v_i .$$

Удалив из маршрута последовательность ребер $v_{m+1} \to \ldots \to v_m$, снова получим маршрут из v_j в v_j.

Если при этом он не будет простой цепью, процедуру можно повторить. Так как число ребер в конечном маршруте конечно, процесс удаления ребер конечен. В результате получим простую цепь из v_j в v_j.

Теорема доказана.⮘

Граф G называется ***связанным***, если имеется маршрут между любыми его двумя различными вершинами.

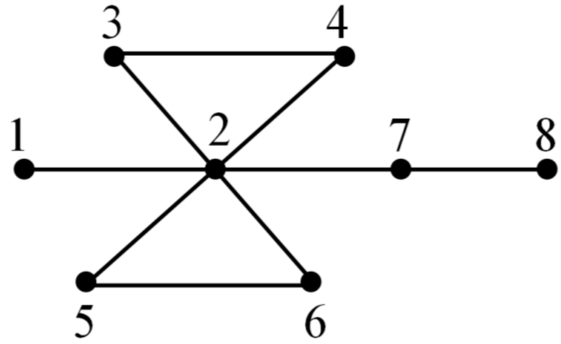
Теорема

Граф G является связным тогда и только тогда, когда между любыми двумя его вершинами существует простая цепь.

Доказательство: следует из предыдущей теоремы.

Полным называется граф, любые две вершины которого смежны.

Пример: Для графа



- (1,2,7,8) или (1,2,3,4) – простая цепь;
- (1,2,4,3,2,7,8) – цепь, не являющаяся простой;
- (3,4,2,5,6,2,4) – маршрут, не являющийся цепью;
- (3,4,2,3) – простой цикл;
- (3,4,2,5,6,2,3) – цикл, не являющийся простым.

Граф называется ***связным***, если любые две вершины в нем можно соединить маршрутом.

Легко видеть, что отношение связности на множестве вершин является отношением эквивалентности. Данное отношение разбивает множество вершин графа на классы, объединяющие вершины, которые можно связать друг с другом маршрутом. Такие классы называются ***компонентами связности***.

Компонента связности графа G — максимальный (по включению) связный подграф графа *G*. Другими словами, это подграф *H*, порожденный множеством вершин, в котором для любой пары вершин $u_i, u_j \in U$ в графе *H* существует (u_i, u_j) - цепь и для любой пары вершин $u_i \in U, u_j \notin U$ не существует (u_i, u_j) - цепи.

Виды алгоритмов обхода графа

Для выделения компонент связности можно использовать поиск в ширину или поиск в глубину. При этом затраченное время будет линейным от суммы числа вершин и числа ребер графа.

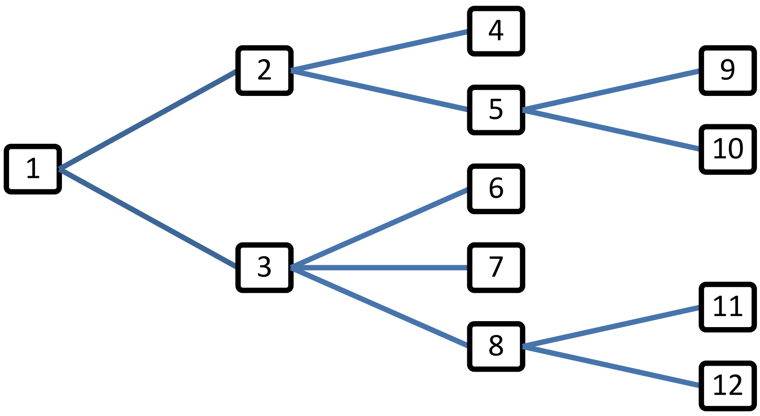
Алгоритм поиска в ширину

Поиск в ширину (англ. breadth-first search, BFS) — один из методов обхода графа. Пусть задан граф $G = \langle V, E \rangle$ и выделена исходная вершина v_0. Алгоритм поиска в ширину систематически обходит все ребра *G* для «открытия» всех вершин, достижимых из v_0, вычисляя при этом расстояние от v_0 до каждой достижимой из нее вершины. Алгоритм работает как для ориентированных, так и для неориентированных графов.

Замечание: Если длины ребер графа равны между собой, поиск в ширину является оптимальным, то есть всегда находит кратчайший путь. В случае взвешенного графа поиск в ширину находит путь, содержащий минимальное количество ребер, но не обязательно кратчайший.

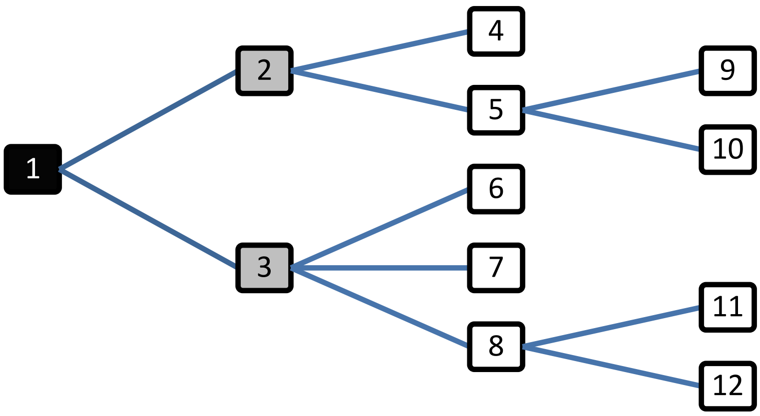
Пример: Для графа приведенного ниже произвести поиск в ширину:

Изначально все вершины белые. В процессе обхода каждая из вершин, по мере обнаружения, окрашивается сначала в серый, а затем в черный цвет. Определенный момент обхода описывает следующие условие: если вершина черная, то все ее потомки окрашены в серый или черный цвет.

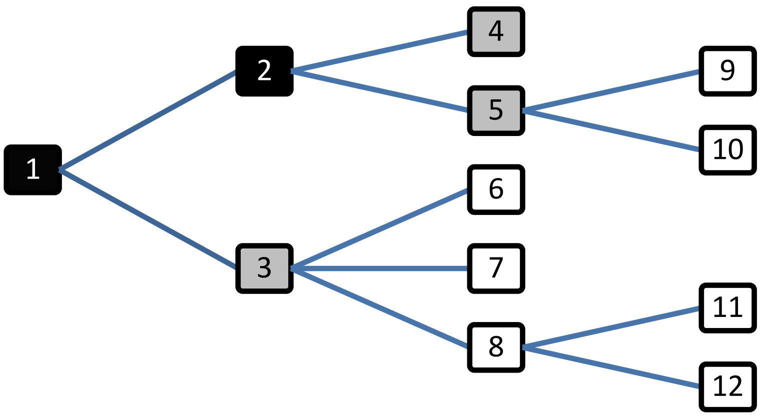


Решение:

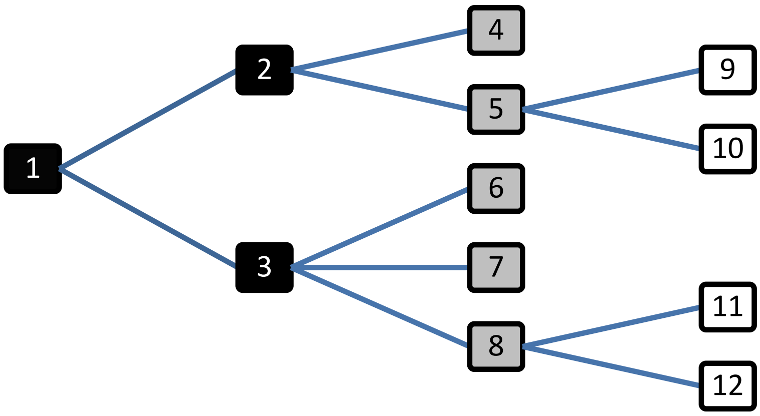
Шаг 1. Включаем вершину 1, просматриваем вершины 2 и 3:



Шаг 2. Включаем вершину 2 в очередь, просматриваем вершины 4 и 5:



Шаг 3. Включаем вершину 3 в очередь, просматриваем вершины 6, 7 и 8:



Шаг 4. Включаем вершину 4 в очередь.

Шаг 5. Включаем вершину 5 в очередь, просматриваем вершины 9 и 10 и т.д.

С помощью данного алгоритма можно выявлять структуру графа и вычислять его метрические характеристики.

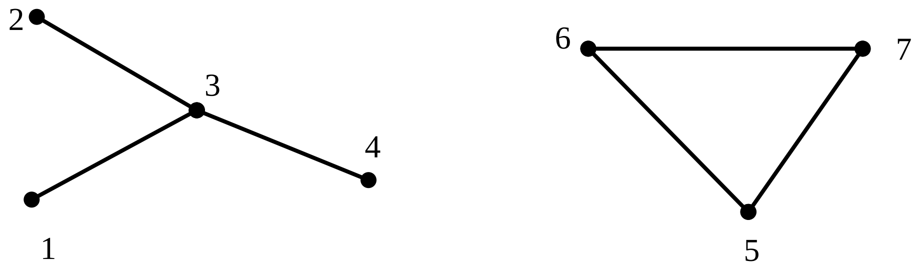
Поиск в ширину имеет такое название потому, что в процессе обхода мы идем вширь, т.е. перед тем как приступить к поиску вершин на расстоянии k+1, выполняется обход вершин на расстоянии k.

Алгоритм поиска в глубину

Поиск в глубину (англ. Depth-first search, DFS) — один из методов обхода графа. Стратегия поиска в глубину, как и следует из названия, состоит в том, чтобы идти «вглубь» графа, насколько это возможно. Алгоритм поиска описывается рекурсивно: перебираем все исходящие из рассматриваемой вершины ребра. Если ребро ведет в вершину, которая не была рассмотрена ранее, то запускаем алгоритм от этой нерассмотренной вершины, а после возвращаемся и продолжаем перебирать ребра. Возврат происходит в том случае, если в рассматриваемой вершине не осталось ребер, которые ведут в нерассмотренную вершину. Если после завершения алгоритма не все вершины были рассмотрены, то необходимо запустить алгоритм от одной из нерассмотренных вершин

Для ориентированных графов определено понятие ***компоненты сильной связности***. Ориентированный граф называется сильно связным, если любые две его вершины v_H и v_K сильно связны, то есть если существует ориентированный путь из v_H в v_K и ориентированный путь из v_K в v_H. Компонентами сильной связности орграфа называются его максимальные по включению сильно связные подграфы. Областью сильной связности называется множество вершин компоненты сильной связности.

Пример: Граф, приведенный на рисунке ниже имеет две компоненты связности с множеством вершин {1,2,3,4} и {5,6,7}.



Теорема

Любой граф представляется в виде объединения непересекающихся связных (сильных) компонент. Разложение графа на связные (сильные) компоненты определяется однозначно.

Примем теорему без доказательства.

Число маршрутов заданной длины в графе

Теорема

Если A – матрица смежности графа G, то элемент a_{ij} *матрицы* A^k *есть число маршрутов длины k из вершины* v_i *в вершину* v_j.

Доказательство:

➢ Применим метод математической индукции.

Для k=1 маршрут длины 1 как раз является ребром G, следовательно, результат теоремы при k=1 вытекает из определения матрицы смежности A.

Пусть теорема верна для $k-1$, докажем, что она верна и для k.

Рассмотрим $A^k = A^{k-1} \cdot A$. Элемент a_{ij} матрицы A^k равен

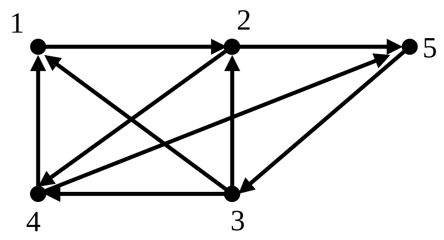
$$a_{ij} = \sum_s a_{is}^{(k-1)} \cdot a_{sj}^1,$$

где $a_{is}^{(k-1)}$ - элемент матрицы A^{k-1}, т.е. число маршрутов длины $k-1$ от вершины v_i до вершины v_s, а a_{sj}^1 - элемент матрицы смежности A, т.е. маршрут от v_s до v_j (один). В итоге имеем, что длина маршрута – k. Поскольку ведется суммирование по всем возможным s, то вычисляется количество всех возможных маршрутов длины k, в которых предпоследней вершиной является v_s (все возможные), т.е вычисляется число всех возможных маршрутов длины k из v_i в v_j. Теорема доказана. ≺

Следствие 1: В графе G ($|G| = n$), тогда и только тогда существует маршрут (v_i, v_j), когда элемент a_{ij} матрицы $A + A^2 + A^3 + \ldots + A^{n-1}$ не равен нулю.

Следствие 2: В графе G ($|G| = n$), тогда и только тогда существует цикл, содержащий вершину v_i, когда элемент a_{ii} матрицы $A + A^2 + A^3 + \ldots + A^{n-1}$ не равен нулю.

Пример: Дан орграф. Найти число маршрутов длины 2 из вершины № 2 в № 4, число маршрутов в графе длины 3:



Построим матрицу смежности данного графа:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Согласно теореме о числе маршрутов длины n их количество находится как A^n. Тогда, число маршрутов длины 2 и 3 соответственно:

$$A^2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}, \quad A^3 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 2 \\ 1 & 1 & 0 & 2 & 1 \\ 1 & 1 & 0 & 1 & 2 \end{pmatrix}.$$

По матрице A^2 найдем число маршрутов длины 2 из вершины № 2 в № 4 – это элемент a_{24}, т.е. 0.

Общее число маршрутов длины 3 – это сумма всех элементов матрицы A^3, т.е. 26.

Число маршрутов длины 4 находятся аналогично (по матрице A^4).

Ответ: число маршрутов длины 2 из вершины 2 в 4 равно 0, общее число маршрутов длины 3 – 26.

Вопросы для самоконтроля:

1. Матричное представление графов. Матрица смежности. Матрица инцидентности.
2. Связность графа. Матрица связности. Выделение компонент связности.
3. Операции над графами.
4. Маршруты.
5. Поиск в ширину и глубину.
6. Теорема о числе маршрутов заданной длины.