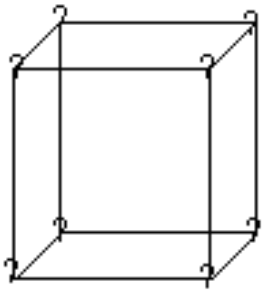
Практика 5-6 Работа с графами

Работа над ошибками по тесту 1

Задача 2

Дан кусок проволоки длиной 120 см. Можно ли , не ломая проволоку, изготовить каркас куба с ребром 10 см?



Куб – это граф, у которого все вершины имеют степень 3. Для того, чтобы быть эйлеровым, надо иметь все вершины четными. У нас это не так. Следовательно, граф не является эйлеровым. Обойти все его ребра , не проходя по одному ребру дважды невозможно., потому что число нечетных вершин больше 2.

(Эйлеров граф можно обойти, пройдя по каждому ребру только один раз в том случае, если граф связный и нечетных вершин у него 0 или 2.)

Теория

Граф определяется как множество узлов и множество ребер, в котором каждое ребро соединяет пару узлов, то есть представляет собой упорядоченную пару G=(V,E) множеств, где V – это вершины или узлы, а E – его ребра.

Степенью вершины в графе называется число выходящих из нее ребер.

Вершина графа, имеющего нечетную степень, называется нечетной, а имеющая четную степень – четной.

Чтобы найти количество ребер графа надо просуммировать степени вершин и полученный результат разделить на два.

В любом графе количество ребер вдвое меньше суммы степеней вершин.

Число нечетных вершин любого графа - четно. (по свойству четности суммы нескольких целых чисел. Эта сумма четна тогда и только тогда, когда среди чисел четное число нечетных.

Сумма степеней всех вершин графа всегда четна - она ровно вдвое больше числа ребер, а следовательно нечетных вершин всегда четное число.

Граф может быть ориентированным (орграф) и неориентированным. В ориентированном графе пути (дуги) имеют направление, а в неориентированном - не имеют.

Пути в неориентированном графе называются рёбрами.

Путь из вершины А в вершину В начинается в А и проходит по набору ребер до тех пор, пока не будет достигнута вершина В.

Связный граф – любые две его вершины можно соединить путем – непрерывной последовательностью ребер.

Эйлеров граф можно обойти, пройдя по каждому ребру только один раз в том случае, если граф связный и нечетных вершин у него 0 или 2.

Взвешенным называется такой граф, для каждого ребра (дуги) которого определена некоторая функция. Эта функция называется весовой. Иначе говоря, к каждому ребру «привязан» вес - фиксированное число или результат какой-то функции. Чаще всего встречается первый вариант.

Цикл – замкнутый путь в графе, в котором все вершины, кроме начальной и конечной попарно различны. Дерево – граф без циклов. (Им, например, является структура содержания книги).

Петлей называется дуга, ведущая из вершины в нее же.

Как хранить графы в памяти компьютера?
Каковы структуры данных для представления графов?

Проблема правильного хранения графа в памяти компьютера очень актуальна. От того, как храним граф в той или иной задаче, зависит и время выполнения задачи, и объем потребляемой памяти, то есть по сути дела – качество вашего алгоритма.

Давайте выясним, какие структуры данных используются для хранения графа в памяти компьютера.

Принято хранить информацию о графах либо в виде матрицы примыканий, либо в виде списка примыканий.

Матрица примыканий обеспечивает быстрый доступ к информации о ребрах графа. Это удобно, если в графе много ребер. В противном случае окажется много незаполненных клеток, что нерационально.

Длина списка примыканий пропорциональна числу ребер в графе, но использование списка примыканий увеличивает время получения информации о ребре.

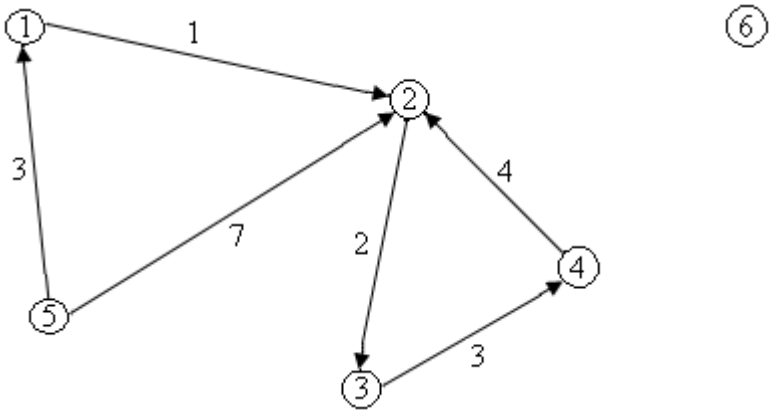
Поэтому в основе выбора одного из этих методов должен лежать предварительный анализ о тех графах, которые вы хотите обработать на компьютере.

Если в графе много вершин, но при этом каждая из них связана с небольшим количеством других вершин, вам выгоднее взять список примыканий.

Если число вершин в графе невелико, то лучше воспользоваться матрицей примыканий.

Матрица примыканий.

Пусть дан граф G=(V,E) с числом вершин |V|=N=6. Запишем его в виде двумерного массива размером N*N. В каждую ячейку этого массива запишем 0, за исключением тех случаев, когда из вершины v_i в вершину v_j ведет ребро. Тогда в ячейку запишем стоимость ребра, или единицу если стоимости не учитываются.



Граф G

Mat[i,j] = $\begin{cases} 1, если v_i v_j \in E \\ 0, если v_i v_j \notin E \end{cases}$ для всех i и j от 1 до N (1 перед “если”пишется для неориентированного графа и значение веса для ориентированного)

Составим матрицу смежности для нашего графа:

	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	2	0	0	0
3	0	0	0	3	0	0
4	0	4	0	0	0	0
5	3	7	0	0	0	0
6	0	0	0	0	0	0

в матрице смежности нам часто нужно хранить большое количество нулей. Например, в нашей матрице "нужных" значений только 6, а остальные 30 - нули, не представляющие для нас почти никакой нужной информации.

Если у нас много вершин и много ребер, то нам понадобится много памяти, что не всегда нас может устроить. Кроме того, сама работа будет недостаточно быстрой. Если же надо будет вывести не номера вершин, а номера ребер, то в матрице примыканий эти номера просто негде хранить, что приведет к целому ряду дополнительных неудобств.

Тогда прибегают к хранению графа в виде списка примыканий.

Список примыканий.

Следующий тип хранения графа в памяти компьютера - список дуг. Чаще всего это двумерный массив размером 3*E, в первой строке которого хранится информация, из какой вершины начинается дуга, во второй - в какой кончается, а в третьей строке - вес дуги. Опять же разберёмся на примере (все тот же граф):

	1	2	3	4	5	6
1	1	2	3	4	5	5
2	2	3	4	2	1	2
3	1	2	3	4	3	7

Мы чётко видим, что почти вся таблица заполнена "нужными" значениями, а не нулями. Это уже хорошо, значит, память экономится.

Как видно, этот способ, в отличие от матрицы смежности, хранит информацию о номере дуги.

Данный метод хранения графа наиболее часто используется при решении задач с большими ограничениями по памяти и по времени

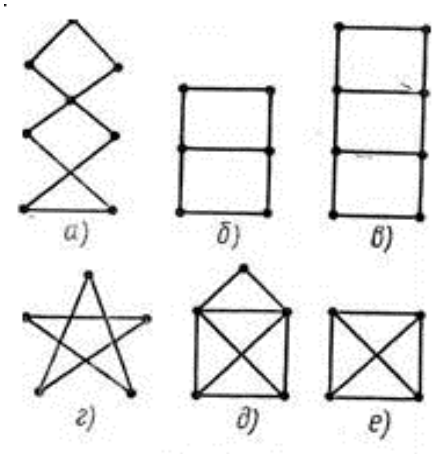
Замечание

Обход графа — это некоторое систематическое перечисление его вершин (и/или ребер). Наибольший интерес представляют обходы, использующие локальную информацию (списки смежности). Среди всех обходов наиболее известны поиск в ширину и в глубину. Алгоритмы поиска в ширину и в глубину лежат в основе многих конкретных алгоритмов на графах.

Задача 4 Можно ли обвести карандашом , не отрывая его от бумаги и не проводя по одной линии дважды, квадрат с диагоналями?

Теперь вы точно знаете, что этого сделать нельзя, Еще раз ответьте-почему? Как это связано с эйлеровыми графами?

Для практики рассмотрите фигуры а, б, в, г, д, е. Как ответить на вопрос 4 задачи для этих фигур?



Самостоятельная работа

Задача 1

Система точек, соединенных отрезками, называется связной, если из любой точки можно пройти в любую другую по этим отрезкам. Можно ли соединить пять точек в связную систему так, чтобы при стирании любого

отрезка образовались ровно две связные системы точек, не связанных друг с другом?

Задача2

Можно ли 77 телефонов соединить между собой проводами так, чтобы каждый соединялся ровно с пятнадцатью?

Задача 3

Сколько всего пятизначных чисел, у которых все цифры нечетные?

Задача 4

Сколько существует вариантов покупки одной розы, если продают 3 алые, 2 белые, 4 желтые розы?

Задача 5

1)Нарисуйте следующий граф:

G=({1,2,3,4,5,6}, {{1,2},{1,4},{2,5},{2,6},{3,4},{3,5},{3,6},{4,5},{4,6},{5,6}})

2) Выпишите матрицу примыканий для этого графа.
3) Выпишите список примыканий для этого графа.

Задача 6

Нарисуйте следующий ориентированный граф:

G=({1,2,3,4,5},{{1,2},{1,4},{1,5},{2,3},{2,4},{2,5},{3,2},{3,4},{3,5},{4,1},{4,2},{4,5},{5,2},{5,3},{5,4}}).

4) Выпишите матрицу примыканий для этого графа.
5) Выпишите список примыканий для этого графа.