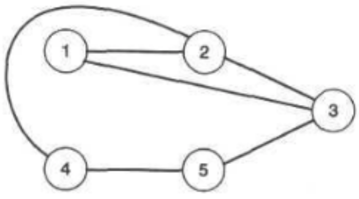
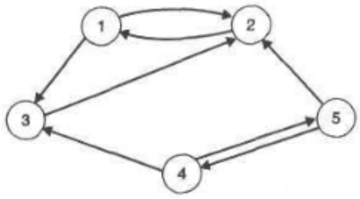
Представление графов

Примеры графов



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



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

С формальной точки зрения граф представляет собой упорядоченную пару *G = (V, E)* множеств, первое из которых состоит из вершин или узлов графа, а второе – из его ребер. Ребро связывает между собой две вершины. При работе с графами нас часто интересует, как проложить путь из ребер от одной вершины графа к другой. Поэтому мы будем говорить о движении по ребру; это означает, что мы переходим из вершины *A* графа в другую вершину *B*, связанную с ней ребром *AB* (ребро графа, связывающее две вершины для краткости обозначается этой парой вершин). В этом случае мы говорим, что *A* примыкает к *B* или что эти две вершины соседние.

Граф может быть ориентированным или нет. Ребра неориентированного графа, чаще всего называемого просто графом, можно проходить в обоих направлениях. В этом случае ребро – это неупорядоченная пара вершин, его концов (если концы ребра совпадают, то речь идет о петле – ребре, соединяющим вершину с самой собой). В ориентированного графе, или орграфе, ребра представляют собой упорядоченные пары вершин: первая вершина – это начало ребра, а вторая – его конец (начало и конец ребра в ориентированном графе также могут совпадать). Далее мы для краткости будем говорить просто о ребрах, а ориентированы они или нет будет понятно из контекста.

Полный граф – это граф, в котором каждая вершина соединена со всеми остальными. Число ребер в полном графе без петель с *N* вершинами равно $(N^2 – N)/2$. В полном ориентированном графе разрешается переход из любой вершины в любую другую. Поскольку в графе переход по ребру разрешается в обоих направлениях, а переход по ребру в орграфе – только в одном, в полном орграфе в два раза больше ребер, т.е. их число равно $N^2 – N$.

Путь в графе или орграфе – это последовательность ребер, по которым можно поочередно проходить. Другими словами, путь из вершины *A* в вершину *B* начинается в *A* и проходит по набору ребер до тех пор, пока не будет достигнута вершина *B*.

Граф называется связным, если всякую пару узлов можно соединить по крайней мере одним путем. Цикл – это путь, который начинается и кончается в одной и той же вершине. В ациклическом графе циклы отсутствуют. Связный ациклический граф называется (неукорененным) деревом. Структура неукорененного дерева такая же, что и у дерева, только в нем не выделен корень. Однако каждая вершина неукорененного дерева может служить его корнем.

Рассмотрим граф с $n = |V|$ вершинами $v_1, ..., v_n$.

Его матрицей смежности (adjacency matrix) называется (n x n) - матрица a, в которой

$$a_{ij} = \begin{cases} 1, \text{если есть ребро из } v_i \text{ в } v_j \\ 0 \quad\quad \text{в противном случае} \end{cases}$$

Ячейка матрицы смежности взвешенного графа содержит ∞, если соответствующее ребро отсутствует, а во всех остальных случаях ее значение равно весу ребра. Диагональные элементы такой матрицы равны 0, поскольку путешествие из вершины в нее саму не стоит ничего. Матрица смежности неориентированного графа, таким образом, симметрична. В таком представлении мы можем за время O(1) проверить, соединены ли данные вершины ребром (посмотрев на один элемент массива). В то же время хранение матрицы требует памяти O(n^2), что во многих случаях неэкономно.

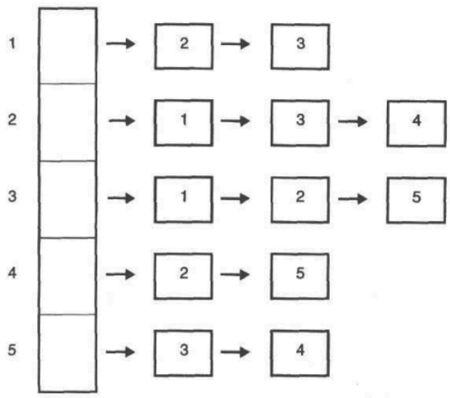
Примеры матрицы смежности для приведенных графов

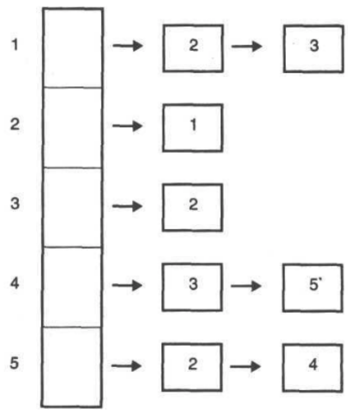
$$
\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & 0 & 1 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 0 & 0 \\
4 & 0 & 0 & 1 & 0 & 1 \\
5 & 0 & 1 & 0 & 1 & 0
\end{array}
\qquad
\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & 0 & 1 & 1 & 0 & 0 \\
2 & 1 & 0 & 1 & 1 & 0 \\
3 & 1 & 1 & 0 & 0 & 1 \\
4 & 0 & 1 & 0 & 0 & 1 \\
5 & 0 & 0 & 1 & 1 & 0
\end{array}
$$

Альтернативное представление — список смежности (adjacency list). Требуемая при этом память пропорциональна размеру графа (сумме числа вершин и числа рёбер). Элементами списка смежности являются связные списки, по одному для каждой вершины графа. Для вершины *u* в таком списке хранятся вершины, в которые ведут рёбра из *u*, то есть вершины *v*, для которых $(u, v) \in E$. Для ориентированного графа каждое ребро входит только в один из этих списков (для начальной вершины), а для неориентированного в два (для двух концов ребра). Список смежности, таким образом, требует памяти O(|V| + |E|). Проверка наличия ребра (u, v) теперь требует просмотра списка вершины *u* (что может быть больше O(1) шагов). Зато в этом представлении легко просмотреть всех соседей заданной вершины (что часто бывает необходимо). Список смежности для неориентированного графа симметричен (если *u* содержится в списке для *v*, то и *v* содержится в списке для *u*).

Элементы списка смежности для взвешенного графа содержат дополнительное поле, предназначенное для хранения веса ребра.

Примеры списков смежности для приведенных графов





Матрица смежности или список смежности?

Что лучше? Ответ зависит от отношения между числом вершин |V| и числом рёбер |E|. Заметим, что |E| может быть довольно малым — порядка |V | (если |E| сильно меньше |V |, то граф уже вырожденный — в частности, содержит изолированные вершины). Или же довольно большИм - порядка $|V|^2$ (если граф содержит все возможные рёбра). Графы с большим числом рёбер называют плотными (dense), с малым — разреженными (sparse).

Выбирая алгоритм, полезно понимать, с какими графами ему в основном придётся иметь дело. Важно это и при хранении: если хранить веб-граф, в котором больше восьми миллиардов вершин, в виде матрицы смежности, то занимать она будет миллионы терабайтов, что сравнимо с общей ёмкостью всех жёстких дисков в мире. И дальше, скорее всего, будет только хуже (матрица может расти быстрее, чем производство дисков).

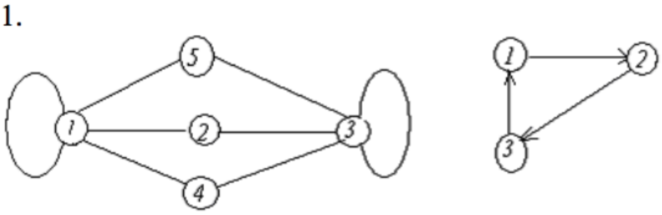
А вот хранить граф Интернета в виде списка смежности вполне разумно, поскольку хранить нужно несколько десятков миллиардов гиперссылок, каждая из которых будет занимать в списке всего несколько байтов, и такого размера диск поместится в карман. Такая разница происходит из-за того, что граф Интернета очень разрежен: страница содержит в среднем пару десятков ссылок на другие страницы—из нескольких миллиардов возможных.

2.1. Выполнение лабораторной работы

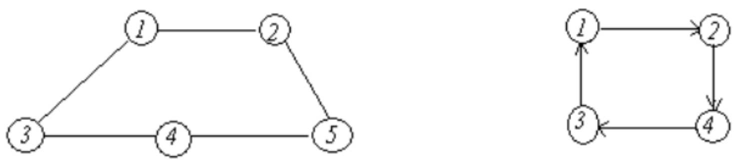
1. Согласно индивидуальному варианту построить матрицу смежности для неориентированного графа (1) и для ориентированного графа (2)
2. Согласно индивидуальному варианту построить список смежности для неориентированного графа (1) и для ориентированного графа (2)
3. Реализовать оба представления на любом выбранном языке программирования. Вывести на экран оба представления
4. Составить отчет в электронном виде.
5. Ответить на дополнительные вопросы преподавателя.

Варианты заданий

1.

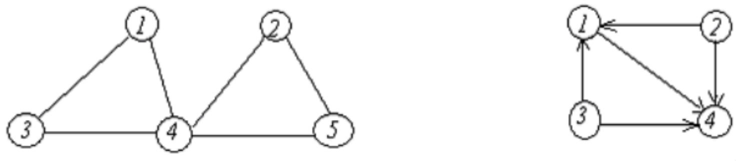


2.

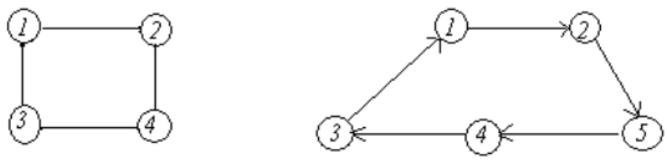


3.

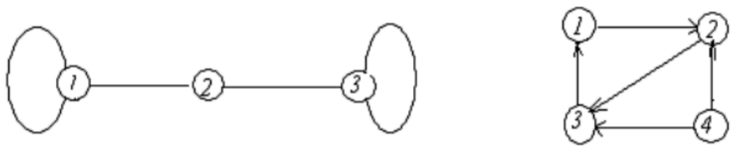
4.



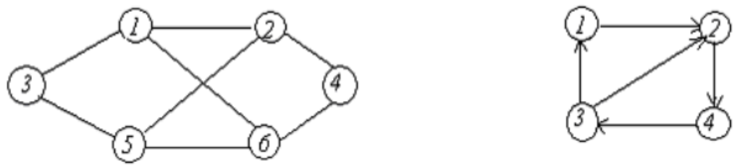
5.



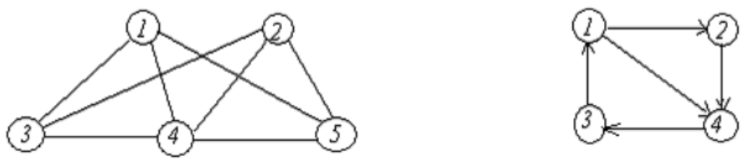
6.



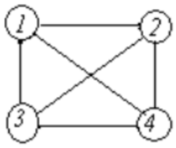
7.

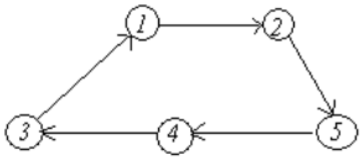


8.

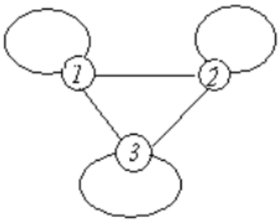


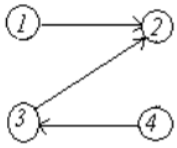
9.



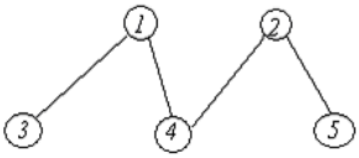


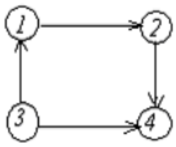
10.





11.





12.