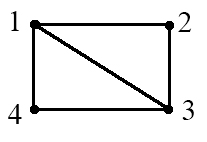
# 1. Машинное представление графов

Самый приятный и полезный для человека способ представить граф – изобразить его на плоскости в виде точек и соединяющих их линий. Очевидно, что для ЭВМ он совершенно бесполезен. Поскольку выбор для этой цели определенной структуры данных очень сильно влияет на эффективность алгоритма, рассмотрим несколько способов машинного представления графов.

**Матрицы смежности** – самый худший с алгоритмической точки зрения способ. Это матрицы *nm* в строках перечисляются вершины, в столбцах – ребра (см. рис. 1.3).

Матрица смежности

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1–2 | 1–3 | 1–4 | 2–3 | 3–4 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 1 | 1 |
| 4 | 0 | 0 | 1 | 0 | 1 |

  
Рис. 1.3. Модельный граф

Этот способ требует *nm* ячеек памяти, причем большинство из ячеек заняты нулями. Ответ на элементарный вопрос: «к каким вершинам ведут ребра из *x*?», требует перебора всех столбцов, т.е. *m* шагов.

**Матрица инциденций** имеет размер *n*×*n*. Если существует ребро (*x–y)*, то в клетках с координатами (*x, y*) и (*y, x*) ставится 1.

Матрица инциденций графа *G* (рис. 1.3):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | 0 | 1 | 1 | 1 |
| 2 | 1 | 0 | 1 | 0 |
| 3 | 1 | 1 | 0 | 1 |
| 4 | 1 | 0 | 1 | 0 |

Независимо от числа ребер требуется объем памяти *n*×*n*, но за один шаг можно ответить на вопрос: «существует ли ребро из *x* в *y*?».

Также граф можно представить в виде списка пар, соответствующих ребрам:

Список ребер графа *G* (рис. 1.3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 2 | 3 |
| 2 | 3 | 4 | 3 | 4 |

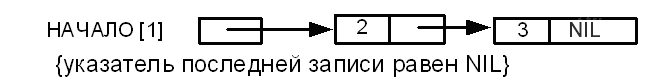
Объем памяти равен 2*m*, но попробуйте ответить на вопрос: «к каким вершинам ведут ребра из вершины *x*?».

Наиболее удобным машинным представлением графа являются **списки** **инцидентности.**

Каждой вершине *v**V* ставим в соответствие список вершин, соединенных ребрами с вершиной *v*. Для неориентированного графа каждое ребро (*v–u*) будет представлено дважды: в списке для *v* и в списке для *u*. Начало каждого списка будет храниться в одномерном массиве НАЧАЛО длины *n*. По вершине *v* находим НАЧАЛО[*v*] – в этой ячейке хранится указатель на начало списка вершин, соединенных с *v*. Каждый элемент такого списка – запись, состоящая из двух полей:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вершина | Указатель на следующую  запись |  | UZL | NEXT |

Если вершина 1 соединена с 2 и 3, то ее список выглядит как на рис. 1.4.

  
Рис. 1.4. Список инцидентности

Число ячеек памяти, необходимое для представления графа списками инцидентности, имеет порядок *O(n+m).*

В дальнейшем при записи алгоритмов список инцидентности для вершины *v* будем называть ЗАПИСЬ[*v*].

Цикл, перебирающий все элементы этого списка, будем условно записывать «for uЗАПИСЬ[v]».

***Вопрос*.** Изобразите списки инцидентности графа *G* (рис. 1.3). ***Ответ*.** См. рис. 1.5.

  
  
Рис. 1.5. Список инцидентности к графу с рис. 1.3