

Графы

Способы хранения

Аминев Б.Д.

October 13, 2017

Некоторые определения

Граф, степень вершины, ориентированность

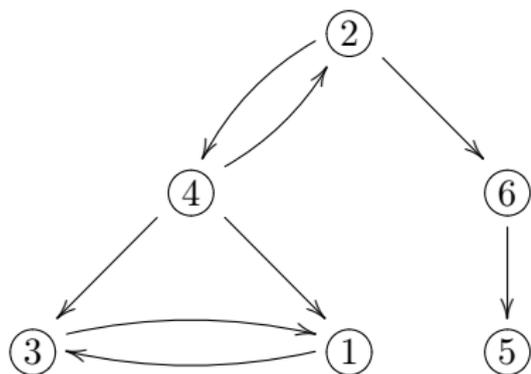
Граф абстрактный математический объект, представляющий собой множество вершин графа и набор рёбер, то есть соединений между парами вершин.

Степенью вершины называется количество рёбер, концом которых она является.

Ориентированный граф это граф, в котором у каждого ребра есть начало и конец.

Хранение графа

Матрица смежности. Представление (ориентированный граф)

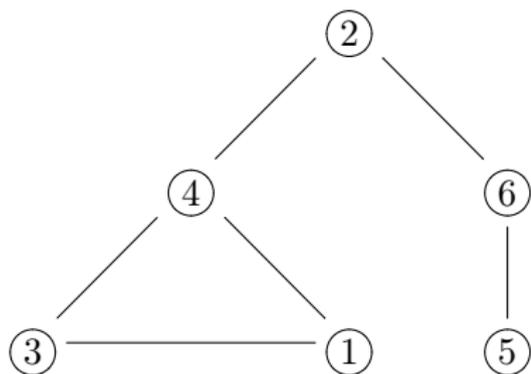


Представление

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

Хранение графа

Матрица смежности. Представление (неориентированный граф)



Представление

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

Хранение графа

Матрица смежности. Анализ

E (edges) — количество рёбер;

V (vertexes) — количество вершин.

Требования по памяти

$\sim V^2$.

Сложность основных операций

- Проверка связности двух вершин:
 ~ 1 .
- Поиск всех вершин, связанных с данной: $\sim V$.

Представление

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

Взвешенный граф

Храним веса вместо единиц в таблице.

Хранение графа

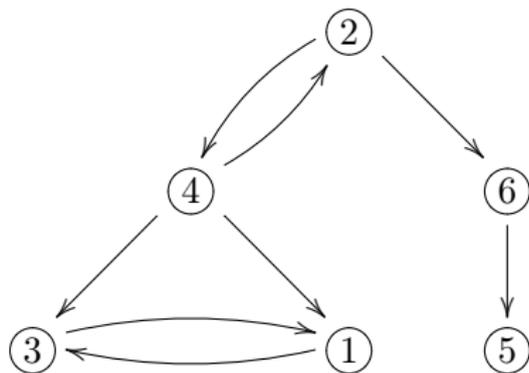
Матрица смежности. Реализация на Python

```
1 adjacency_matrix = [ [0, 0, 1, 0, 0, 0]
2                       , [0, 0, 0, 1, 0, 1]
3                       , [1, 0, 0, 0, 0, 0]
4                       , [1, 1, 1, 0, 0, 0]
5                       , [0, 0, 0, 0, 0, 0]
6                       , [0, 0, 0, 0, 1, 0]
7                       ]
num_vertices = len( adjacency_matrix )
```

adjacency_matrix.py

Хранение графа

Списки смежных вершин. Представление (ориентированный граф)



Представление

1 : 3

2 : 4, 6

3 : 1

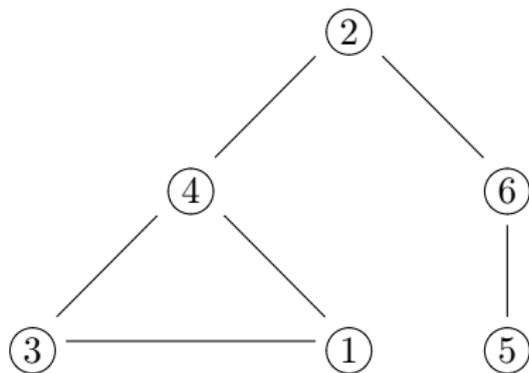
4 : 1, 2, 3

5 :

6 : 5

Хранение графа

Списки смежных вершин. Представление (неориентированный граф)



Представление

1 : 3, 4

2 : 4, 6

3 : 1, 4

4 : 1, 2, 3

5 : 6

6 : 2, 5

Хранение графа

Списки смежных вершин. Анализ

E (edges) — количество рёбер;

V (vertexes) — количество вершин.

Требования по памяти

$\sim E + V$.

Сложность основных операций

- Проверка связанности двух вершин: обычно \sim степени вершины, в худшем случае $\sim V$.
- Поиск всех вершин, связанных с данной: обычно \sim степени вершины, в худшем случае $\sim V$.

Представление

1 : 3

2 : 4, 6

3 : 1

4 : 1, 2, 3

5 :

6 : 5

Улучшения

Множества смежных вершин — быстрый поиск смежной вершины.

Хранение графа

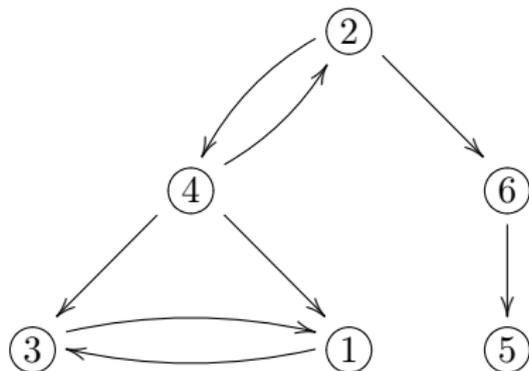
Списки смежных вершин. Реализация на Python

```
2 adjacency_list = [ [3]
                    , [4, 6]
                    , [1]
                    , [1, 2, 3]
                    , []
                    , [5]
                    ]
8 num_vertices = len( adjacency_list )
```

adjacency_list.py

Хранение графа

Список рёбер. Представление (ориентированный граф)



Представление

2, 4

4, 2

2, 6

4, 3

4, 1

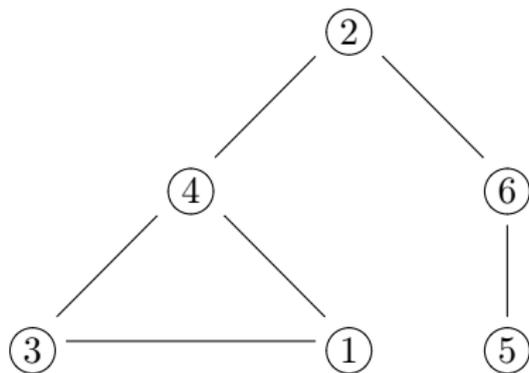
3, 1

1, 3

6, 5

Хранение графа

Список рёбер. Представление (неориентированный граф)



Представление

2, 4

2, 6

4, 3

4, 1

3, 1

6, 5

Хранение графа

Список рёбер. Анализ

E (edges) — количество рёбер;

V (vertexes) — количество вершин.

Представление

Требования по памяти

$\sim E$.

2, 4

4, 2

Сложность основных операций

- Проверка связанности двух вершин $\sim E$.
- Поиск всех вершин, связанных с данной $\sim E$.

2, 6

4, 3

4, 1

3, 1

Использование

В случаях, когда некоторое действие надо применить ко всем рёбрам.

1, 3

6, 5

Хранение графа

Список рёбер. Реализация на Python

```
num_vertices = 6
edges_list = [ [2, 4]
               , [4, 2]
               , [2, 6]
               , [4, 3]
               , [4, 1]
               , [3, 1]
               , [1, 3]
               , [6, 5]
               ]
```

edge_list.py