

Работа с графами и визуализация ЛАБОРАТОРНАЯ РАБОТА

<http://www.kimrt.ru>

Онлайн-курс

[Суперкомпьютерные технологии в задачах моделирования](http://www.kimrt.ru)

Граф — математическая абстракция реальной системы любой природы, объекты которой обладают парными связями. Граф как математический объект есть совокупность двух множеств — множества самих объектов, называемого множеством вершин, и множества их парных связей, называемого множеством рёбер. Элемент множества рёбер есть пара элементов множества вершин.

Графы находят широкое применение в современной науке и технике. Они используются и в естественных науках (физике и химии) и в социальных науках (например, социологии), но наибольших масштабов применение графов получило в информатике и сетевых технологиях.

В качестве простейшего примера из жизни можно привести схему перелётов определённой авиакомпании, которая моделируется графом, где вершинами графа являются города, а рёбрами — рейсы, соединяющие пары городов. Дерево каталогов в компьютере также является графом: диски, папки и файлы являются вершинами, а рёбра показывают вложенность файлов и папок в папки и диски. Строение Википедии моделируется ориентированным графом, в котором статьи — вершины графа, а гиперссылки — дуги (тематическая карта). Графы являются основным объектом изучения теории графов [1].

Способы представления графов включают в себя следующие подходы.

- Матрица смежности — таблица, где как столбцы, так и строки соответствуют вершинам графа. В каждой ячейке этой матрицы записывается число, определяющее наличие связи от вершины-строки к вершине-столбцу (либо наоборот).
- Матрица инцидентности — таблица, где строки соответствуют вершинам графа, а столбцы соответствуют связям (рёбрам) графа.
- Список смежности — список, где каждой вершине графа соответствует строка, в которой хранится список смежных вершин. Такая структура данных не является таблицей в обычном понимании, а представляет собой «список списков».
- Список ребер — список, где каждому ребру графа соответствует строка, в которой хранятся две вершины, инцидентные ребру.

<http://www.kimrt.ru>

Рассмотрим стандартную задачу поиска кратчайших путей между некоторыми городами. Данная задача моделируется взвешенным связным графом. Дано некоторое множество вершин, объединенных ребрами, имеющими заданные веса. В качестве примера рассмотрим следующий граф (рис. 1).

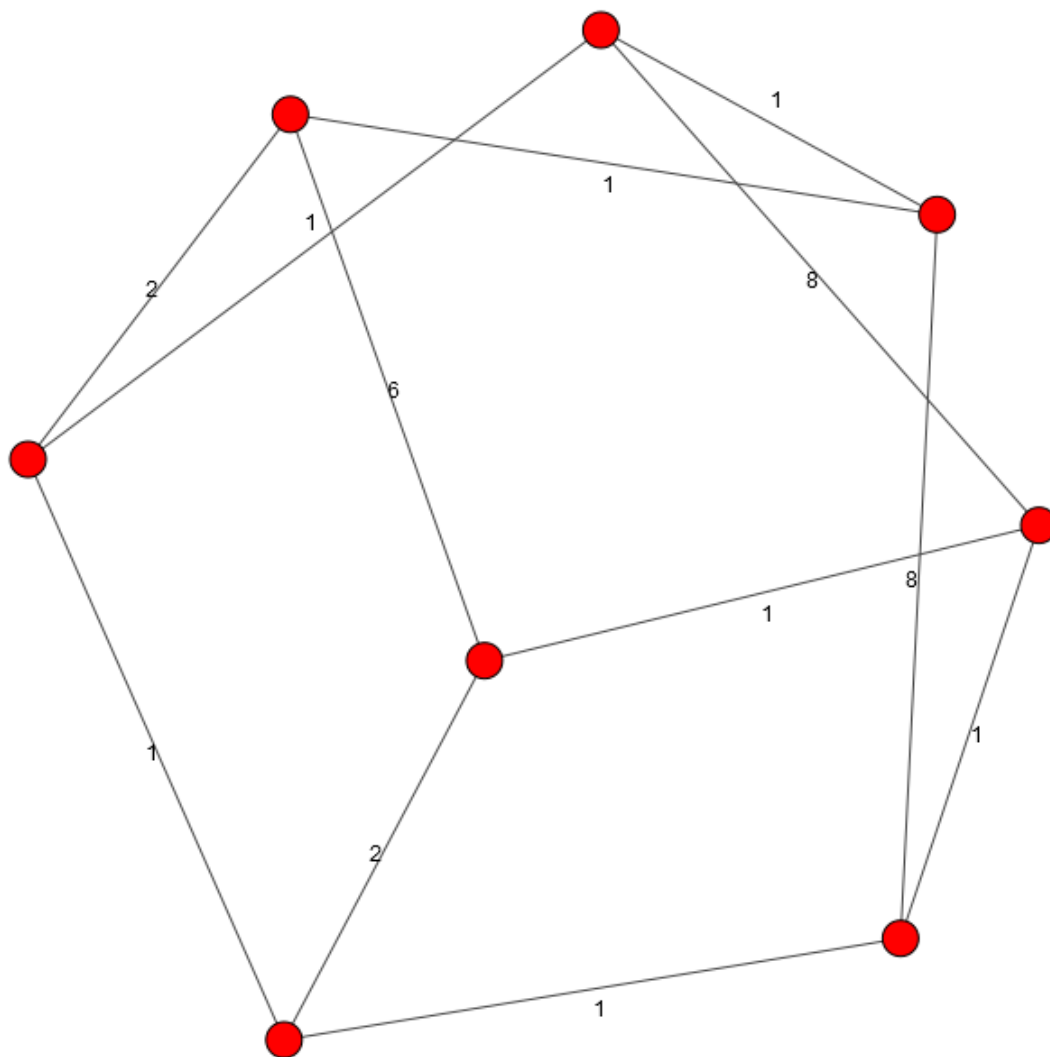


Рис. 1. Взвешенный неориентированный граф

В данном примере представлен взвешенный неориентированный связный граф. **Связный граф** — это граф, в котором из любой вершины можно построить маршрут в любую другую вершину. Изображение данного графа было получено с помощью библиотек *igraph* и *pycairo* на языке *python*. Для их установки нужно выполнить в терминале команды.

```
pip install igraph
```

```
pip install pycairo
```

В данном примере библиотека *pycairo* служит лишь для вывода изображения в файл. Библиотека *igraph* — это инструмент, имеющий множество функций для работы с графами. Задача о нахождении кратчайших путей из одной вершины до всех остальных вершин в ней решается одной строчкой [2].

```
short = g.get_all_shortest_paths(v, weights=w)
```

Здесь *g* — это заданный граф, *v* — это индекс вершины, от которой строятся кратчайшие пути, *w* — это список весов ребер графа [3]. Данная функция возвращает список путей. После получения данного списка вычислить расстояния до вершин не составит труда.

Задание. Реализуйте программу на языке *python*, позволяющую находить в заданном графе кратчайшие пути от заданной вершины до всех остальных вершин. После нахождения всех кратчайших путей программа должна вывести в файл изображение графа. В полученном изображении для всех вершин должны быть выведены расстояния до них.

http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. Граф (математика). [https://ru.wikipedia.org/wiki/Граф_\(математика\)](https://ru.wikipedia.org/wiki/Граф_(математика))
2. Python-igraph API reference.
https://igraph.org/python/doc/api/igraph.igraph.GraphBase.html#get_all_shortest_paths
3. Самостоятельная работа. Визуализация графов средствами языка Python и библиотеки *igraph*.
http://www.kimrt.ru/courses/stm/self_work/igraph_library.pdf