Введение в теорию графов. Задачи поиска пути в графе.

Хайрулин Сергей Сергеевич
s.khayrulin@gmail.com

Overview

- Изоморфизм графов.
- Способы задания графов.
- Матрицы смежности и инцидентности, их свойства.
- Задача о кратчайшем пути.
- Постановка задачи.
- Обзор алгоритмов.
- Алгоритм Флойда-Уоршелла
- Алгоритм Форда-Беллмана.
- Алгоритм Дейкстры
- Алгоритм Дейкстры для разреженных графов
- Алгоритм поиска A*.

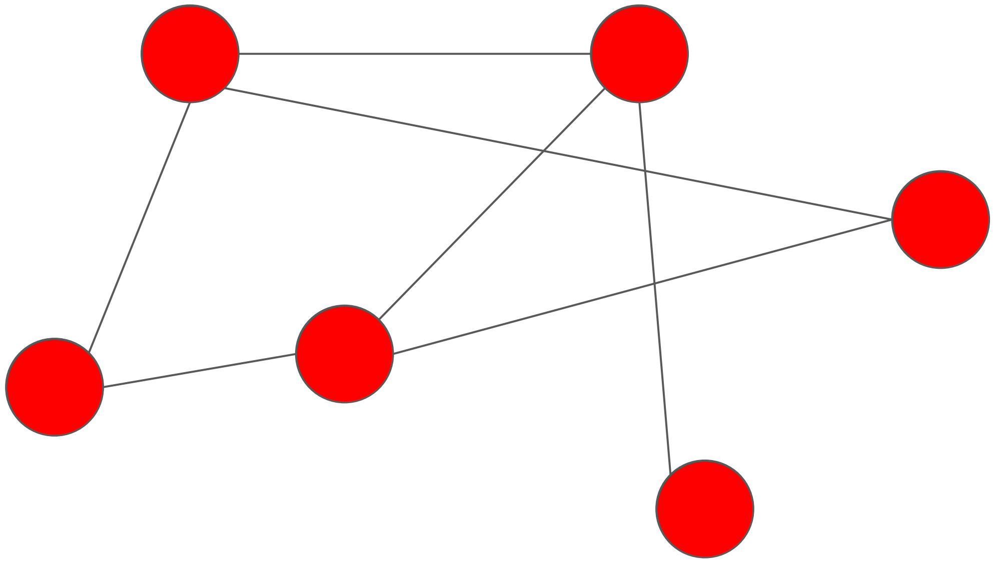
Литература и др. источники

- Дональд Эрвин Кнут. Искусство программирования (Том 1, 2, 3) // Вильямс 2015.
- Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман. Структуры данных и алгоритмы // Вильямс 2000.
- Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И. Лекции по теории графов // М.: Наука, 1990.
- Харари Ф. Теория графов // М.: Мир, 1973.
- Косточка А. В. Дискретная математика. Часть 2 //Новосибирск: НГУ, 2001.
- Котов В. Е., Сабельфельд В. К. Теория схем программ // Наука 1991.
- http://algolist.manual.ru
- ….

Основные понятия теории графов

Граф состоит из конечного множества вершин и V и некоторого множества E неупорядоченных пар вершин называемые ребра. И обычно обозначается $G:=\langle V, E\rangle$

Основные понятия теории графов



Маршрут/путь в графе

- *Маршрут* в графе G — это такая последовательность $R = v_0, e_1, v_1, e_2, \ldots, e_n, v_n$ вершин и ребер в G, что ребро e_i соединяет вершины v_i и v_{i+1} при всех $0 \le i \le n - 1$.

  Если $v_0 \neq v_n$, то говорят, что R *соединяет* вершины v_0 и v_n; если к тому же все вершины в R различны, то он называется *путем*

Связный граф

- Граф G называется *связным*, если каждая пара его вершин соединена путем (или, что равносильно, маршрутом).

  Маршрут R, в котором $v_0 = v_n$, называется *замкнутым*, а если при этом он не имеет других повторяющихся вершин, то называется *циклом*. Легко видеть, что любой минимальный по включению замкнутый маршрут есть цикл.

Инцидентность

- Ребро (u, v) и вершина v *инцидентны*, а *степенью* $\deg(v)$ вершины v в графе называют число ребер, инцидентных v. Вершина степени 0 называется *изолированной*, а вершина степени 1 — *висячей*.

  Для графа G через $V\ (G)$ и $E(G)$ обозначим множества вершин и ребер G, соответственно. При работе с графами часто полезны следующих два факта.

Лемма о рукопожатиях. Лемма о цикле.

- **Лемма о рукопожатиях.** *Сумма степеней вершин любого графа G равна удвоенному числу его ребер,*

$$\sum_{v \in V(G)} \deg(v) = 2|E(G)|$$

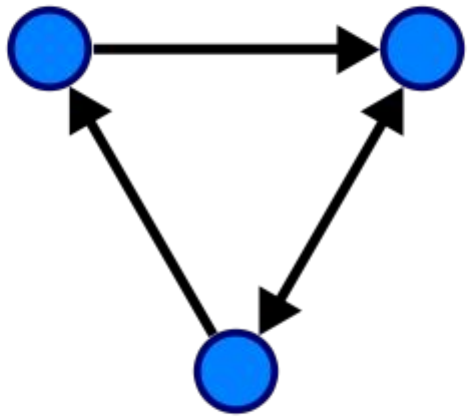
Лемма о цикле. *Если в графе есть хотя бы одно ребро, но нет изолированных и висячих вершин, то в нем есть цикл.*

Полный граф. Клика

- Полный граф — граф, в котором для каждой пары вершин v_1 и v_2, существует ребро, инцидентное v_1 и инцидентное v_2 (каждая вершина соединена ребром с любой другой вершиной).

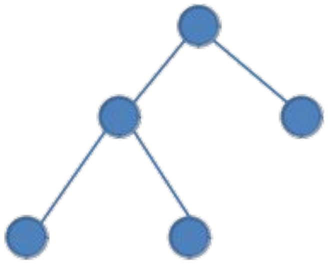
  Клика — подмножество вершин графа, полностью соединённых друг с другом, то есть подграф, являющийся *полным графом*.

Ориентированный граф



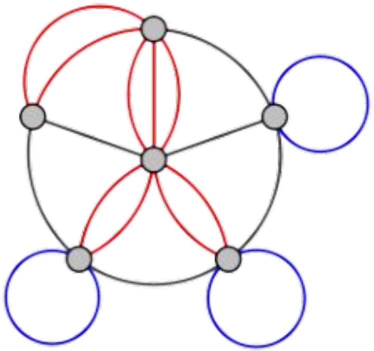
Деревья

Связный граф без циклов называется *деревом*.



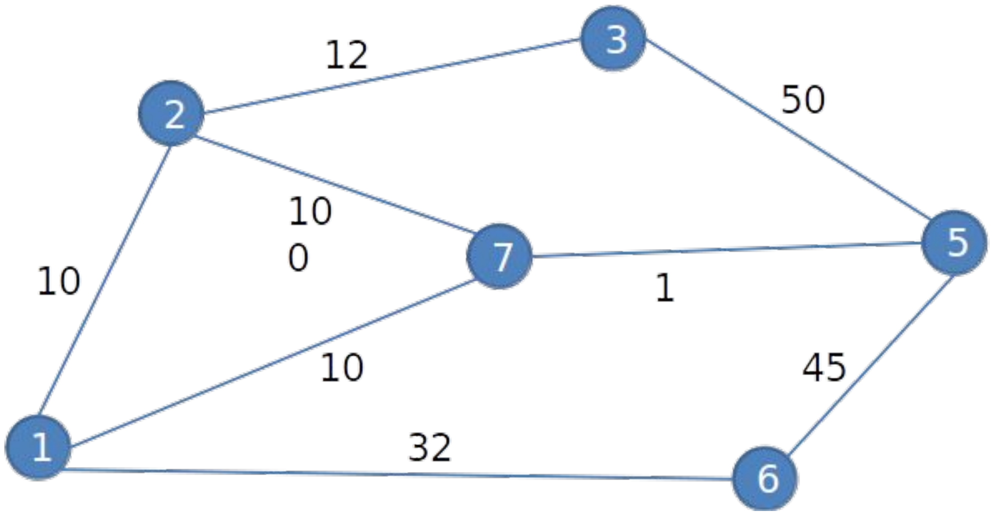
Мультиграф

Мультиграф — граф, в котором может быть пара вершин, которая соединена более чем одним ребром (ненаправленным), либо более чем двумя дугами противоположных направлений.

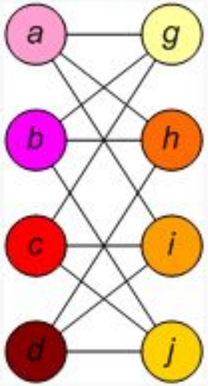
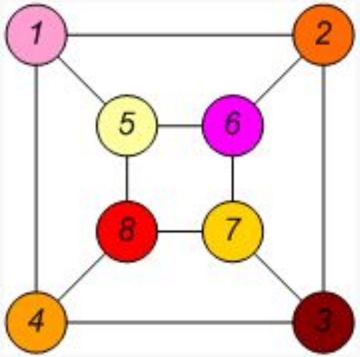


Взвешенный граф

Взвешенный граф — граф, каждому *ребру* которого поставлено в соответствие некое значение (*вес ребра*).



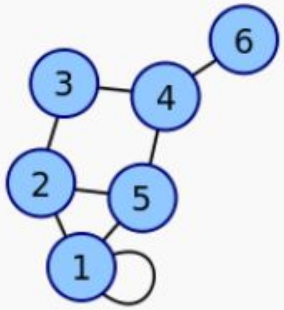
Изоморфизм графов

Граф G	Граф H	Изоморфизм между графами G и H	Подстановка изоморфизма f
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$	$\begin{pmatrix} a & b & c & d & g & h & i & j \\ 1 & 6 & 8 & 3 & 5 & 2 & 4 & 7 \end{pmatrix}$

WIKI

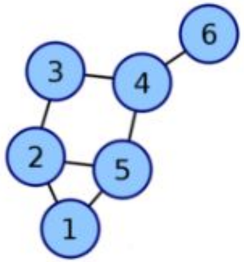
Способы задания графов

- **Матрица смежности** G с конечным числом вершин n (пронумерованных числами от 1 до n) — это квадратная матрица A размера n, в которой значение элемента a_{ij} равно числу рёбер из i-й вершины графа в j-ю вершину.

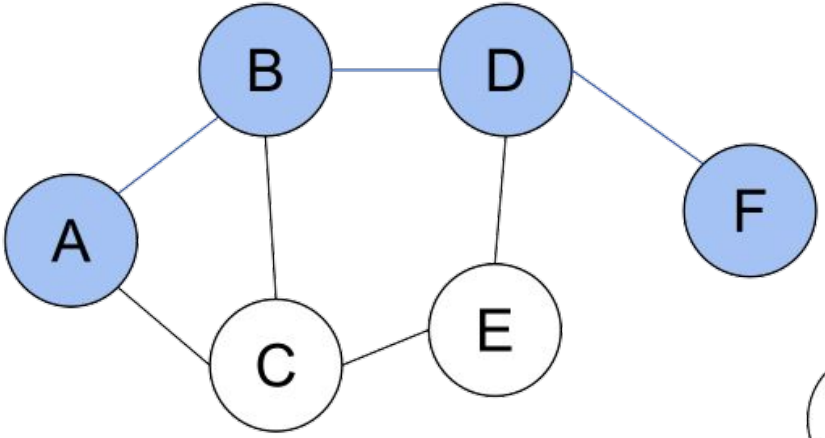
Граф	Матрица смежности
	$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$

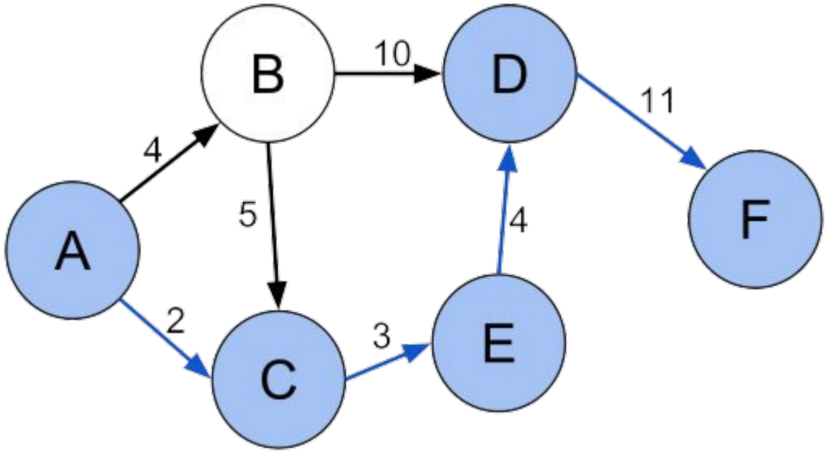
Способы задания графов

- **Матрица инцидентности** графа G— одна из форм представления графа, в которой указываются связи между инцидентными элементами графа (ребро(дуга) и вершина). Столбцы матрицы соответствуют ребрам, строки — вершинам. Ненулевое значение в ячейке матрицы указывает связь между вершиной и ребром (их инцидентность).
  - В случае ориентированного графа каждой дуге $< x, y >$ ставится в соответствующем столбце: «-1» в строке вершины x и «1» в строке вершины y; если связи между вершиной и ребром нет, то в соответствующую ячейку ставится «0».

Граф	Матрица инцидентности[1]
	$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Задача о кратчайшем пути.





Постановка задачи.

- Граф $G \coloneqq (V, E)$, необходимо найти минимальный маршрут(путь) в графе между двумя вершинами графа.

Обзор алгоритмов.

Название алгоритма	**Сложность**
Флойда-Уоршела	$O(\|V\|^3)$
Форда-Беллмана	$O(\|V\|\|E\|)$
Дейкстра	$O(\|V\|^2)$
Алгоритм Дейкстра для разреженных графов	$O(\|E\|log(\|V\|))$
A*	-
...	

Алгоритм Флойда-Уоршелла

Находит расстояние от каждой вершины до каждой за количество операций порядка . Веса могут быть отрицательными, но у нас не может быть циклов с отрицательной суммой весов рёбер.

```
for k = 1 to n
  for i = 1 to n
    for j = 1 to n
      W[i][j] = min(W[i][j], W[i][k] + W[k][j])
```

Алгоритм Форда-Беллмана

- Находит расстояние от одной вершины (дадим ей номер 0) до всех остальных за количество операций порядка $O(|V||E|)$.

for $v \in V$
  for i = 0 to $|V| - 1$
    do A_{vi} = $+\infty$
A_{s0} = 0
for i = 1 to $|V| - 1$
  do for $(u, v) \in E$
    if $A_{vi} > A_{u,i-1} + w(u, v)$
      then A_{vi} = $A_{u,i-1} + w(u, v)$
        P_{vi} = u

Алгоритм Дейкстры

- Находит расстояние от одной вершины (дадим ей номер 0) до всех остальных за количество операций порядка $O(|V|^2)$. Все веса неотрицательны.

Алгоритм Дейкстры для разреженных графов

- Делает то же самое, что и алгоритм Дейкстры, но за количество операций порядка $|E| \cdot log(|V|)$. Следует заметить, что m может быть порядка $|V|^2$, то есть эта вариация алгоритма Дейкстры не всегда быстрее классической, а только при маленьких $|E|$

Указания для задач

Для замера работы функции нужно использовать метод now() класса datetime модуля datetime

Указания для задач

```
array = [0] * N
array.insert(N,0)
```

```
import datetime


def main():
    t1 = datetime.datetime.now()
    #You'r code here
    ...
    print(datetime.datetime.now() - t1)

if __name__ == '__main__':
    main()
```

Указания для задач

```
import numpy as np


...
# Generate numpy Array with N random numbers
array = np.random.rand(N)
#Sort Array by quick sort
sorted(array)
...
```

Указания для задач

jupyter MyTests Last Checkpoint: a minute ago (unsaved changes)

File Edit View Insert Cell Kernel Help | Python 2

Code CellToolbar

In [6]:
```
import datetime
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
```

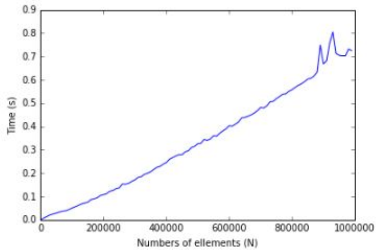
In [7]:
```
times = []
for N in xrange(1000, 1000000, 10000):
    array = np.random.rand(N)
    t1 = datetime.datetime.now()
    sorted(array)
    t_res = datetime.datetime.now() - t1
    times.append([N,t_res])
y = [t[1].total_seconds() for t in times]
x = [t[0] for t in times]
```

In [8]:
```
%matplotlib inline
```

In [9]:
```
plt.plot(x, y)

plt.xlabel('Numbers of ellements (N)')
plt.ylabel('Time (s)')

plt.savefig("test.png")
plt.show()
```



Задачи

- Реализовать алгоритм перемножения квадратных матриц. Матрицы могут задаваться как список списков. Считывать можно из файла потока ввода, или задавать случайным образом (используя функцию np.random.rand(N)). Оценить временную и ассимптотическую сложность алгоритма, построить график.
- Найти все пифагоровы тройки ($c^2 = a^2 + b^2$) для заданного интервала. Интервал задается парой чисел через пробел считанных из входного потока (например: 10 100) помните, что верхняя грань отрезка должна быть больше нижней. Если задано одно число, то считаем, что ограничение снизу равно по умолчанию 1. Оценить временную и ассимптотическую сложность алгоритма, построить график.
- Реализовать алгоритм факторизации числа (разложение числа как произведение двух других чисел). Оценить временную и ассимптотическую сложность алгоритма, построить график.
- Реализовать алгоритм рассчитывающий сочетания и размещения.
- Факториал довольно емкостная функция, при расчете которого для больших значений может случится переполнение (т.е. полученное число будет больше чем максимально возможное число в вашей системе). Подумайте как преодолеть эту проблему.

Задачи

Написать оболочку для работы с графами:

- создавать графы
- Выводить граф (в виде таблицы смежности)
- Удалять ребра
- Ищет путь в графе для заданных вершин
    - Флойда-Уоршела
    - Форда-Беллмана
    - Дейкстра

Спасибо за внимание!