нетология

Сегодня на лекции

Познакомимся с модулями в Python

Узнаем что такое графы и зачем они нужны

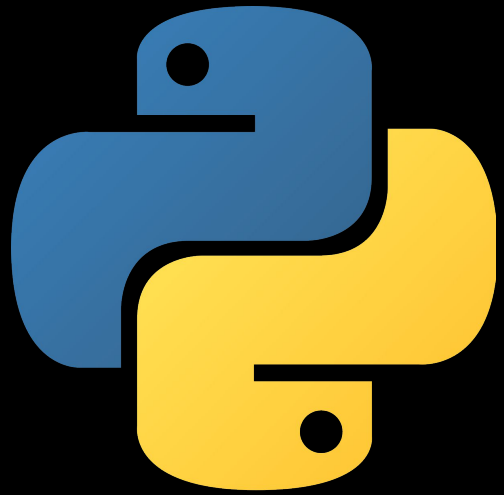
Опишем маршрут Фродо в виде графа и найдем для него самый простой путь

Модули

Встроенные в язык программирования функции доступны сразу. Чтобы их вызвать, не надо выполнять никаких дополнительных действий.

Доступа к дополнительным возможностям языка возможен через т.н. Модули.

Каждый модуль содержит коллекцию функций и классов, предназначенных для решения задач из определенной области.



нетология

Установка модуля

Многие модули свободно публикуются в интернете и доступны любому человеку.

Однако, перед их использованием их надо установить.

Чаще всего это делается при помощи команды pip в консоли компьютера.

```
pip install networkx
```

нетология

Импорт модуля

По умолчанию дополнительные модули не доступны в программе - для хранения их функций и классов нужно выделять память и тп.

Для работы с модулем его нужно импортировать в программу. После импорта Python узнает о существовании его функций и классов.

Разные способы импорта:

import networkx

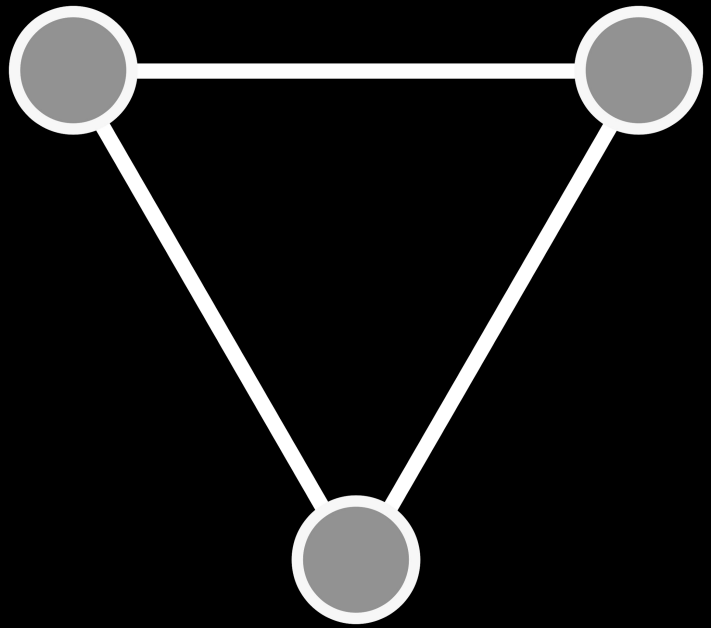
import networkx as nx

from networkx import *

from networkx import Graph

нетология

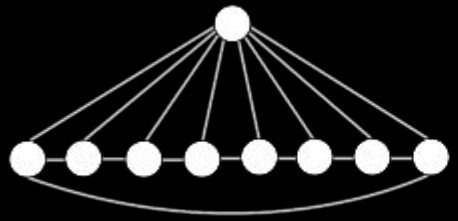
Графы



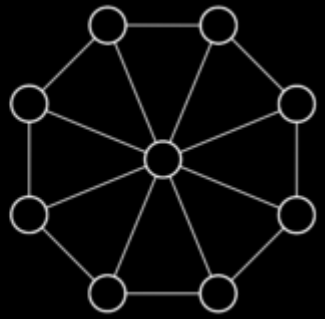
нетология

Граф

Граф - способ представления информации, основанный на точках и связях между ними.



Формально, граф обозначается как **G(V,E)**, где



- **V** - множество вершин
- **E** - множество ребер

У одного и тоже графа есть бесконечно много способов его визуализации.

нетология

Зачем нужны графы?

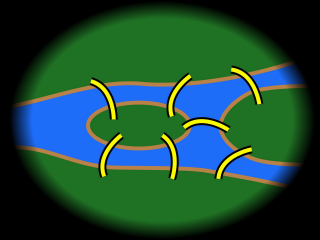
Графы - мощный и хорошо разработанный математический инструмент.

Его часто применяют в задачах:

- Поиска пути
- Минимального потока
- И тд

Задача о семи кёнигсбергских мостах

Можно ли пройти по всем семи мостам Кёнигсберга, не проходя ни по одному из них дважды.



Ответ: Нет, нельзя (Л. Эйлер, 1736 г.)

нетология

Графы в Python

Для работы с графами в python используется библиотека networkx.

```
!pip install networkx
```

Перед использованием ее необходимо установить и проимпортировать.

```
import networkx as nx
```

Основной объект - Граф.

```
g = nx.Graph()
```

Перед использованием его надо явно создать.

нетология

Графы в Python

Основные методы в графе:

- **add_node** - добавить узел
- **add_edge** - добавить ребро
- **draw** - отобразить

```
g.add_node("Шир")
g.add_node("Мордор")
```

```
g.add_edge('Шир', 'Мордор')
```

```
nx.draw(g, with_labels=True)
```

нетология

Практика

1. Откроем https://colab.research.google.com
2. Создадим новый блокнот
3. Установим и подключим библиотеку networkx
4. Создадим там граф из двух узлов (Шир, Мордор)
5. Выведем его на экран

Модификация графа

В граф можно динамически добавлять и удалять узлы/ребра и тп.

```
g.remove_edge('Шир', 'Мордор')
```

```
g.add_node("Пригорье")
```

```
g.add_edge('Шир', 'Пригорье')
```

нетология

Практика

Попробуем восстановить путь Фродо во Властелине Колец в виде графа

нетология

Поиск минимального пути

Для поиска самого короткого пути из точка А в точку Б используется готовая команда **shortest_path**

```
nx.shortest_path(g, 'Шир', 'Роковая Гора')
```

```
>>> ['Шир',
 'Пригорье',
 'Ривенделл',
 'Мория',
 'Лотлориен',
 'Мордор',
 'Роковая Гора']
```

нетология

Практика

1. Найдем самый короткий путь из Шира к Роковой Горе
2. Добавим промежуточный пункт и ребра в и из него
3. Убедимся, что короткий маршрут перестраивается

нетология

Веса в графе

Сейчас мы предполагаем, что все дороги в графе одинаково сложны.

Однако, на самом деле - это не так.

Для анализа таких графов ребрам можно назначать вес.

Назначение веса

Чтобы назначить вес ребру надо использовать параметр weight у функции add_edge

```
g.add_edge('Шир',
'Пригорье', weight=1)
```

нетология

Поиск пути

Для поиска пути в графе с весами у ребер надо использовать другую функцию из библиотеки networkx.

`nx.algorithms.shortest_paths.weighted.dijkstra_path`

```
nx.algorithms.shortest_paths.weighted.dijkstra_path(
    G,
    'Шир',
    'Роковая Гора',
    weight='weight'
)
```

https://ru.wikipedia.org/wiki/Алгоритм_Дейкстры

нетология

Практика

1. Создадим новый граф с весами
2. Посмотрим как меняется кратчайший путь в зависимости от сложности ребер

нетология

Итоги

Познакомились с графами и узнали как с ними работать в Python

Узнали как искать кратчайшие пути из точки А в точку Б

Нашли самый короткий путь из Шира к Роковой Горе

нетология

Немного на дом

Попробуйте добавить в граф пути Арагорна, Гендальфа и Мери с Пипином

Узнайте чей путь к Роковой Горе был самым простым

нетология

Спасибо за внимание