

# Интеллектуальные системы и технологии

## (практика)

Судаков Владимир Анатольевич

[sudakov@ws-dss.com](mailto:sudakov@ws-dss.com)

# Правила

- Решения публиковать на github. Присылать мне ссылку на почту, каждый раз когда хотите мне их показать
- Решения присылать за 48 часов до занятий
- Лучше прислать недоделанное решение, чем вообще ничего
- Решение можно исправлять после отправки
- Можно и нужно спрашивать. Лучше если вопрос будет коротким
- У нас будут контрольные, но когда – заранее неизвестно
- Опоздание больше 15 минут – пропуск. Отсутствие результатов работы – пропуск.
- За активность/ответы на занятиях будут дополнительные плюсы.
- Вовремя решенные ДЗ, контрольные без ошибок, и не более 2-х пропусков – автомат на экзамене

# ИСТОЧНИКИ

- Марк Лутц. Изучаем Python
  - Luciano Ramalho. Fluent Python
  - Joel Grus. Data Science from Scratch
  - Allen B. Downey. Think Complexity
  - PEP8
- 
- [https://github.com/sudakov/lab\\_it](https://github.com/sudakov/lab_it)

# Дзен Python

- Красивое лучше, чем уродливое.
- Явное лучше, чем неявное.
- Простое лучше, чем сложное.
- Сложное лучше, чем запутанное.
- Плоское лучше, чем вложенное.
- Разреженное лучше, чем плотное.
- Читаемость имеет значение.
- Особые случаи не настолько особые, чтобы нарушать правила.
- При этом практичность важнее безупречности.
- Ошибки никогда не должны замалчиваться.
- Если не замалчиваются явно.
- Встретив двусмысленность, отбрось искушение угадать.
- Должен существовать один — и, желательно, *только* один — очевидный способ сделать это.
- Хотя он поначалу может быть и не очевиден, если вы не голландец.
- Сейчас лучше, чем никогда.
- Хотя никогда зачастую лучше, чем *прямо* сейчас.
- Если реализацию сложно объяснить — идея плоха.
- Если реализацию легко объяснить — идея, *возможно*, хороша.
- Пространства имён — отличная вещь! Давайте будем делать их больше!

# Метод $k$ ближайших соседей

- Метрика близости
- Голосование  $k$  ближайших соседей
- А если результат равный?

# Задача № 1

- Давайте познакомимся:
  - Что предпочитаете iPhone или Android? Научите ИИ прогнозировать разновидность телефона методом k ближайших соседей.
- Разбиваемся на команды 3-4 человека:
  - Распределение ролей
  - Парное программирование
  - Подготовка исходных данных
  - Тестирование
  - Анализ – какое k лучше?
  - Показ решения
  - Code review чужой бригадой.
- Обсуждение
  - Какое решение лучше и почему?

# Центральность в графе

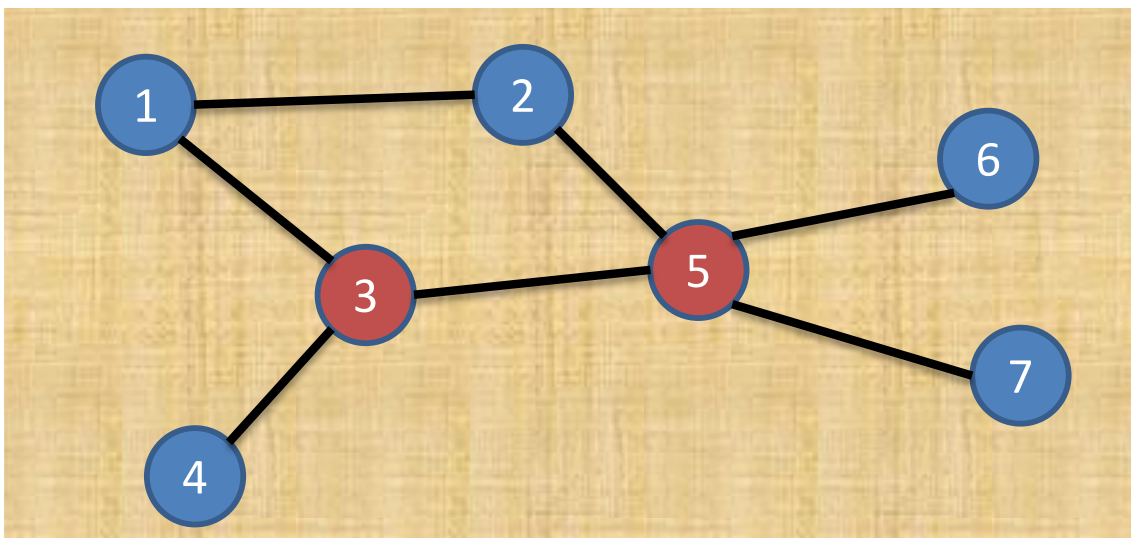
Центральность вершин в графе – это вектор, сопоставляющей каждой вершине графа некоторое число (индекс).

Наиболее распространенные индексы:

- Степенная центральность (degree centrality);
- Центральность по близости (closeness centrality);
- Центральность по посредничеству (betweenness centrality);
- Центральность по собственному вектору (eigenvector centrality);
- Центральность PageRank.

# Центральность по близости

Вершина, находящаяся ближе всех к другим вершинам сети, является наиболее центральной

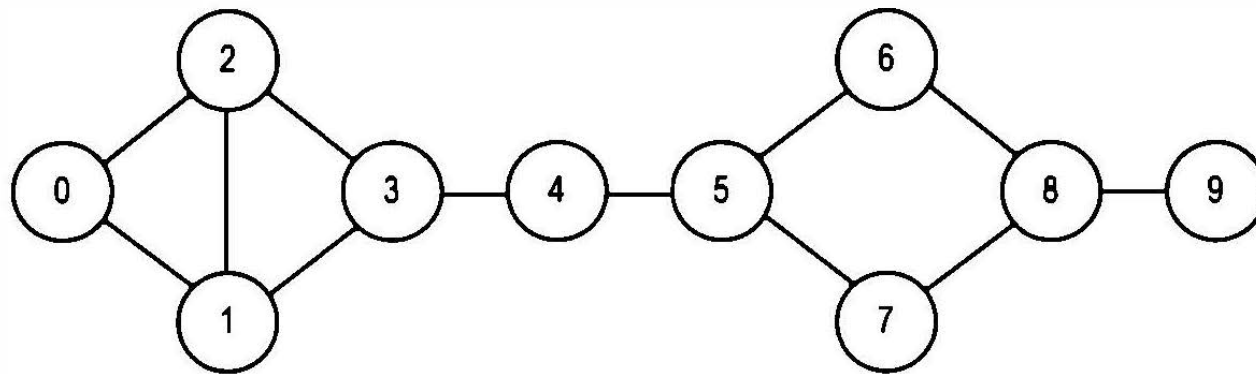


$$C_i = \frac{1}{\sum_j d_{ij}} \quad C_i = \sum_j \frac{1}{d_{ij}}$$



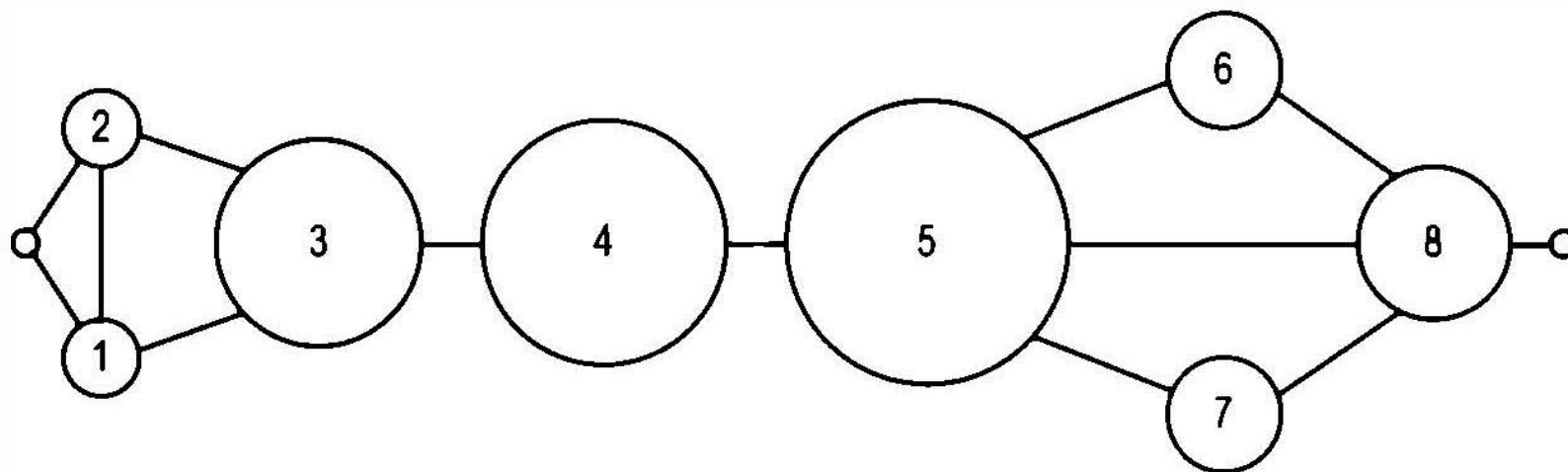
# Центральность по посредничеству

Вершина, через которую проходит наибольшее число кратчайших путей, является наиболее центральной.



$$C_i = \sum_{jk} \frac{w_{jk}(i)}{w_{jk}}$$

# Центральность по посредничеству



# Центральность по собственному значению

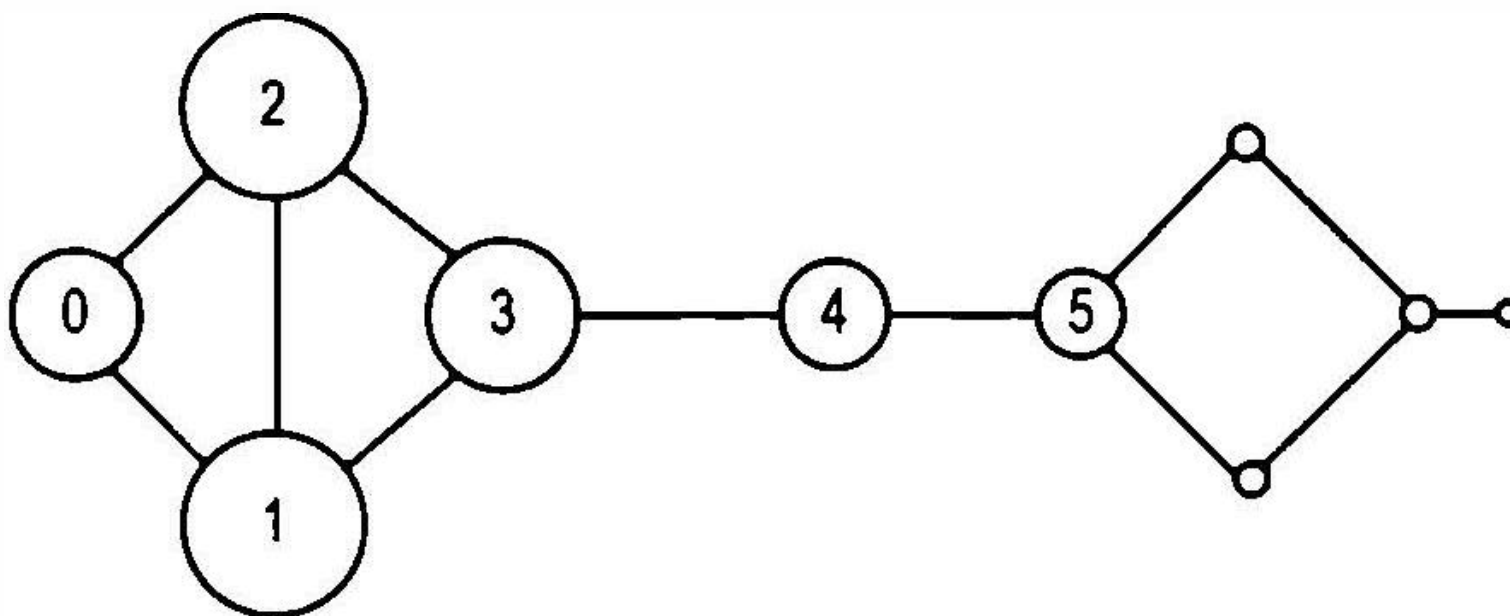
Центральность вершины  $i$  зависит от центральностей соседей вершины  $i$ .

$$x_i = \frac{1}{\lambda} \sum_{j \in F_i} x_j = \frac{1}{\lambda} \sum_j a_{ij} x_j$$

$$\lambda x = Ax$$

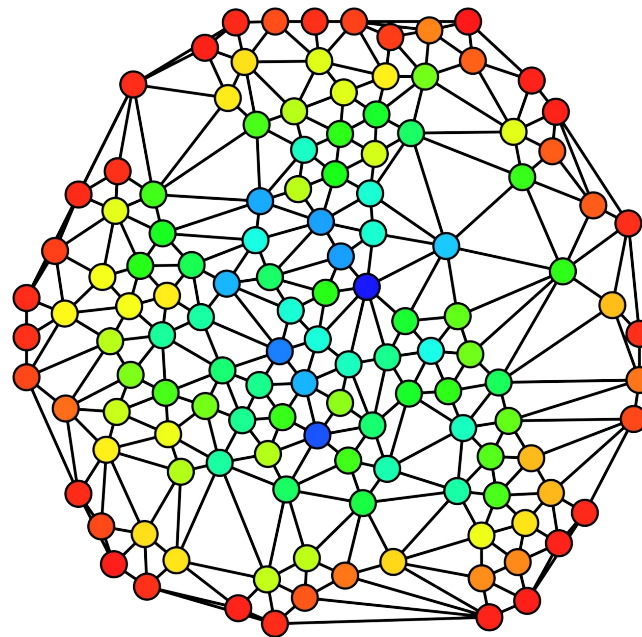
- Выбирается собственный вектор, соответствующий максимальному собственному значению.
- Данная центральность учитывает дальние взаимодействия.
- Наиболее центральными считаются вершины, которые сами указывают на сильные вершины.

# Центральность по собственному значению



# Задача № 2

- Давайте соберем информацию о друзьях и друзьях друзей из VK для членов Вашей группы
- Оценить центральность: по посредничеству, по близости, собственного вектора (только для членов Вашей группы)



## Задача № 3

- Решить задачу определения самолета: военный или гражданский по фотографии.
- Для решения использовать провайдеров LLM на hugging face.
- Сравнить метрики двух выбранных LLM.
- Составьте выборку так, чтобы там были нетривиальные случаи.
- Не используйте большую выборку (30 самолетов достаточно), так как на hugging face ограничение на число бесплатных запросов.
- Вместо самолетов можно взять любые другие объекты.

# Задание №4

- Бабушка разлила по 5 стаканам компот. Но она сделала это неровно.
- Надо перелить компот так, чтобы во всех стаканах было одинаковый объем компота.
- Научите агента переливать компот, чтобы он поднимал стаканы для переливания минимальное число раз (можно поднять один стакан и разлить на несколько других - это одно поднятие).
- На какие вопросы нужно ответить прежде чем начать программирование?
- Нужно обязательно вывести кривую обучения (зависимость награды агента от числа эпизодов обучения).