Филиал Московского Государственного Университета

имени М.В.Ломоносова в городе Ташкенте

Факультет прикладной математики и информатики

Кафедра прикладной математики и информатики

**Борисов Артур Александрович**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**на тему:**

**«О нахождении степени посредничества в динамически меняющихся структурах»**

**по направлению 01.03.02 «Прикладная математика и информатика»**

ВКР рассмотрена и рекомендована к защите зав. кафедрой «ПМиИ», д.ф.-м.н., профессором, Гасановым Э. Э.

Научный руководитель: к.ф.-м.н. **Алексеев Дмитрий Владимирович**

«20» мая 2024 г.

Ташкент 2024

Оглавление

[1 Аннотация 3](#_Toc167308790)

[2 Введение 3](#_Toc167308791)

[3 Готовые решения: 3](#_Toc167308792)

[4 Постановка задачи. 4](#_Toc167308793)

[5 Интерфейс программы 7](#_Toc167308794)

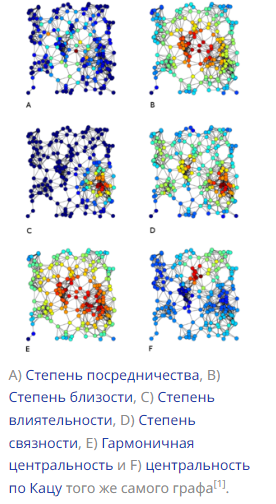
[5 Заключение 7](#_Toc167308795)

[6 Список литературы: 8](#_Toc167308796)

# 1 Аннотация

В данной работе рассматривается задача нахождения степени посредничества для неориентированных невзвешенных графов. Цель работы заключается в разработке алгоритма определения степени посредничества на основе Алгоритма Брандеса [1] и последующем сравнение с существующими методами из библиотеки Networkx; реализация интерфейса с динамическим изменением структуры и визуализации вычисляемых метрик.

# 2 Введение

Понятие степени посредничества (betweenness centrality) в теории графов было впервые введено в 1977 году социологом Линусом Фрименом. Он использовал это понятие для изучения взаимодействия между участниками в социальной сети.

Одной из задач, которая решается с помощью степени посредничества, является поиск важных узлов в сетях. Важность узла зависит от его степени посредничества. Например, в социальной сети важные узлы могут быть связаны с влиятельными людьми или общими интересами. В биологической сети важные узлы могут означать гены, ответственные за регуляцию других генов.

Другой задачей, связанной со степенью посредничества, является поиск наиболее быстрых маршрутов в сетях. Как правило, маршруты через узлы с высоким посредничеством более быстрые и надежные.

# 3 Готовые решения:

3.1 Библиотека Networkx [2].

Модуль позволяет строить различные типы графов и реализовывать достаточно сложные алгоритмы сетевого анализа. Модуль позволяет хранить базы данных, оформленные в виде сетей как в собственных форматах данных, так и в обычном графическом виде.

Networks позволяет строить сети по алгоритмам (как уже известным, так и вновь создаваемым), а также сети случайного и классического вида, позволяя создавать сложные математические модели многих видов. Любой сетевой граф состоит из узлов, содержащих какие-либо данные, а также соединяющих узлы рёбра (в некоторых случаях имеющие направление), также состоящих из любых данных. Достоинствами данного модуля является простота освоения при обладании широкой сферой использования. Сетевой граф чаще всего генерируется из какой-либо базы данных, представленной в виде файла, либо онлайн-структурой.

В библиотеке NetworkX для нахождения степени посредничества можно использовать следующие метод на основе алгоритма Ульрика Брандеса [1]:

***betweenness\_centrality(G, normalized, weight, endpoints, seed)***

Этот метод вычисляет степень посредничества для всех вершин в графе G.

* Параметр normalized (по умолчанию False) определяет, должны ли значения степени посредничества быть нормализованы в диапазоне от 0 до 1.
* Параметр weight позволяет учитывать веса ребер при вычислении степени посредничества.
* Параметр endpoints указывает, должны ли конечные вершины ребер учитываться при вычислении степени посредничества.
* Параметр seed задает начальное значение для генератора случайных чисел.

Оценка времени выполнения данного метода равна *O*(n\*m) в невзвешенных сетях и в взвешенных, n – количество вершин, m — количество ребер) при использовании O(n+m) памяти.

# 4 Постановка задачи.

Определение 1. Неориентированным графом 𝐺 называется пара 𝐺 = (𝑉, 𝐸), где 𝑉 — конечное непустое множество, а 𝐸 ⊂ 𝑉, то есть является множеством неупорядоченных пар различных элементов из 𝑉. Элементы множества 𝑉 называются вершинами, а множества 𝐸 —– ребрами.

Определение 2. Граф G, называется ориентированным, если 𝐸 ⊂ 𝑉 × 𝑉, то есть множество 𝐸 является подмножеством упорядоченных пар различных элементов из 𝑉. В таком случае элементы множества 𝐸 также могут называться дугами.

Определение 3. Граф 𝐺 = (𝑉, 𝐸) называется взвешенным, если его каждому из его ребер поставлено в соответствие некоторое числовое значение (вес ребра).

Определение 4. Путем в графе называется последовательность вида

где ∈ 𝑉, ∈ 𝐸, = ( , ), 𝑖 = 1, …, 𝑘.

Определение 5. Дана весовая функция , которая отображает ребра на их веса, значения которых выражаются действительными числами, и неориентированный граф G. Тогда кратчайшим путём из вершины в вершину будет называться путь (где и ), который имеет минимальное значение суммы ). Если все ребра в графе имеют единичный вес, то задача сводится к определению наименьшего количества обходимых ребер.

Определение 6.

Пусть

– кратчайший путь между вершинами «s» и «t» в графе G

– кол-во кратчайших путей из «s» в «t»

– кол-во кратчайших путей из «s» в «t» в которых есть вершина «v»

– множество предшественников вершины «v» на кратчайших путях из «s»

Псевдокод Алгоритма Брандеса:

C\_B[v] <— 0, v ∈V;

for s V do

S <— empty stack;

P[w] <— empty list, w ∈V;

σ[t] <— 0, t ∈V; σ[s] <— 1;

d[t]<— (-l), t ∈V; d[s] <— 0;

Q <— empty queue;

enqueue s —> Q;

while Q not empty do

dequeue v <— Q;

push v —> S;

foreach neighbor w of v do

// w found for the first time?

if d[w] < 0 then

enqueue w —> Q;

d[w]<—d[v] + 1;

end

// shortest path to w via v?

if d[w] = d[v] + 1 then

σ[w] <—σ[w] + σ[v];

append v —> P[w];

end

end

end

δ[v] <— 0, v ∈V;

// S returns vertices in order of non-increasing distance from s

while S not empty do

pop w <— S;

for v ∈P[w] do δ[v] <— δ[v] + (1 + δ[w]);

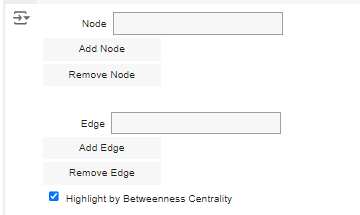
if w ≠s then C\_B[w] <— C\_B[w] +δ[w]/2;

end

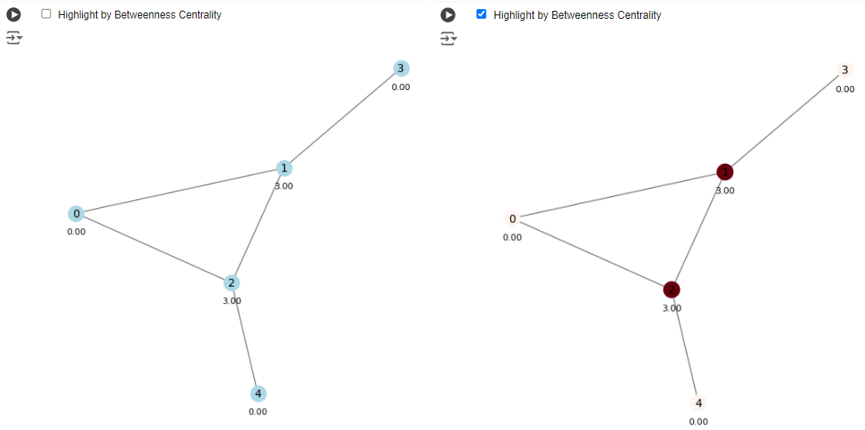
end

# 5 Интерфейс программы

Реализация интерфейса выполнена в бесплатная интерактивной облачной среде для работы с кодом на языке Python от Google – Colab.



Первое поле отвечает за добавление вершин, Второе за добавление ребер. Галочка в нажатом состоянии красить градиентом найденные степени для всех вершин (в не нажатом графа не раскрашен).



Под каждой вершиной указывается найденная степень посреднечества.

# 5 Тестирование

# 5 Заключение

В ходе дипломной работы был реализован алгоритм Брандеса и алгоритм поиска в ширину для нахождения степени посредничества в неориентированных невзвешенных графах. Разработана программа динамического изменения графа с пересчетом значений степеней, и визуализация с выделением вершин градиентом по возрастанию значения степени. Все результаты выложены в открытый доступ на GitHub.

Ссылка на репрезенторий: <https://github.com/ArturBorisov01/Algoritm-Brandes/tree/master>

# 6 Список литературы:

1. Статья на Википедии.Определение и описание индексов центральности <https://w.wiki/6kaC>
2. Документация библиотеки Networkx для python <https://networkx.org/documentation/stable/tutorial.html>
3. Ульрик Брандес: более быстрый алгоритм централизации промежуточности . Журнал математической социологии 25 (2): 163–177, 2001 г. [https://doi.org/10.1080/0022250X.2001.9990249 .](https://doi.org/10.1080/0022250X.2001.9990249%20.)
4. Веб среда разработки <https://colab.research.google.com/>