Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ РАНЖУВАННЯ НА ОСНОВІ НАУКОМЕТРИЧНИХ ПОКАЗНИКІВ

Текстова частина до курсової роботи

Виконала студентка 1 курсу
МП «Інженерія програмного забезпечення»
Андрусів Соломія Ігорівна

Керівник курсової роботи
доцент
Олецький Олексій Віталійович

Київ 2022

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем,

доцент, к. ф.-м. н. О. П. Жежерун

_____

(підпис)

“____”_____2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студента Андрусів Соломії Ігорівни факультету інформатики 1 курсу

Тема: Порівняльний аналіз методів ранжування на основі наукометричних показників

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план

Зміст

Перелік умовних позначень

Вступ

Розділ 1 Аналіз предметної області. Постановка завдання курсової роботи

Розділ 2 Теоретичні відомості

Розділ 3 Опис розробки програмного продукту

Висновки

Перелік використаних джерел

Дата видачі “___” _____ 2022 р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Календарний план виконання курсової роботи

Тема: Розробка API для системи запису на вибіркові дисципліни

Календарний план виконання роботи:

№ п/п	Назва етапу курсового проекту (роботи)	Термін виконання етапу	Примітка
2.1.	Отримання завдання на курсову роботу.	21.10.2021	
2.2.	Ознайомлення з існуючою інформацією по темі	22.10.2021-18.01.2022	
2.3.	Ознайомлення з існуючими системами-аналогами роботи	22.10.2021-18.01.2022	
2.4.	Початок створення практичної частини	18.01	
2.5.	Початок написання теоретичної частини	15.03	
2.6.	Подання проміжної версії практичної частини	22.03	
2.7.	Аналіз практичної частини; її коригування	27.03	
2.8.	Остаточне завершення написання теоретичної частини роботи та розробки практичної частини; коригування	31.05-08.06	
2.9.	Створення презентації	09.06-13.06	
2.10	Захист курсової роботи	14.06	

Студент Андрусів С.І.

Керівник Олецький О. В.

“_____” _____

ЗМІСТ

Перелік термінів та умовних позначень

Датасет – набір даних, зазвичай багато рядків з однотипною інформацією, використовується для статистичного аналізу та у машинному навчанні.

Ранжування – впорядкування даних у порядку спадання починаючи з того, що найбільш підходить під заданий запит.

Релевантність – ступінь відповідності знайденої інформації до запиту.

PageRank – алгоритми оцінки важливості веб-сторінки, що базується на ранжуванні посилань.

ВСТУП

Основною метою роботи є аналіз головних наукометричних показників та методів формування рейтингу науковця на їх основі. У роботі буде використано такі основні метрики, як індекс Хірша та PageRank.

Завданням курсової роботи є написання програмного коду, що буде обраховувати оцінку активності науковців кількома методами та проведення аналізу їх релевантності на основі отриманих значень. Визначення шаблонів ситуацій, в яких отримані результати збігаються, та в яких заперечують один одного. Моделювання ситуацій, що відображають певні проблеми релевантності результатів метрик.

Об’єктом дослідження є система методів ранжування науковців на основі таких наукометричних показників, як кількість публікацій, посилань на публікації, частота написання текстів, цитування, індекс Хірша тощо.

Предметом дослідження є відмінності у результуючій оцінці активності науковця в залежності від використаних показників та методу ранжування, а також уникнення підвищення рейтингу на основі штучного збільшення цих показників.

Практична частина роботи написана на мові програмування Python, для математичних обрахунків використано бібліотеку numpy для роботи з матрицями та реалізації випадкового блукання. Для статистичного аналізу використано датасет з даними про 100000 науковців, що був використаний для дослідження, проілюстрованого журналом PlosBiology[1].

Текстова частина до курсової роботи складається з вступу, трьох основних розділів та висновків.

У першому розділі «Аналіз предметної області. Постановка завдання курсової роботи» розглядаються основні методи формування оцінки науковця на момент дослідження та визначаються основні кроки для реалізації практичної частини.

У другому розділі «Теоретичні відомості» надано основну інформацію про обидва порівнюваних методи та їх модифікації, а також описано основні проблеми кожного з них, проілюстровано ситуації, коли обидва методи дають різні значення.

У третьому розділі «Опис розробки програмного продукту» описано процес реалізації практичної частини, а також функції та методи, реалізовані у застосунку.

У висновках підбиваються підсумки дослідження та надаються ідеї щодо усунення певних проблем вищезазначених методів.

Посилання на наукові роботи та інші джерела інформації містяться у списку використаної літератури. Всі матеріали, що доповнюють текст, зокрема діаграми, зображення та програмний код знаходяться у «Додатках».

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАВДАННЯ КУРСОВОЇ РОБОТИ

1.1. Аналіз сучасного стану питання та обґрунтування теми

На даний момент, ранжування на основі наукометричних показників все частіше використовують для оцінки наукових публікацій та їх авторів, на рівні з особистими рецензіями експертів в потрібній галузі.

Для чого потрібно ранжувати науковців та їх роботи?

По-перше, важливим завданням є виокремлення авторів, що вміють ефективно проводити дослідження та описувати результати, робити правильні висновки та вносити новизну у світову наукову базу, з метою залучення їх до важливих наукових проектів, можливо, навіть на державному рівні.

По-друге, прозора система формування рейтингу допомагає авторам знайти релевантні наукові джерела для цитувань та підтвердження або спростування висновків, зроблених на основі своїх досліджень.

По-третє, для забезпечення середньостатистичного користувача достовірною інформацією на будь-який запит. Зараз, із розвитком мережі Інтернет, де всі наукові роботи можна знайти в електронному вигляді, питання достовірності інформації є одним з найважливіших. Тому дуже важливо, щоб на запит користувача в пошуковій системі йому надавалися релевантні та унікальні статті, що містять достовірні дані. Для всіх трьох сторін – користувача, автора релевантної статті та пошукової системи, така взаємодія є найбільш вигідною та бажаною.

Основне завдання наукометрики – забезпечити найбільш точні статистичні дані щодо розвитку науки у різних галузях.

Історично виокремилися методи та показники, що досі залишаються основними для обчислення таких даних. Серед таких методів – індекс Хірша, що базується на кількості цитувань автора, та є основою для багатьох модифікацій.

Метод Page Rank, що був вперше впроваджений для ранжування веб-сторінок під запит користувача на основі посилань, згодом почав також використовуватися в наукометриці, в основному для електронних наукових ресурсів, як-от Google Scholar, Scopus, Index Copernicus тощо. Ця методика заглиблюється у зв’язки між документами, а не просто підраховує їх кількість, як це робить індекс Хірша. Переваги і недоліки такого заглиблення будуть розглянуті у Розділі 2.

На даний момент, немає одного уніфікованого методу ранжування, адже формування релевантного рейтингу залежить від великої кількості показників. Навколо деяких показників, як-от кількість років між першою та останньою публікацією, все ще ведуться дискусії. Тому у своєму дослідженні я розгляну два вищезгаданих методи, перевірю їх на змодельованих даних.

1.2. Аналіз предметної області

Для перших кроків дослідження використала датасет з статті, опублікованої у журналі PlosBiology[1]. Датасет містить дані про сто тисяч науковців з різних країн світу станом на 2017 рік. Він містить зокрема такі дані про кожного автора, як повне ім'я, назву та країну навчального закладу, до якого прив'язана більшість статей, рік першої та останньої публікації, галузь та підгалузь, у якій опублікована більшість робіт, а також уже обраховані наукометричні показники, зокрема:

- кількість написаних робіт(з 1996 до 2017 року);

- індекс Хірша (включаючи та виключаючи самоцитування);

- hm-індекс, який є модифікацією індексу Хірша для наукових робіт, написаних у співавторстві (включаючи та виключаючи самоцитування);

- кількість цитувань усіх робіт автора (у співавторстві та окремо);

- кількість робіт з та без співавторства;

- відсоток самоцитування;

- ранг PageRank.

Ці дані дозволяють ідентифікувати певні проблеми, що заважають коректно обчислити рейтинг науковця, зокрема:

- Самоцитування;
- Коректне обчислення вкладу конкретного науковця у роботу, що була написана у співавторстві;
- Невідповідність результатів, обчислених для індексу Хірша та рангу PageRank.

Також, існують певні проблеми, які даний датасет не може проілюструвати, зокрема:

- Не включення прямих цитат на відомі джерела(наприклад, «як сказав Ейнштейн…»);
- Накручування рейтингу групою авторів, що цитують один одного;
- Проблема обчислення рейтингу для науковців у кого багато робіт з маленькою кількістю цитувань, та у кого одна-дві роботи, які досить активно цитуються.

Як бачимо, при обчисленні кількості цитувань повинен враховуватися людський фактор, а не тільки прямий результат посилань, і якщо самоцитування доволі легко виявити та усунути, то кількох людей, що можуть цілеспрямовано цитувати один одного з метою підвищення рейтингу нечесним шляхом, виявити досить складно.

Проаналізувавши всі дані, що містяться у датасеті, виділила наукометричні показники, які є статичними, та ті, які залежать від людського фактору.

До статичних показників можна віднести дату першої та останньої публікації та кількість робіт.

Цитування – один із основних показників, що використовуються усіма популярними методами ранжування. Можу назвати кілька сценаріїв, у яких він може бути залежним від людського фактору, у мірі зростання складності їх ідентифікації:

- наявність самоцитування;
- цитування групою людей робіт один одного;

Такі сценарії заважають обчислити той рейтинг науковця, на який він заслуговує, і допомагають йому піднятися за рахунок нечесних маніпуляцій.

1.3. Постановка завдання

Зважаючи на перелічені вище проблеми з достовірністю наукометричних показників та відсутність універсального методу ранжування науковців, основними завданнями мого дослідження є:

1. Дослідження популярних методів ранжування науковців. Ідентифікація можливих проблем.
2. Дослідження датасету та візуалізація статистичних даних, що підтверджують існування проблем, зазначених у пункті 1.
3. Практична імплементація методів обчислення індексу Хірша, його модифікацій, та PageRank.
4. Практичне моделювання ситуацій, при яких виникають суттєві розбіжності у результатах обчислених обраними методами ранжування.

5. Пропозиції щодо того, як можна уникнути певних вищезазначених проблем з показниками для ранжування.

РОЗДІЛ 2. ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1. Загальна інформація про індекс Хірша

Індекс Хірша вперше був запроваджений Хорхе Хіршем у 2005 році. Обчислення цього індексу доволі просте – він дорівнює максимальній кількості N робіт, цитування яких рівне або більше N. Таким чином, з рейтингів виключаються роботи, що взагалі не цитуються, проте не враховано, скільки робіт з малою кількістю цитувань було відкинуто, та наскільки максимальна кількість цитувань роботи автора більша за N[3].

Цей показник використовується у багатьох бібліографічних базах даних та інших наукових електронних порталах, що містять велику кількість наукових робіт в певних галузях. Наприклад, у бібліографічній базі Scopus індекс Хірша є одним з основних показників, та обраховується автоматично для всіх авторів, що публікували свої роботи у рейтингових наукових журналах. У науковій бібліотеці Google Scholar зареєстровані автори також можуть миттєво обчислити свій показник h-index, проте, він не є основним.

Ще одною перевагою індексу Хірша є його незалежність від тривалості кар'єри публіциста. Як бачимо з датасету[1], індекс Хірша та стаж у написанні текстів не є прямо пропорційними(рис.1).

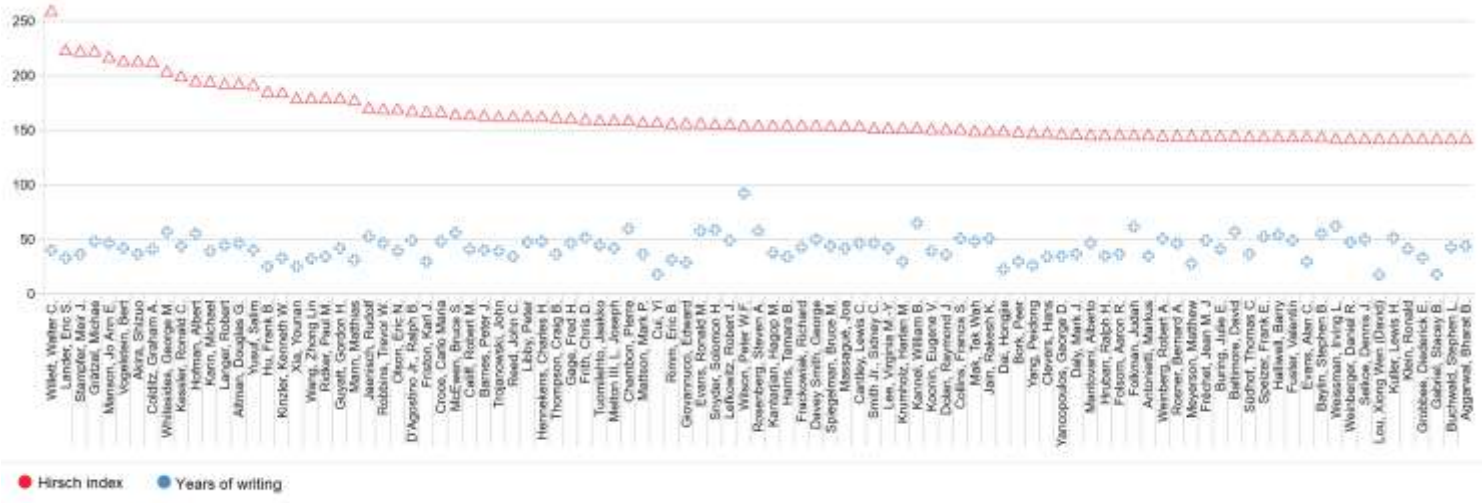


Рисунок 1: Залежність індексу Хірша від віку публікацій

2.2. Основні проблеми індексу Хірша та його модифікації

Для авторів, що опублікували одну-дві роботи, які досить активно цитуються, обчислений індекс Хірша проігнорує великий відрив у кількості цитувань від N та буде дуже низьким, у чому заключається одна з основних проблем індексу Хірша.

Також, проблеми з'являються при досягненні високого індексу Хірша. Чим більше число N, тим більше цитувань повинна набрати наступна робота, щоб це число збільшилося. Тому зі збільшенням індексу він стає більш статичним і може не змінюватися роками, протягом яких науковець активно працює та публікує нові статті.

Для усунення цих двох основних недоліків на даний момент створено ряд модифікацій індексу Хірша, тобто метрик, що мають в своїй основі h-index, або були створені базуючись на його ідеї [6]. До них відносять:

- g-index - індекс, створений з метою удосконалення обчислення індексу для автора, у якого небагато робіт, що активно цитуються. Суть метрики полягає в тому, що n статей повинні набрати сумарно не менш ніж n^2 цитувань. Тобто це може бути кілька об'ємних статей з великою кількістю цитувань, та всі менші, у яких практично немає цитувань, і які ігноруються при обчисленні стандартного h-index [4].
- $\text{hg\_index} = \sqrt{\text{h\_index} * \text{g\_index}}$.

  Середнє геометричне h-індексу та g-індексу. Така метрика урівноважує індекс з обох сторін – не будуть враховуватися всі роботи з низьким рівнем цитування, які можуть бути включені при обчисленні g-індексу, та не будуть ігноруватися роботи із значним відривом від n-цитувань при обчисленні h-індексу.

- R_index $= \sqrt{\sum_{j=1,h} c_j}$, де c_j – кількість цитувань j-ої публікації.

  AR_index $= \sqrt{\sum_{j=1,h} \frac{c_j}{a_j}}$, де a_j – вік j-ої публікації.

  Квадратний корінь сумарного цитування n робіт, що були враховані в індексі Хірша. Комбінування h та AR-індексів дозволяє створити метрику, що буде збільшуватися чи зменшуватися з часом, навіть при повній бездіяльності автора, та може використовуватися в галузях, що активно розвиваються, де дослідження швидко втрачають актуальність та новизну [7].

- e_index $= \sqrt{\sum_{j=1,h} c_j - h^2}$, де c_j – кількість цитувань j-ої публікації

  Квадратний корінь із надлишкового цитування h-індексу. Метою e-індексу є розрізнення вчених із подібними h-індексами, але різними моделями цитування. Він схожий за своєю природою на g-індекс тим, що приділяє більше уваги статтям, які активно цитуються.

- $m\_quotient = \frac{h}{y}$, де y – вік найдавнішої публікації. Метрика, також запропонована Хіршем у 2005 році, відносна метрика, що враховує тривалість кар'єри публіциста. Для молодих науковців така метрика не єфективною, адже може суттєво зменшити їх індекс порівняно з старшими колегами.

- i10 - метрика, що використовується у Google Scholar. Підраховує кількість статей, що мають більше десяти цитувань.

2.3. Загальна інформація про PageRank

PageRank – алгоритм ранжування веб-сторінок, що вперше був використаний пошуковою системою Google [2].

Суть алгоритму лежить у присвоєнні кожній веб-сторінці певної ваги, що залежить від кількості вхідних посилань.

На вхід подається орієнтовний граф, вершини якого - документи електронної бібліотеки, а ребра – цитування документів. Вага кожного документу формується на основі ваг всіх інших робіт, що на нього посилаються. Таким чином уникається проблема накручування кількості цитувань іншими непопулярними роботами – у таких робіт маленька вага і їх вклад у PageRank документу практично ігнорується.

Для обчислення ваг усіх вершин вихідного зважного графа може бути використано різні алгоритми. До прикладу, у веб-пошуковиках на вагу впливають результати індексування сторінок за ключовими словами відповідно до запиту користувача.

У своїй практичній реалізації для знаходження ваг я скористалася ланцюгами Маркова та обчислила зважений граф всіх документів як стаціонарний розподіл [8].

Ланцюги Маркова демонструють основну властивість Маркова – залежність наступного стану тільки від поточного, а не від усіх, що були до нього. В основі алгоритму лежить матриця переходів P та вектор станів S, у якому кожен наступний стан вираховується за формулою:

$$s_{i+1} = s_i * P$$

Стаціонарний розподіл для ланцюгів Маркова – такий розподіл, для якого $s_{i+1} = s_i$ тривалу кількість кроків, що означає, що випадкове блукання стабілізувалося та певний час залишається незмінним.

На виході алгоритм PageRank повертає вектор ваг для кожного документу, що по суті є ймовірностями переходу користувача на заданий документ, та в сумі становлять одиницю. Для обчислення рангу автора найпростішим способом є сума PageRank усіх його робіт. Тоді усі ранги науковців також сумарно становитимуть одиницю і баланс буде збережено.

2.4. Основні проблеми PageRank

Одним з недоліків PageRank можна назвати те, що вага кожного документу залежить від всіх інших документів в мережі, та нормується відносно роботи з найбільшою кількістю вхідних посилань. Тобто, якщо у системі буде дуже великий відрив між мінімумом та максимумом вхідних лінків, PageRank для робіт з маленькою кількістю посилань може бути близьким до нуля. В той же час, обчислення індексу Хірша для певного автора не залежить від якості робіт, написаних іншими авторами. Це і є основний недолік «заглиблення» PageRank у зв’язки між документами.

Щоб виправити ситуацію, можна знайти так-званий одиничний усереднений PagePank. У такому випадку сума рангів для N робіт буде становити не одиницю, а N, але в середньому для роботи він буде близьким до одиниці. Важливо пам'ятати, що при такій модифікації PageRank вже некоректно називати ймовірністю переходу на документ, тепер він виражений відносними одиницями.

Ще однією проблемою, з якою зіткнуться переважно ті, хто спробує самостійно впроваджувати алгоритм для ранжування наукових робіт, є значна розрідженість графу. Історично алгоритм було створено для використання у веб-мережах, де з великою ймовірністю цей граф буде майже повністю зв'язним. Для наукових робіт же він виглядатиме приблизно як зв'язні острівки документів в окремих галузях, що не пов'язані між собою, як, наприклад, математика та лінгвістика. Тобто такий граф, представлений матрицею розміром NxN, буде часто містити рядки з N-1 кількістю нулів.

Також, важливо зазначити, що на відміну від індексу Хірша, PageRank не є основною метрикою у наукових бібліотеках, і автору практично неможливо дізнатися свій ранг, та порівняти його з рангами інших науковців.

2.5. Ситуації коли методи дають різні результати

З даних досліджуваного датасету відобразила на діаграмах медіану індексу Хірша та PageRank для кожної наукової галузі(рис.2-3).

Рисунок 2: Медіана індексу Хірша у різних галузях

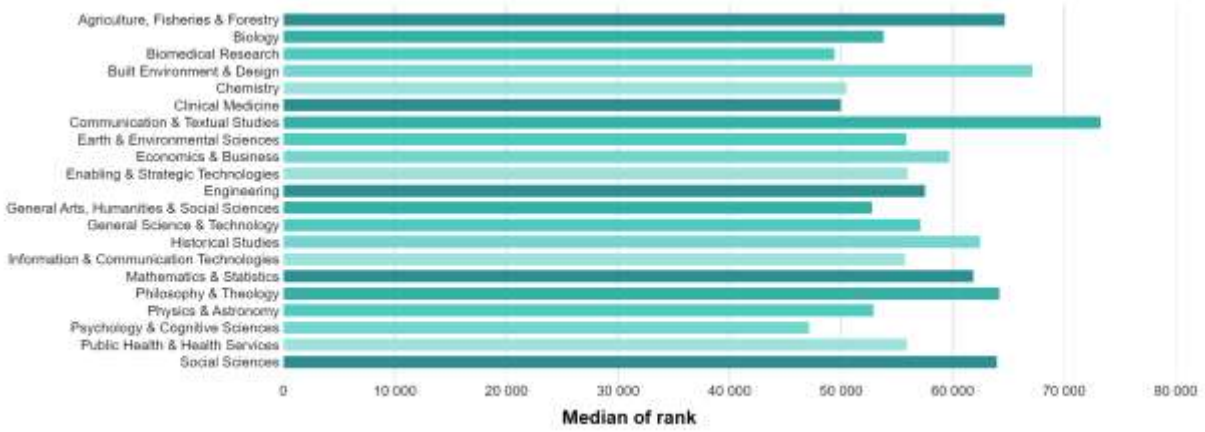


Рисунок 3: Медіана PageRank у різних галузях

Як видно з діаграм на рис. 2-3, іноді індекс Хірша та PageRank дають різні та навіть протилежні результаті. Це видно зокрема на метриках для галузей «Biomedical Research» та «Clinical Medicine», що мають один із найвищих показників Page Rank і при цьому найменший індекс Хірша.

Змоделюємо приклад невідповідності результатів.

Нехай автор a має три статті, кожна з яких цитується 10 разів, а автор b має три статті, кожна з яких цитується по три рази. Індекс Хірша

обох авторів буде однаковим, хоч кількість посилань на статті першого автора є вищою, ніж у другого. Алгоритм PageRank враховує загальну кількість посилань, тому ранг першого автора буде очікувано більшим, ніж у другого.

Змоделюємо також приклад протилежних результатів[5].

Нехай у автора а є одна стаття, з великою кількістю посилань. Незалежно від кількості посилань, індекс Хірша автора, що написав тільки одну статтю, буде рівний 1. У цей же час, інші три автори, у яких по три статті з трьома посиланнями кожна, будуть мати індекс Хірша рівний 3. При цьому PageRank першого автора буде вищим, ніж у всіх інших. Забігаючи наперед скажу, що він буде рівний 0,63 для першого та по 0,04 для трьох інших.

РОЗДІЛ 3. ОПИС РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

3.1. Аналіз технічного завдання

Програма призначена для технічного моделювання та імплементації двох основних досліджуваних алгоритмів ранжування та їх модифікацій.

Зокрема, програма дозволяє виконувати такі обчислення, як:

- PageRank
- h-index
- g-index
- hg-index
- m-quotient
- R-index
- e-index
- i10

Детальний опис цих індексів та формул для їх обчислення можна знайти у Розділі 2.2.

Застосунок являє собою сукупність методів, що приймають на вхід дані про авторів, їх роботи, та цитування робіт одна одною, та обраховують результуючі індекси. При потребі, можна згенерувати звіт з усіма вхідними та вихідними даними у вигляді html-документу.

3.2. Архітектура та процес розробки програми

Проект реалізовано у вигляді Python-застосунку.

Основні функції програми, що експортуються:

- hirsch() – приймає на вхід авторів, їх роботи та матрицю посилань, і повертає масив h-індексів для авторів.
- g_index() – приймає на вхід авторів, їх роботи та матрицю посилань, і повертає масив g-індексів для авторів.
- hg_index() – приймає на вхід авторів, їх роботи та матрицю посилань, і повертає масив hg-індексів для авторів.
- R_index() – приймає на вхід авторів, їх роботи та матрицю посилань, і повертає масив R-індексів для авторів.
- E_index() – приймає на вхід авторів, їх роботи та матрицю посилань, і повертає масив E-індексів для авторів.
- i10() – приймає на вхід авторів, їх роботи та матрицю посилань, і повертає кількість робіт, що мають більше 10-ти посилань для кожного автора.
- time_relative_index() – приймає на вхід авторів, їх роботи, та уже обчислений індекс Хірша, і повертає масив m-quotient для авторів.
- page_rank() – приймає на вхід авторів, їх роботи та матрицю зв’язків між ними, і повертає масиви PageRank для авторів, обчислені на основі середнього та максимального значення PageRank для їх робіт.

Програма також містить ряд допоміжних функцій, зокрема:

- no_self_citations_n_duplicates() – приймає на вхід матрицю зв’язків між документами та усуває дублікати і спроби самоцитування.
- calculate_authors_hirsch() – обчислює індекс Хірша для одного автора.
- calculate_g_index() – обчислює g-index для одного автора.
- calculate_papers_rank() – обчислює PageRank для усіх документів в мережі.
- countP() – обчислює матрицю переходів для ланцюга Маркова.

- print_matrix() та print_indexes() – функції, створені для зручного виведення матриць та масивів у консоль.

Для генерування звіту додано функцію, що містить уже змодельовані вхідні дані, та повертає обчислені індекси для зручного відображення на html-сторінці. Для генерування html-сторінки написано простий сервер на фреймвоку Flask, що викликає функцію та передає дані в html документ. В майбутньому його можна удосконалювати та створити застосунок, що обчислюватиме необхідні індекси чи PageRank на основі заданої матриці посилань.

3.3. Опис файлів даних

Функції приймають на вхід такі основні структури з даними:

- Масив авторів, у нашому випадку це масив string що містить імена авторів.
- Масив документів, кожен документ містить два поля – назву типу string та рік створення типу int. Рік потрібен для обчислення індексу m-quotient.
- Для зв’язку авторів з їх роботами використано структуру мови python – dictionary. У цій структурі даних кожному int id автора відповідає масив з int id документів.
- Граф посилань усіх документів реалізовано у вигляді двовимірного масиву типу int, кожен i-тий рядок якого відповідає вихідним посиланням з i-того документа, а кожен j-ий стовпець відповідає кількості вхідних посилань на j-ий документ. Масив містить значення 0, що вказує на відсутність вихідних посилань, та 1, що вказує на їх наявність. Якщо документ a посилається на документ b більш ніж один раз, у

обрахунках всерівно буде враховано тільки одне вихідне посилання.

Функції для обчислення індексу Хірша та його модифікацій повертають на вихід масиви зі значеннями індексів для усіх авторів впорядкованих по id автора. Функція для обчислення PageRank повертає два масива - PageRank авторів, обчислений як середнє PageRank його документів, та ще один, обчислений за максимальним PageRank серед документів автора.

3.4. Результати виконання програми

Усі результати виконання програми візуалізовано у HTML-сторінці зі звітом(рис.4)

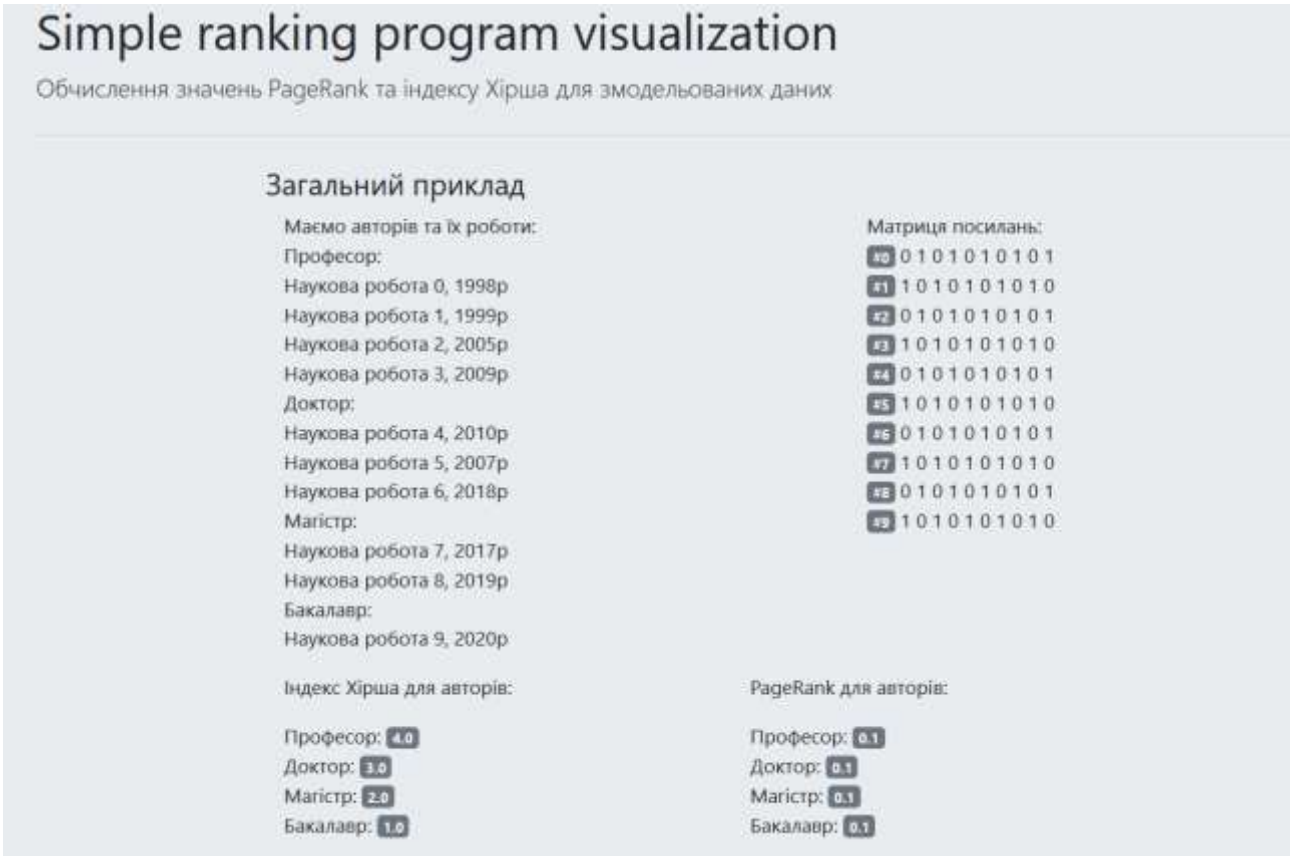


Рисунок 4: Загальний вигляд звіту

Звіт з виконання починається з загального прикладу, де крім основних показників обчислені усі модифікації(рис.5)

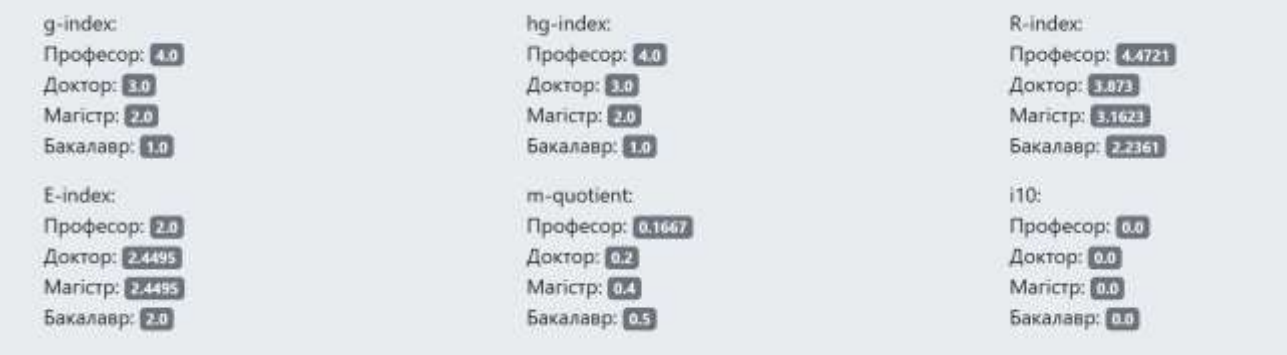


Рисунок 5: Модифікації індексу Хірша в звіті

Наступні приклади містять змодельовані ситуації, що показують невідповідність методик(рис. 6)

Рисунок 6: Змодельовані ситуації невідповідності

ВИСНОВКИ

В ході виконання курсового проекту детально проаналізовано метрики формування рейтингу науковця на основі наукометричних показників та вибрано дві для практичної реалізації - цими метриками стали індекс Хірша та PageRank.

Детально проаналізовано дані, необхідні для обчислення кожного показника, їх переваги та недоліки. Досліджено ряд модифікацій, спрямованих на вдосконалення

Визначено шаблони ситуацій, коли методики дають різні та навіть суперечливі значення. Відтворено так ситуації на змодельованих даних.

При дослідження альтернатив для вищезгаданих методик знайдено один цікавий приклад. У 2016 році науковцями з Southwest China Normal University було запропоновано так-званий PR-індекс, що являє собою сукупність двох досліджуваних мною алгоритмів[8]. Коротко описуючи алгоритм, він обчислюється за аналогією до індексу Хірша, тільки замість цитувань використовує ранг Page Rank для обчислень. Основною ідеєю науковців було те, що цей алгоритм об’єднає два найпопулярніші на даний момент методики, при цьому не збільшуючи складність обчислень.

В ході виконання практичної частини розроблено застосунок, що обраховує значення індексу Хірша та Page Rank для заданої матриці цитувань, а також модифікації індексу Хірша, зокрема g-index, hg-index, R-index, m-quotient та i10. Програма реалізована у вигляді простого веб-сервера, що генерує html-сторінку зі звітом виконання.

Проект має багато шляхів для вдосконалення, зокрема:

- Написання повноцінного застосунку з графічним інтерфейсом, що дозволятиме користувачу обчислювати необхідні індекси, надаючи на вхід матрицю посилань

- Реалізація методів, що коректно обраховують індекси робіт, написаних у співавторстві
- Удосконалення алгоритму PageRank, а зокрема частин, що відповідають за:
    - випадкове блукання
    - обчислення рангу автора на основі рангів його робіт
    - роботу з розрідженими матрицями посилань типу «всі на всіх»

Список використаної літератури

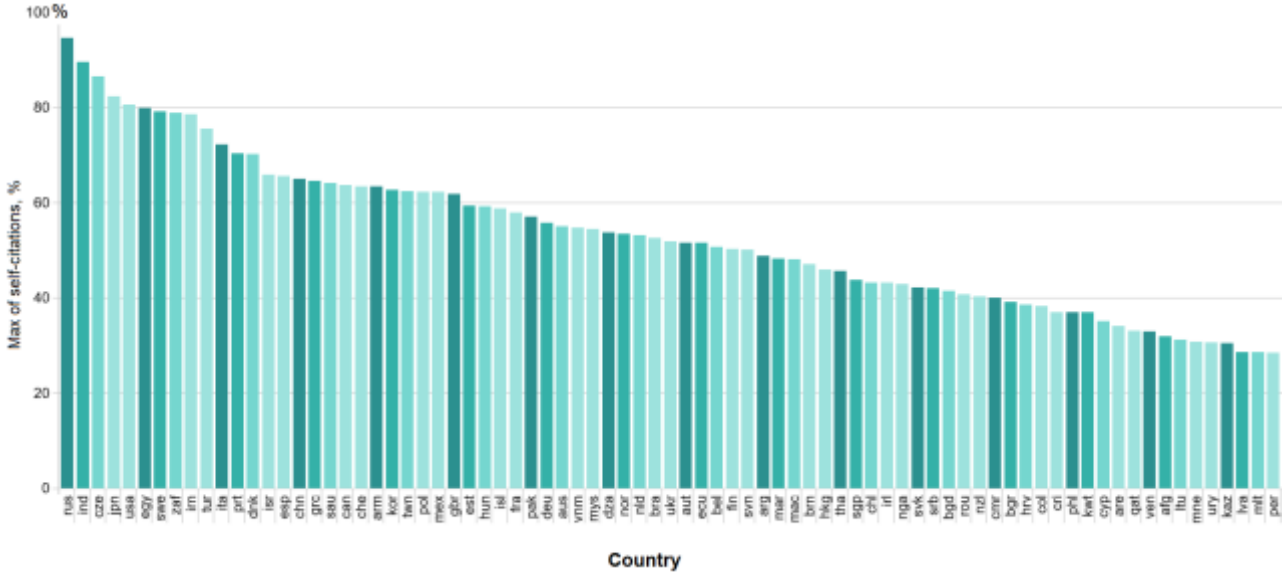
1. Ioannidis JPA, Baas J, Klavans R, Boyack KW "A standardized citation metrics author database annotated for scientific field" *PLoS Biol 17(8): e3000384* August 12, 2019.[Електронний ресурс]. Режим доступу: https://doi.org/10.1371/journal.pbio.3000384
2. David Austin "How Google Finds Your Needle in the Web's Haystack" *American Mathematical Society* December 2006 [Електронний ресурс]. Режим доступу: http://www.ams.org/publicoutreach/feature-column/fcarc-PageRank
3. H-індекс [Електронний ресурс]. Режим доступу: https://uk.wikipedia.org/wiki/H-%D1%96%D0%BD%D0%B4%D0%B5%D0%BA%D1%81
4. Anne-Wil Harzing "Metrics: h and g-index: Publish or Perish tutorial" *Harzing.com* February 6, 2016 [Електронний ресурс]. Режим доступу: https://harzing.com/resources/publish-or-perish/tutorial/metrics/h-and-g-index
5. Григорій Гнатієнко, Олексій Олецький "Порівняння методик ранжування науковців на основі індексів Гірша та PageRank: суперечли-ві ситуації" Сучасні інформаційні технології, №1(1), 2021
6. Штовба С. Д., Штовба О. В "Огляд наукометричних показників для оцінки публікаційної діяльності вченого" Управління великими системами-44[Електронний ресурс]. Режим доступу: https://www.researchgate.net/publication/279535245_Obzor_naukometriceskih_pokazatelej_dla_ocenki_publikacionnoj_deatelnosti_ucenogo
7. Luming Liang, Ronald Rousseau "The R-and AR-indices: Complementing the H-index" *ResearchGate* March 2007 [Електронний ресурс]. Режим доступу: https://www.researchgate.net/publication/257688184_The_R-and_AR-indices_Complementing_the_H-index

8. Herman Scheepers "Markov Chain Analysis and Simulation using Python" *Towards Data Science* November 20, 2019 [Електронний ресурс]. Режим доступу: https://towardsdatascience.com/markov-chain-analysis-and-simulation-using-python-4507cee0b06e
9. Стаття про наукометричні показники, *КПІ імені Ігоря Сікорського* [Електронний ресурс]. Режим доступу: https://webometr.kpi.ua/citation-data
10. Gao C, Wang Z, Li X, Zhang Z, Zeng W "PR-Index: Using the h-Index and PageRank for Determining True Impact" *PLoS ONE 11(9)* July 18, 2016 [Електронний ресурс]. Режим доступу: https://www.researchgate.net/publication/308122194_PR-Index_Using_the_h-Index_and_PageRank_for_Determining_True_Impact

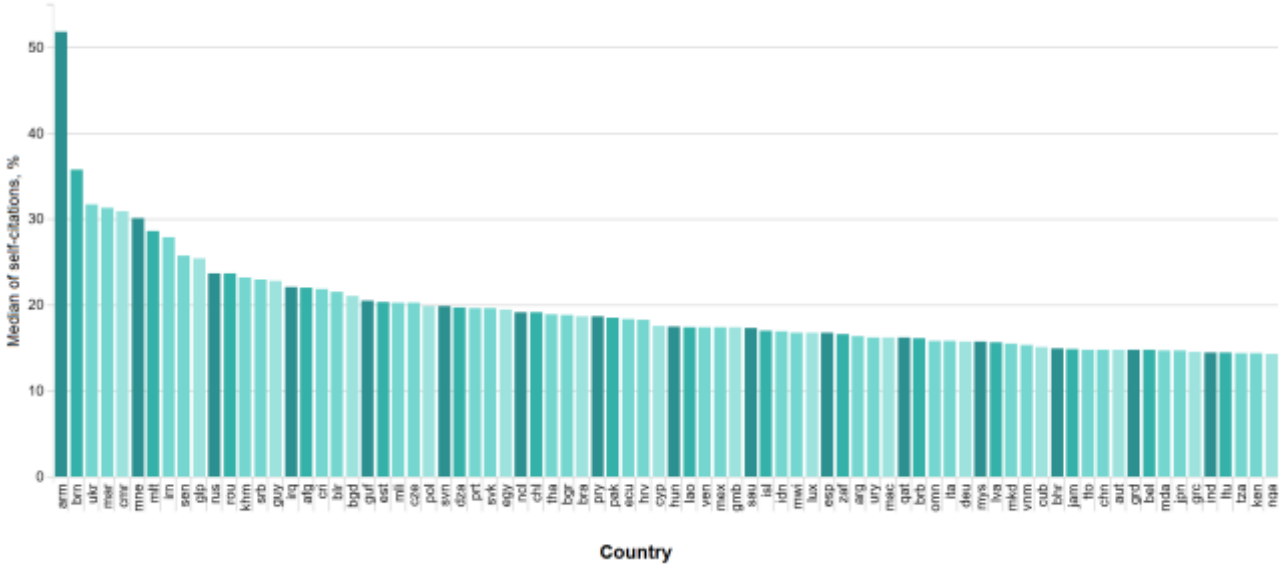
ДОДАТКИ

Додаток 1: Діаграми

(довідниковий)

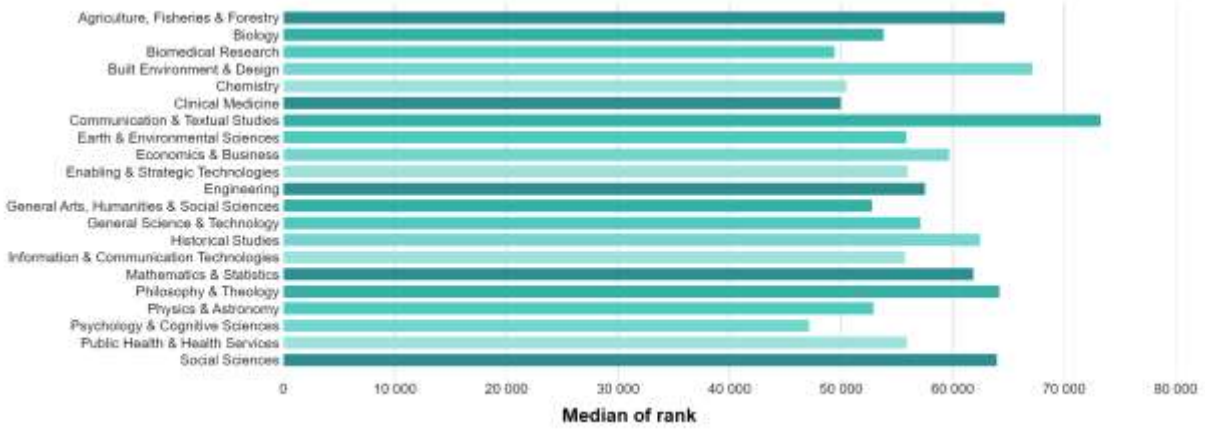


Діаграма 1:Максимальна кількість самоцитувань

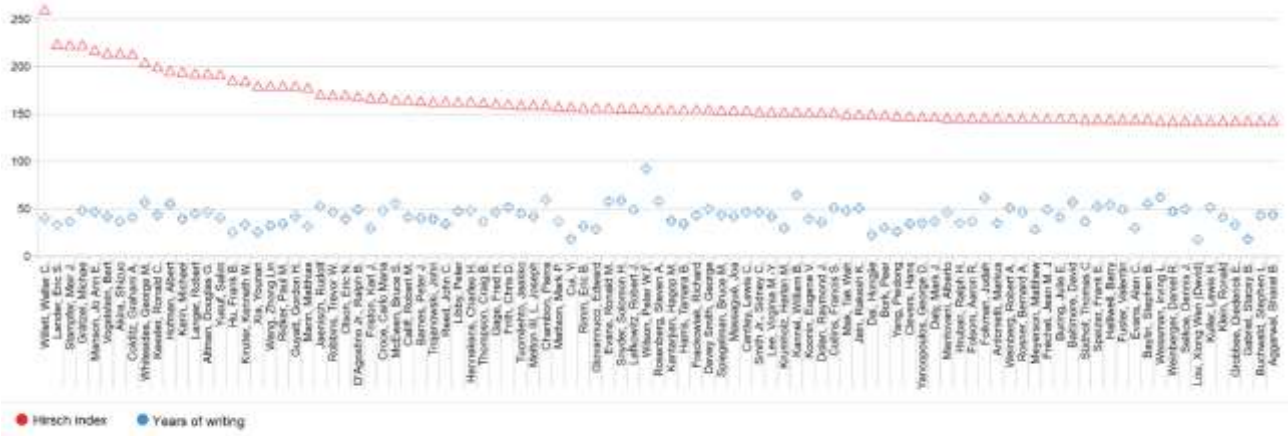


Діаграма 2: Медіана самоцитувань

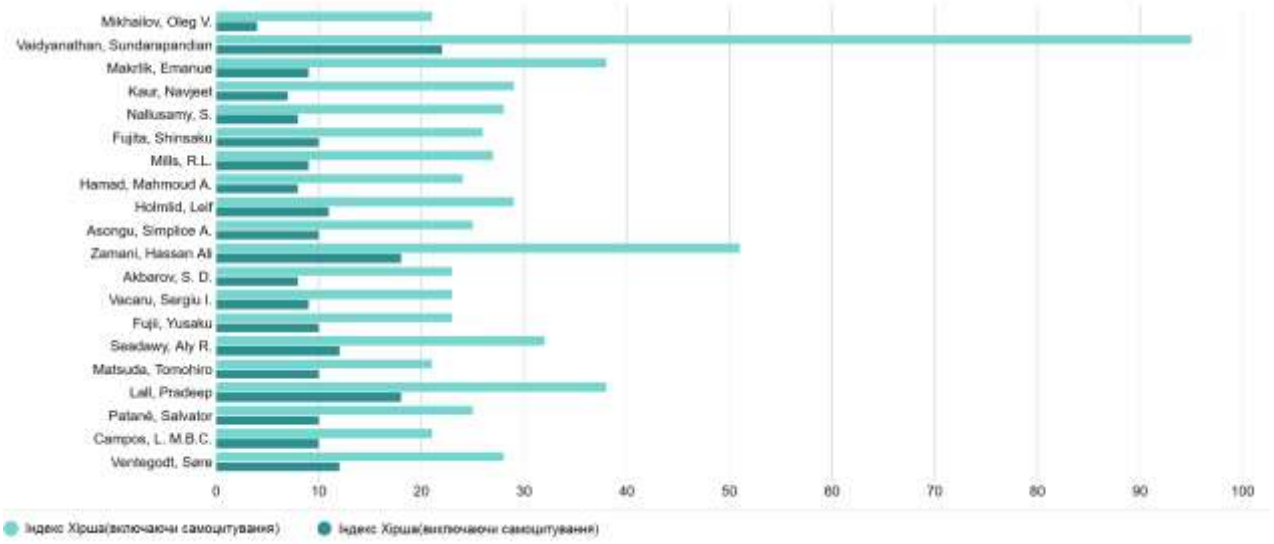
Діаграма 3: Медіана індексу Хірша по сферах діяльності



Діаграма 4:Медіана PageRank по сферах діяльності



Діаграма 5: Залежність індексу Хірша від віку першої публікації



Діаграма 6: Порівняння індексу Хірша з самоцитуваннями та без

Додаток 2: Програмний код

(обов’язковий)

```
import math
import numpy as np
from datetime import date


def hirsch(papers_authors, links):
    paper_links = np.sum(links, axis=0)
    authors_h_index = np.zeros(len(papers_authors))
    for a in papers_authors:
        authors_h_index[a] = calculate_authors_hirsch(papers_authors[a], paper_links)
    return authors_h_index


def no_self_citations_n_duplicates(links):
    for i in range(0, len(links)):
        for j in range(0, len(links[i])):
            if links[i][j] > 0:
                links[i][j] = 1
            if i == j:
                links[i][j] = 0
    return links


def calculate_authors_hirsch(papers, links):
    paper_links = np.array(links)[papers]
    paper_links = -np.sort(-paper_links)
    index = 0

    for i in range(0, len(paper_links)):
        if paper_links[i] == 0:
            diff = 0
        else:
            diff = (i + 1) / paper_links[i]
        if paper_links[index] == 0:
            curr = 0
        else:
            curr = (index + 1) / paper_links[index]
        if abs(diff - 1) < abs(curr - 1):
            index = i

    return index + 1
```

```python
def g_index(papers_authors, links):
    paper_links = np.sum(links, axis=0)
    authors_g_index = np.zeros(len(papers_authors))
    for a in papers_authors:
        authors_g_index[a] = calculate_g_index(papers_authors[a], paper_links)
    return authors_g_index


def hg_index(h, g):
    if len(h) != len(g):
        return
    n = len(h)
    hg = []
    for i in range(n):
        hg.append(math.sqrt(h[i] * g[i]))
    return hg


def calculate_g_index(papers, links):
    paper_links = np.array(links)[papers]
    paper_links = -np.sort(-paper_links)
    index = 0
    for i in range(0, len(paper_links)):
        g2 = (i + 1) * (i + 1)  # i^2
        # if links for i indexes are >= g2 then index=i
        links_sum = np.sum(paper_links[:i + 1])
        if links_sum >= g2:
            index = i

    return index + 1


def calculate_papers_rank(links):
    beta = 0.85
    n = len(links)
    S = []  # вихідні дуги з кожної вершини
    for line in links:
        S.append(np.sum(line))

    # probability matrix P
    transition_matrix = countP(beta, n, links, S)

    initial_dist = np.full((n), 1 / n)

    for _ in range(n * 10):
        update = initial_dist @ transition_matrix
        initial_dist = update

    print("Page rank for every article:\n", initial_dist)
    return initial_dist
```

```
def page_rank(papers_authors, links):
    initial_dist = calculate_papers_rank(links)

    v_max = np.zeros(len(papers_authors))
    v_ave = np.zeros(len(papers_authors))
    for a in papers_authors:
        papers = np.take(initial_dist, papers_authors[a])
        v_max[a] = np.round(np.max(papers), 4)
        v_ave[a] = np.round(np.average(papers), 4)
    return v_max, v_ave


def countP(beta, n, links, S):
    P = []
    for i in range(n):
        pi = []
        for j in range(n):
            if S[i] == 0:
                rij = 0
            else:
                rij = links[i][j] / S[i]
            pij = beta * rij + (1. - beta) / n
            pi.append(pij)
        P.append(pi)
    # print_matrix(P)
    return P

def time_relative_index(papers, papers_authors, hirsch):
    n = len(papers_authors)
    if n != len(hirsch):
        print("err")

    first_paper_time = np.full(n, 10000)
    for a in papers_authors:
        for i in papers_authors[a]:
            first_paper_time[a] = min(first_paper_time[a], papers[i][1])
    m_quotient = np.zeros(n)
    year = date.today().year
    for i in range(n):
        m_quotient[i] = hirsch[i] / (year - first_paper_time[i])
    return np.round(m_quotient, 4)
```

```
def i10(papers_authors, links):
    cites = np.sum(links, axis=1)
    count10 = np.zeros(len(papers_authors))
    for a in papers_authors:
        papers = np.take(cites, papers_authors[a])
        count10[a] = sum(x > 10 for x in papers)
    return count10


def R_index(papers_authors, links):
    cites = np.sum(links, axis=1)
    R = np.zeros(len(papers_authors))
    for a in papers_authors:
        R[a] = math.sqrt(np.sum(np.take(cites, papers_authors[a])))
    return np.round(R, 4)


def E_index(papers_authors, links, hirsch):
    cites = np.sum(links, axis=1)
    E = np.zeros(len(papers_authors))
    for a in papers_authors:
        papers = np.take(cites, papers_authors[a])
        E[a] = math.sqrt(abs(np.sum(papers) - hirsch[a] * hirsch[a]))
    return np.round(E, 4)

def print_matrix(matrix):
    for line in matrix:
        print(np.round(line, 4))


def print_indexes(message, authors, indexes):
    print(message)
    for i in range(0, len(indexes)):
        print(authors[i], " index: ", indexes[i])

papers_authors = {0: [0, 1, 2, 3],
                  1: [4, 5, 6],
                  2: [7, 8],
                  3: [9]}
links = [
    [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 1, 0, 1, 0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 1, 0, 1, 0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 1, 0, 1, 0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 1, 0, 1, 0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 1, 0, 1, 0, 1, 0, 1, 0], ]
```