

УДК 519.6

DOI <https://doi.org/10.32782/2663-5941/2023.4/19>**Трофименко О.Г.**

Національний університет «Одеська юридична академія»

Прокоп Ю.В.

Національний університет «Одеська політехніка»

Ченурна О.Є.

Національний університет «Одеська юридична академія»

Баландіна Н.М.

Національний університет «Одеська юридична академія»

РОЛЬ МАТЕМАТИКИ У РІЗНИХ СФЕРАХ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті розглянуто прикладні аспекти можливого застосування ІТ-фахівцями математичних компетентностей у різних сферах розробки програмного забезпечення. Ступінь використання математики залишається темою активних дискусій між тими, хто виступає за твердий інженерний підхід з математичною строгістю, і тими, хто виступає за більш легкий підхід з мінімізацією математичної складової, особливо під час навчання ІТ-фахівців. У роботі проаналізовано деякі сфери застосування математики для ІТ-фахівців та студентів відповідної галузі: від веброзробки сайтів та вебзастосунків до програмних засобів криптології, баз даних та машинного навчання. З'ясовано, що ступінь застосування математики в ІТ не є рівномірним. Більшість ІТ-фахівців не використовують математичні навички повсякденно. Хоча розробка програм потребує певного розуміння математичного апарату, але це переважно стосується логіки та методу проб і помилок, ніж розуміння абстрактних математичних ідей. З іншого боку, аналіз прикладів застосування математичних алгоритмів та інструментів свідчить про доволі широкий спектр сфер застосування їх в ІТ. Використання математики є критично важливим для областей криптографії, складної візуалізації даних, машинного навчання тощо. Вивчення прикладної математики для ІТ-студентів має важливе значення, оскільки вона навчить використовувати сучасні методи пошуку прихованих зв'язків у великих наборах даних і потенційно може допомогти інженерам-програмістам у розробці високоякісних програмних продуктів, безпечних у використанні. З іншого боку, застосування математики в багатьох інших областях програмної інженерії є менш значним, як-от: тестування програмного забезпечення, веброзробка тощо. Нині не існує універсального підходу щодо того, як інженер-програміст має використовувати математику. Це залежить від конкретної області розробки програмного забезпечення і від того, які спеціалізовані математичні знання необхідні в цій сфері.

Ключові слова: математичні навички, математика в ІТ, розробка програмного забезпечення, дискретна математика, теорія графів, графові бази даних, машинне навчання, криптографія.

Постановка проблеми. Досвід останніх років свідчить про певні складнощі викладання та сприйняття дисциплін математичного спрямування спочатку в школі, а згодом і в коледжах та університетах, навіть для здобувачів ІТ-галузі, де знання математичного апарату є важливим для успішного засвоєння багатьох дисциплін старших курсів, а тому і подальшої успішності кар'єри в ІТ.

Аналіз останніх досліджень і публікацій. Дослідження науковців у цій сфері дещо різняться: від підходу, що «математична освіта занепадає» і «далеко не для всіх ІТ-фахівців варто витратити час на вивчення математики» [1] до

«математика – цариця всіх наук» та «математика є основою ІТ» [2, 3]. У дослідженні [4] зроблено висновок, що більшість ІТ-фахівців не використовують математичні навички повсякденно, і хоча розробка програм потребує певного розуміння математичного апарату, але це здебільшого стосується логіки та методу проб і помилок, ніж розуміння абстрактних математичних ідей. У статті [3] стверджується, що з появою та розвитком машинного навчання, штучного інтелекту, віртуальної і доповненої реальності потреба у володінні математично-аналітичними компетентностями ІТ-фахівцями та здобувачами не те що не

згасає, а виходить на новий рівень затребуваності. Дослідники [5] зосередились на математичних основах квантового обчислення. Автори статті [6] дійшли висновку, що створення моделі машинного навчання залежить від таких математичних понять, як теорія ймовірностей та статистика, адже побудова адекватних моделей і отримання точних результатів має мати мінімальну похибку. Саме тому ймовірність використовується для моделей прогнозування та класифікації, а статистичні дані – для класифікації та вибірки в машинному навчанні. Ступінь використання математики залишається темою активних дискусій між тими, хто виступає за твердий інженерний підхід з математичною строгістю, і тими, хто виступає за більш легкий підхід з мінімізацією математичної складової під час навчання.

Мета статті: проаналізувати сфери застосування математики для IT-фахівців та студентів відповідної галузі в сучасному сценарії мультидисциплінарного підходу шляхом вивчення прикладів практичного застосування математики у розробленні програмного забезпечення різного роду: від веброзробки сайтів та вебзастосунків до програмних засобів криптології, баз даних та машинного навчання.

Виклад основного матеріалу. Веброзробка. Сьогодні веброзробка є одною з найзатребуваніших сфер в IT-галузі. Веброзробник без особливих знань математики може впоратися і з back-end, і з front-end звичайного сайту. Веброзробка – одна зі сфер розробки програмного забезпечення, де потреба в математичних інструментах є найменшою, проте вона залишається актуальною. Так,

наприклад, функція $CSS\ matrix(a, b, c, d, tx, ty)$ [7] визначає однорідну матрицю двовимірної трансформації зображень та блоків (рис. 1).

За потреби вбудованої в сайт ефективної пошукової системи для швидкого відбору даних за певними фільтрами та критеріями з великого набору даних, а обсяги даних для відбору нині зростає щодня [8], виникає доцільність застосування математичного підходу, наприклад, алгоритму, заснованого на частотному ранжуванні термінів, щоб ідентифікувати дублікати, і використовувати словник домену для перевірки відповідного документа для підвищення ефективності пошукової системи. Тому вважати веброзробку позбавленою використання математичних компетентностей є перебільшенням.

Машинне навчання. Всі моделі машинного навчання (Machine Learning, ML) будуються з використанням математичних рішень та ідей, адже ML займається створенням моделей для розуміння мислення. Методи машинного навчання та інтелектуального аналізу даних можуть покращити прогнозування як поширення інформації, так поширення хвороб. Для прогнозування будь-то погоди, будь-то можливих проблем із трафіком потрібне дослідження й аналіз величезних обсягів даних. При цьому потрібне врахування обчислювальних, алгоритмічних та моделюючих проблем, характерних для аналізу великих обсягів даних. Оскільки складні дані і системи можна подати у вигляді зв'язного графа, для виконання завдань класифікації, кластеризації та регресії нейронних мереж можуть бути використані алгоритми графових структур даних. На рис. 2 подано

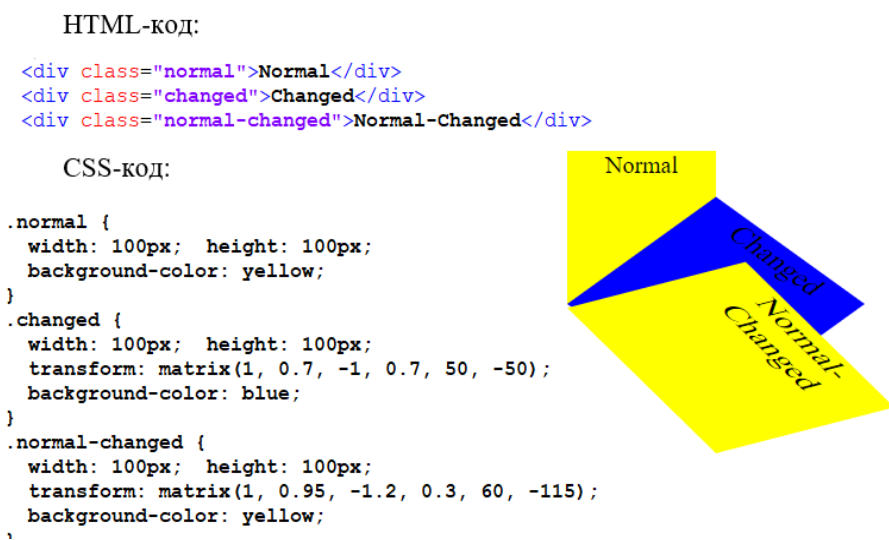


Рис. 1. Приклад використання функції matrix()

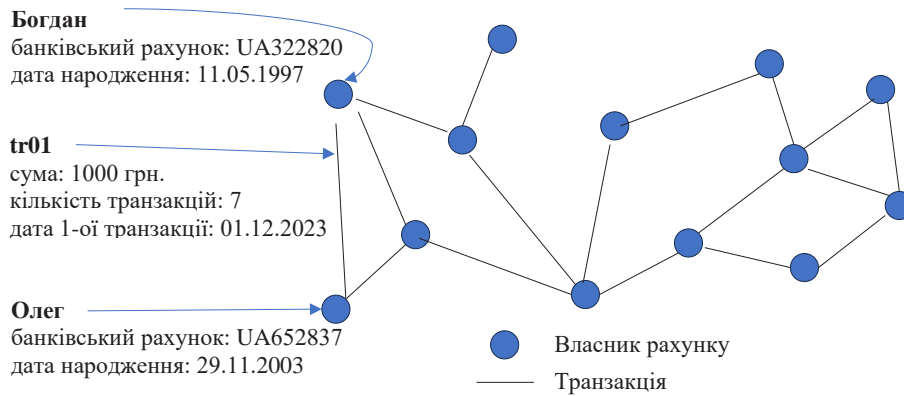


Рис. 2. Неорієнтований граф моделі грошових переказів між клієнтами

приклад моделі мережі клієнтів банку і переказів між ними у вигляді графа.

Завдання методів машинного навчання з графами полягають у вибудовуванні вузлів, їх класифікації і прогнозування можливих зв'язків між вузлами [9]. Аналітика графів зосереджена на дослідженні взаємозв'язків між вузлами у наборі даних. Взаємозв'язки є одними з найсильніших прогностичних елементів поведінки системи. Доповнення наявних процесів машинного навчання функціоналом графів, виявлення у наборах даних про взаємозв'язки дозволяє суттєво покращити процеси машинного навчання [10]. Розуміння графової структури даних дозволяє створювати графові нейронні мережі, як точні моделі машинного навчання, виявляти закономірності взаємозв'язків у даних, масштабувати ці нейронні мережі за допомогою генеративних моделей для графів та робити високоточні прогнози.

Застосування графів в ІТ не обмежується сферою машинного навчання. Одним із найпоширеніших сфер застосувань теорії графів у програмуванні є керування залежностями між модулями, бібліотеками чи то іншими компонентами коду, які необхідні для правильного функціонування програми та уникання проблем і помилок із сумісністю при збільшенні розміру й складності програмних систем [11]. Подання залежності між компонентами у вигляді спрямованого графа дозволяє визначити циклічні взаємозв'язки і вирішити проблеми з їх сумісністю. У тестуванні використання теорії графів надає можливість протестувати всі функції програми за мінімальний час [12]. Ще однією з численних сфер застосування графів в ІТ є проектування баз даних. Окрім суто програмування та програмного забезпечення, графи використовуються для планування робочих ІТ-процесів та проектування систем.

Бази даних. При проектуванні та програмуванні баз даних (БД) потрібні специфічні математичні навички з основ теорії множин, реляційної алгебри, реляційного числення та логіки [13]. Незайвими будуть знання лінійної алгебри для виконання обчислень. Щодо основ реляційної алгебри, то їх використовує мова процедурних запитів на основі різних операцій (об'єднання, перетин, різниця, Декартовий добуток тощо) над таблицями вхідних даних. Наприклад, у БД, що містить інформацію про клієнта, реляційний аспект дозволяє комп'ютерній системі знати, як пов'язати ім'я, адресу, номер телефону та іншу відповідну інформацію клієнта. Це все робиться за допомогою дискретної математичної концепції множин. Набори дозволяють групувати та впорядковувати дані. Оскільки кожна частина інформації і кожна ознака, що належить до цієї частини інформації, є дискретними, організація такої інформації в базі даних вимагає дискретних математичних методів.

Іншим цікавим прикладом математичного апарату є графові БД, які використовують графові структури для зберігання даних і вирішують проблеми реалізації зв'язків «багато-до-багатьох». Структура графа дозволяє обійти обмеження реляційних БД, пропонує концептуальне подання даних, тісно пов'язаних із реальним світом, оскільки приділяє багато уваги стосункам між даними. Так, при моделюванні телекомунікаційних систем та мережевих послуг вузлами графа можуть бути: фізичні пристрої (маршрутизатори, сервери, комутатори), програмні рішення, діяльність (дзвінки користувачів, передача медіа), інформація про клієнтів (їхні права, підписки) тощо [14]. Такі БД дозволяють просто моделювати складні зв'язки між даними. На рис. 3 подано графову модель соціальних стосунків певної групи людей.

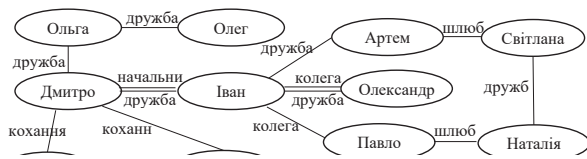


Рис. 3. Неорієнтований граф моделі соціальних стосунків

Графові БД є нереляційними (NoSQL). В силу своєї природи ці БД мають гнучку, адаптивну структуру і при цьому задають чітке подання взаємозв'язків між даними. Вони пропонують просту, масштабовану модель даних, що дозволяє відстежувати цифрові ресурси: документи, розрахунки, контракти тощо. Прив'язка всіх даних до однієї загальної точки забезпечує сталість і точність даних, що дуже важливо для великомасштабних компаній світового рівня. Інформація передається через мережу шляхом пошуку оптимальних маршрутів, і це робить графові бази даних оптимальним варіантом для впровадження маршрутизації. Такий підхід може стати в пригоді під час аналізу профілів і підписок користувачів соціальних мереж. Також ця технологія може використовуватися для аналізу економічної активності контрагентів для виявлення різних схем шахрайства. Наприклад, можна відстежити використання певних рахунків, карт чи реквізитів контрагентів у різних операціях [15]. На рис. 4 наведено графову модель використання фінансових операцій з одного банківського рахунку на інший, де банківські рахунки є вузлами графа, а банківські операції – зв'язками.

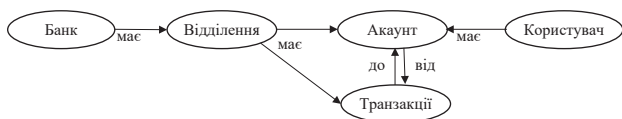


Рис. 4. Орієнтований граф моделі банківських операцій

Графові бази даних довели свою успішність в управлінні операційними та бізнес-системами, а також в управлінні Інтернетом речей, в аналізі шахрайств та в аналізі відповідності нормативним вимогам [16].

Криптографія як наука про структури безпеки електронних паролів базується переважно на дискретній математиці, хоча використовує багато понять із теорії чисел, алгебри та теорії ймовірностей. Частково це пов'язано з тим, що комп'ютери надсилають інформацію дискретно у формі бітів [17, 18].

Теорія чисел як одна з важливих частин дискретної математики дозволяє криптографам створювати безпечні паролі та методи шифрування паролів. Саме теорія чисел є основою сучасної криптографії з відкритим ключем [19]. Модульна арифметика ($x \bmod N$), найбільший спільний дільник, розширені алгоритми Евкліда широко використовуються в криптографії, особливо при обчисленні зворотного мультиплікативу по модулю (коли цілі числа a і b є взаємно простими) для виведення ключових пар шифрування RSA з відкритим ключем. Модульна арифметика використовується для створення математичної основи алгоритмів шифрування та дешифрування. Вона забезпечує спосіб виконання арифметичних операцій над великими числами, гарантуючи, що результат буде в межах певного діапазону.

Алгебра використовується в сучасній криптографії для створення криптографічних алгоритмів із симетричним ключем, які є одним із найшвидших і найпростіших способів шифрування повідомлень. Криптографія з симетричним ключем передбачає використання одного секретного ключа для шифрування та дешифрування повідомлень. Найпоширенішим криптографічним алгоритмом із симетричним ключем є Advanced Encryption Standard (AES), який базується на математичній структурі, яка називається кінцевим полем. Інші алгебраїчні структури, такі як групи та кільця, також використовуються для створення криптографічних алгоритмів із симетричним ключем [20].

Теорія ймовірностей використовується в криптографії для створення безпечних протоколів обміну ключами та автентифікації повідомлень. Наприклад, протокол обміну ключами Діффі-Хеллмана використовує задачу дискретного логарифмування в кінцевих полях. Безпека цього протоколу базується на складності обчислення дискретних логарифмів і на припущенні, що обчислювально неможливо вирішити певні математичні проблеми.

Криптографічні функції гешування по суті є математичним алгоритмом для незворотного перетворення тексту, що повсюдно використовуються для безпекової автентифікації користувачів і зберігання паролів. Існують різні алгоритми гешування: від MD5 і SHA-1, які раніше використовувалося для автентифікації та зберігання паролів, а зараз в силу своєї специфіки не є небезпечними, а через це вважаються застарілими, до сучасних PBKDF2, Argon2, bcrypt та scrypt. Саме PBKDF2 має підтвердження відповідності стандарту безпеки FIPS-140 і його рекомендує засто-

совувати NIST у своїх рекомендаціях зі зниження організаційних ризиків кібербезпеки [21].

Приклад програмного гешування пароля засобами алгоритму PBKDF2 мовою Python:

```
import hashlib
import os

salt = os.urandom(32)
password = 'mypassword'

key = hashlib.pbkdf2_hmac(
    'sha256', # The hash digest algorithm for HMAC
    password.encode('utf-8'), # Convert the password to bytes
    salt, # Provide the salt
    100000, # It is recommended to use at least 100,000 iterations of SHA-256
    dklen=128 # Get a 128 byte key
)

storage = salt + key # Store them

# Getting the values back out
salt_from_storage = storage[:32] # 32 is the length of the salt
key_from_storage = storage[32:]
```

Приклад Python-коду для верифікації пароля, наданого користувачем:

```
import hashlib

salt = b" # Get the salt you stored for *this* user
key = b" # Get this users key calculated
password_to_check = 'password246' # The password provided by the user

# Use the exact same setup you used to generate the key,
# but this time put in the password to check
new_key = hashlib.pbkdf2_hmac(
    'sha256',
    password_to_check.encode('utf-8'), # Convert the password to bytes
    salt,
    100000
)

if new_key == key:
    print('Password is correct')
else:
    print('Password is incorrect')
```

Математичні алгоритми гешування PBKDF2, Argon2, bcrypt і scrypt додатково застосовують засолювання (Salting). Сіллю (Salt) виступає випадково згенероване значення, яке зазвичай зберігається у базі даних із гешем пароля і призначене для унеможливлення використання геш-таблиць для зламу паролів. Додаткову безпеку гешування надає додавання перцю (Pepper) – статичного значення, яке має бути секретним і зберігатися окремо від бази даних для уникнення втрати даних у разі компрометації бази даних. Перець потрібно генерувати заново для кожної програми, інакше злам однієї програми призведе до зниження безпеки іншої. До речі, генерування випадкових чисел відбувається також за допомогою математичних алгоритмів.

Висновки. Аналіз прикладів математичних алгоритмів та інструментів свідчить про доволі широкий спектр застосування їх у сфері розробки програмного забезпечення. Не даремно Едсгер Дейкстра стверджував, що інженерія програмного забезпечення є розділом математики [22]. Нині спектр розв’язуваних задач в ІТ-галузі все частіше потребує математичних знань для програмної реалізації задач побудови математичних моделей та їх розв’язків, дослідження статистичних залежностей, аналізу великих даних, розпізнавання образів, побудови нейронних мереж тощо.

Через певні складнощі вивчення математичних дисциплін на молодших курсах навчання бакалаврату ІТ-спеціальностей і нерозуміння здобувачами того, де саме і як їм будуть корисні відповідні математичні знання, у багатьох студентів виникають певні проблеми з цими освітніми компонентами. Для усунення подібного роду проблем викладачам відповідних математичних дисциплін варто самим знати сфери можливого використання тих чи інших математичних інструментів та алгоритмів, щоб зацікавити здобувачів у вивченні своїх дисциплін, пояснивши їм потребу вивчення того чи іншого математичного засобу на конкретних прикладах їх подальшого можливого використання його у майбутній професії. Сучасним викладачам математики у вишах варто вибудувати курси відповідно до потреб здобувачів і з наведенням відповідних прикладів та пояснень. Адже студенти, які прийшли навчитися програмувати, на молодших курсах не розуміють, де у професії їм може знадобитися вивчення основ теорії множин, реляційної алгебри, реляційного числення, логіки тощо. Тому ставляться до вивчення цих дисциплін поверхнево. Свідоме проходження відповідних курсів із розумінням сфер можливого застосування відповідного математичного апарату формуватиме як розуміння важливості ролі математики, так якісного ІТ-фахівця, який зможе безпроблемно на практиці скористатись відповідними набутими під час навчання знаннями. Проте це потребує певної самостійної перепідготовки викладачів відповідних математичних дисциплін. Адже поширено у вишах України їх викладання покладено на класичних математиків, які мають академічну математичну освіту і не розуміються на програмуванні, а отже їм складно проводити паралелі між тим чого, вони навчають, і тим, як це застосовувати у програмуванні. Тут корисними можуть бути курси підвищення кваліфікації таких викладачів на ІТ-підприємствах, які допоможуть усунути відповідні прогалини.

Загалом вивчення прикладної математики для ІТ-студентів має важливе значення, оскільки вона навчає використовувати сучасні методи пошуку прихованих зв'язків у великих наборах даних і потенційно може допомогти інженерам-програмістам у розробці високоякісних програмних продуктів, безпечних у використанні. При цьому ступінь застосування математики в ІТ не є рівномірною. З одного боку, використання математики є критично важливим для областей криптографії,

складної візуалізації даних, машинного навчання тощо. А з іншого, її застосування в багатьох інших областях програмної інженерії є менш значним, як-от: тестування програмного забезпечення, веброзробка тощо. Не існує універсального підходу щодо того, як інженер-програміст повинен використовувати математику. Це залежить від конкретної області, в якій інженер-програміст працює, і від того, які спеціалізовані математичні знання необхідні в цій сфері.

Список літератури:

1. Gopal T.V. Teaching mathematics with the software engineering body of knowledge [SWEBOK]. *Scientific conference with international participation (STEMEDU-2022)*. Veliko Tarnovo, Bulgaria, 2022. P. 8–12.
2. Dhiraj Y. Application of Mathematics in Computer Science. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*. 2021. Vol. 10(1). P. 339–344. <https://doi.org/10.48175/IJARSCT-2051>
3. Bhadane S., Kothawade M., Shinde M., Sulakhe V. Applications of Mathematical in Computer Science. *International Journal for Research in Applied Science and Engineering Technology*. 2023. Vol. 11. P. 705–709. <https://doi.org/10.22214/ijraset.2023.48667>
4. Do You Need Math for IT (Information Technology)? URL: https://campusleaders.org/math-requirements-for-information-technology/#google_vignette
5. Fleury G., Lacomme Ph. Mathematical Foundations of Quantum Computing for Computer Science Researchers. *arXiv preprint arXiv:2303.01938*. 2023. P. 1–20. <https://doi.org/10.48550/arXiv.2303.01938>
6. Patil M., Jadhav S., Talekar S., Bag V. (2023). The role of mathematics in machine learning. *Journal of Data Acquisition and Processing*. 2023. Vol. 3(1). P. 1062–1073. <https://doi.org/10.5281/zenodo.7702430>
7. matrix(). URL: <https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/matrix>.
8. Mistry K., Desai Ch, S. Lal S., Patel K. A Mathematical Method with Word Frequency Sorting for Mining Web Content Outliers. *Journal of Theory and Practice of Engineering Science*. 2021. Vol. 1. Issue 1. [https://doi.org/10.53469/jtpes.2021.01\(01\).01](https://doi.org/10.53469/jtpes.2021.01(01).01)
9. Hamilton W., Ying R., Leskovec J. Representation Learning on Graphs: Methods and Applications. *Social and Information Networks: Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*. 2017. Vol. 40. P. 52–74. <https://doi.org/10.48550/arXiv.1709.05584>.
10. Needham M., Hodler A. Graph Algorithms: Practical Examples in Apache Spark and Neo4j. 1st Edition. O'Reilly, 2019. 265 p. ISBN-10 1492047686.
11. Isuru U., Indika P. Using dependency graph and graph theory concepts to identify anti-patterns in a microservices system: A tool-based approach. (2021). *IEEE Moratuwa Engineering Research Conference (MERCCon)*. 2021. P. 699-704. <https://doi.org/10.1109/MERCCon52712.2021.9525743>.
12. Nados W., Alsaedi R., Kadhim Q. The application of the graph theory for developing and testing the software. *International Journal of Civil Engineering and Technology (IJCIET)*. 2019. Vol. 10. Issue 01. P. 2026–2031.
13. Організація баз даних : навч. посібник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, І. М. Копитчук. Одеса : Фенікс, 2019. 246 с.
14. Nagothu D., Dolgikh A. iCrawl: A Visual High Interaction Web Crawler. *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*. 2017. P. 91–103. https://doi.org/10.1007/978-3-319-65127-9_8.
15. Moscisk L. Do graph databases scale. URL: <https://morioh.com/a/bcdfb1496c12/do-graph-databases-scale>.
16. A Guide to Graph Databases. URL: <https://www.influxdata.com/graph-database>
17. Mathematics in Cryptography: Part 1. URL <https://medium.com/dsc-sastra-deemed-to-be-university/mathematics-in-cryptography-part-1-3749e5b354c>
18. Ma J. Basic application of mathematics in cryptography. *IEEE International Conference on Modern Education and Information Management (ICMEIM)*. 2020. P. 871–875. <https://doi.org/10.1109/ICMEIM51375.2020.00192>.
19. Wenchao Sh. Development of Number Theory and the Application in Cryptography. *Theoretical and Natural Science*. 2023. Vol. 2. P. 188–193. <https://doi.org/10.54254/2753-8818/2/20220139>.
20. Sethumadhavan K. PImplementing cryptography with Python. URL: <https://blog.logrocket.com/implementing-cryptography-python/>
21. NIST Special Publication 800-63B. Digital Identity Guidelines. URL: <https://pages.nist.gov/800-63-3/sp800-63b.html#memsecretver>
22. Tadepalli G. Teaching Mathematics with the Software Engineering Body of Knowledge. *Innovative STEM Education*. 2022. Vol. 4. P. 8–12. <https://doi.org/10.55630/STEM.2022.0401>.

Trofymenko O.H., Prokop Yu.V., Chepurina O.Ye., Balandina N.M. THE ROLE OF MATHEMATICS IN DIFFERENT AREAS OF SOFTWARE DEVELOPMENT

The article explores the applied aspects of the possible application of mathematical competencies by IT specialists in various areas of software development. The extent to which mathematics is used remains a topic of active debate between those who favour a hard engineering approach with mathematical rigour and those who favour a lighter approach with minimal maths in IT training. The paper analyses some areas of application of mathematics for IT professionals and students of the relevant field: from web development to software tools of cryptology, databases, and machine learning. It was found that the degree of application of mathematics in IT is not uniform. Most IT professionals do not use math skills daily. Although software development requires some understanding of the mathematical apparatus, it is more about logic and trial and error than understanding abstract mathematical ideas. On the other hand, the analysis of examples of the application of mathematical algorithms and tools indicates a wide range of areas of their application in IT. The use of mathematics is critical to the fields of cryptography, complex data visualization, machine learning, and more. The study of applied mathematics is important for IT students because it teaches them how to use modern techniques to find hidden relationships in large data sets. It can potentially help software engineers develop high-quality software products that are safe to use. In addition, the application of mathematics in many other areas of software engineering is less significant, such as software testing, web development, etc. Currently, there is no universal approach to how a software engineer should use mathematics. It depends on the specific area of software development and what specialized mathematical knowledge is needed in that area.

Key words: *mathematical skills, mathematics in IT, development software, discrete mathematics, graph theory, graph databases, machine learning, cryptography.*