

АСПЕКТИ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ НА ПРИКЛАДІ СКРИПТІВ JAVASCRIPT НА ПЛАТФОРМІ
MOODLE

Анотація: Дослідження присвячене застосуванню інформаційно-комунікаційних технологій у навчальному процесі, зокрема використанню скриптів JavaScript на платформі Moodle для викладання дискретної математики. Впровадження цифрових технологій у навчання сприяє інтерактивності, адаптивності та автоматизації оцінювання. Особливу увагу приділено алгоритму Форда-Фалкерсона для знаходження максимального потоку в графах та його практичному застосуванню у моделюванні транспортних і комунікаційних мереж. Використання бібліотек JavaScript, таких як Vis.js та Chart.js, допомагає ефективно візуалізувати графові структури. Moodle надає можливість інтегрувати кастомні плагіни для алгоритмічних розрахунків, що підвищує якість навчання. Впровадження JavaScript у навчальний процес не лише економить час викладачів і студентів, а й сприяє глибшому засвоєнню складних математичних концепцій та підвищує рівень цифрової грамотності студентів.

Ключові слова: інформаційно-комунікаційні технології, JavaScript, Moodle, дискретна математика, візуалізація графів, алгоритм Форда-Фалкерсона, цифрова грамотність

Інформаційно-комунікаційні технології суттєво впливають на всі аспекти суспільного життя, включаючи освіту, бізнес, науку та повсякденне спілкування. Їх активний розвиток не лише змінює способи обробки й передачі інформації, а й сприяє трансформації навчальних методів, які використовуються у вищих навчальних закладах.

Цифрові технології дозволяють студентам та викладачам миттєво отримувати доступ до великої кількості освітніх матеріалів, включаючи наукові статті, навчальні відео, інтерактивні курси та онлайн-бібліотеки. З використанням ІКТ можна створювати адаптивні навчальні програми, які підлаштовуються під потреби студентів, їхні здібності та рівень знань. Сучасні технології дозволяють краще пояснювати складні концепції за допомогою мультимедійних презентацій, симуляцій, віртуальної реальності та доповненої реальності. Навчальні платформи допомагають викладачам автоматично перевіряти завдання, вести оцінювання та зворотний зв'язок у режимі реального часу. Розширюються можливості дистанційної освіти – студенти можуть навчатися онлайн без фізичної прив'язки до місця навчання, що особливо важливо для міжнародних програм та під час кризових ситуацій.

Moodle — це потужна платформа для управління навчанням (LMS), яка широко використовується у закладах освіти для організації дистанційного та змішаного навчання. Вона дозволяє викладачам створювати інтерактивні курси, проводити тестування, організовувати дискусії та відстежувати успішність студентів.

Використання скриптів JavaScript у викладанні дискретної математики з використанням платформи Moodle може покращити освітній процес, зробивши його більш інтерактивним та практичним. JavaScript дозволяє створювати веб-додатки для візуалізації математичних концепцій, автоматизації перевірки завдань. JavaScript чудово підходить для створення автоматизованих систем оцінювання та перевірки завдань. Використання JavaScript допомагає викладачам і студентам економити час, підвищує об'єктивність оцінювання та знижує ймовірність людських помилок. Автоматична перевірка дозволяє миттєво отримувати результати. Об'єктивність – уникнення суб'єктивних факторів в оцінюванні. Масштабованість – можливість перевіряти велику кількість студентських робіт одночасно.

Таблиця 1

Застосування JavaScript у візуалізації

Генерація графів	Побудова таблиць істинності	Візуалізація дерев та мереж
Використання бібліотеки Vis.js для малювання графічних об'єктів.	Автоматичне генерування логічних таблиць істинності на	JavaScript може зчитувати графові структури з бази даних і

	основі введених логічних виразів.	передавати їх до JavaScript для рендерингу.
Використання vis.min.css-бібліотеки, таких як Chart.js або D3.js, у зв'язці з JavaScript для створення інтерактивних графів	Використання HTML і CSS для стилізації та зручного представлення даних	Використання бібліотеки Vis.js для автоматичної побудови графів і дерев.

Для реалізації алгоритмів та візуального супроводження були використані 1) для графів бібліотеки "vis.min.js" - для відображення графів, 2) "vis-network.min.css" для задання візуалу графам. Другою перевагою скриптів JavaScript в Moodle є візуалізація математичних структур – за допомогою JavaScript можна генерувати графи, таблиці істинності та інші математичні об'єкти. JavaScript дозволяє створювати та візуалізувати математичні структури, що особливо корисно при викладанні дискретної математики. За допомогою бібліотек та інструментів можна генерувати графи, діаграми, таблиці істинності та інші математичні об'єкти. Крім того завдяки Vis.js можна задавати координати вузлів вручну (наприклад, через x і y), додавати мітки (labels) для пропускної спроможності та налаштовувати вигляд ребер (стрілки, криві лінії тощо). Це забезпечує інтерактивність, наприклад, масштабування чи перетягування вузлів, що полегшує сприйняття графа студентами. Стосовно досвіду застосувань скриптів до теми алгоритм Форда - Фалкерсона пошуку максимального потоку в графі.

Moodle дозволяє додавати кастомні плагіни для інтеграції алгоритмічних розрахунків. Використання зовнішніх програмних пакетів для автоматизованої перевірки кодів студентів.

The screenshot shows the Moodle LMS interface. At the top, there is a header with the Moodle logo, the text "MOODLE3 ЧНУ імені Петра Могили", and navigation links: "Зв'язок з адміністратором", "Документи", and "Корисні посилання". Below the header, there is a sidebar with a "Загальне" (General) section and a "Зміст завдання" (Task content) section. The main content area displays the task title "Практичне 12. Алгоритм Форда Фалкерсона" and a rich text editor. The code editor shows the following HTML code:

```

1 <p dir="ltr"><span style="font-size: 0.9375rem;"></span><br></p>
2
3 <table>
4   <tbody>
5     <tr>
6       <th>p:&nbsp;</th>
7       <td><input id="rsa_p" type="number" min="1"></td>
8     </tr>
9     <tr>

```

Рис. 3. Подання на платформі Moodle

Потокові графи широко використовуються для розв'язування практичних задач. Однією (дуже поширеною!) задачею є задача про максимальний потік: у поточковому графі знайти максимальний потік, який може бути переданий з витіка до стока. Прикладом такої задачі є знаходження максимальної пропускної спроможності комп'ютерної мережі. Задача про максимальний потік розв'язується на основі теореми Форда - Фалкерсона про максимальний потік у графі: величина максимального потоку у графі дорівнює величині пропускної спроможності його мінімального розрізу.

Транспортна мережа – це технічний об'єкт, яким розповсюджуються потоки. Прикладами таких мереж є водогінна мережа, комп'ютерна мережа, мережа автомобільних доріг тощо. Потік - це кількість речовини, енергії або інформації, яка проходить через переріз за одиницю часу.

Моделлю транспортної мережі є зважений граф, у якому задані потоки $v(x,y)$ і пропускні спроможності ребер $c(x,y)$ - максимальні кількості потоку, які можуть проходити через ребра від витіку до стоку, де x - початкова вершина ребра, y - кінцева вершина. Значення потоків визначаються функцією потоку $f(x,y)$, причому: $f(x,y) \leq c(x,y)$.

Приклад мережі показаний на рис. 1. На рисунку кожне ребро охарактеризоване пропускнуою здатністю (перша цифра) і величиною потоку (друга цифра). Наведена мережа має один виток, один сток і 4 проміжних вузлів.

Для розуміння прикладних застосувань у завданні є різні типи умов.

Завдання по алгоритму Форда-Фалкерсона

Завдання 1: Система автодоріг

Система автодоріг, що проходять через область, може забезпечити пропускні спроможності, показані на графі (тисячі автомашин на годину).

Питання:

1. Який максимальний потік через цю систему (тис. автомашин на годину)?
2. Скільки автомашин на годину повинно проїхати по дорозі 5-6, щоб забезпечити максимальний потік?

Номер варіанту (0-30):



Максимальний потік (тис. автомашин/годину): Потік дорогою 5-6 (тис. автомашин/годину):

Вірно! Обидві відповіді вірні.

Рис. 1. Приклад мережі для завдань

В мережі існує три типи вузлів:

- джерело(виток) S , з якого виходить більше потоку, ніж входить в нього;
- сток T , в який входить більше потоку, ніж виходить з нього;
- проміжні вузли, в які скільки виходить потоку, стільки ж і входить.

Розглянемо *метод Форда-Фалкерсона* визначення максимального потоку, йдеться про метод, оскільки існує декілька алгоритмів, які його реалізують.

Таблиця 2

Метод Форда-Фалкерсона визначення максимального потоку

Крок 0. Для кожної дуги орграфа $(k,i) \in E$ встановлюють $x_{k,i} = 0$.

Крок 1. Визначають збільшувальний шлях в орграфі.

Крок 2. Якщо збільшувального шляху не існує, то потік оптимальний (завершення роботи). У протилежному випадку визначають резерви збільшення/зменшення потоків дугами, які утворюють збільшувальний шлях.

Крок 3. Обчислюють значення D , яке дорівнює мінімальному з резервів збільшення/зменшення потоків дугами.

Крок 4. Збільшують потік прямими дугами збільшувального шляху на значення D .

Крок 5. Зменшують потік зворотними дугами збільшувального шляху на значення D .

Крок 6. Переходять на перший крок.

Завдання 2

Завдання 3

Завдання 4

Завдання 5

Завдання 6

Завдання 6: Мережа хімічних труб

Хімічний завод має модель з труб, представлених двома переточеними різними лініями, щоотримують з однієї частини підприємства в іншій. Мережа труб і пропускну спроможності (тис. автомашин на годину) показані на малюнку.

Питання:

1. Який максимальний потік для системи, якщо завод збирається перевести з вузла 1 до вузла 9 стільки рівних хімікатів, скільки це можна?
2. Скільки хімікатів буде надходити через секцію 3-5?

Номер варіанту (0-30):

Максимальний потік (тис. автомашин/годину): Потік по секції 3-5 (тис. автомашин/годину):

Вірно! Обидві відповіді вірні.

Модель мережі з природними умовами для алгоритму Форда-Фалкерсона — це підхід до розв’язання задачі максимального потоку в мережах з урахуванням реальних обмежень, таких як змінна пропускна здатність, втрати на маршруті та інші природні фактори. Алгоритм Форда-Фалкерсона, класично

використовуваний для знаходження максимального потоку в графах, може бути модифікований для роботи з реалістичними умовами мережевого середовища. Недоліком алгоритму є залежність від вибору збільшувального шляху, потенційно довгий час роботи.

Ця тема є актуальною для аналізу транспортних, комунікаційних та водних мереж, а також для оптимізації потоків у складних реальних системах.

Таблиця 3

Моделі мережі з природніми умовами

Завдання 2

Завдання 3

Завдання 4

Завдання 5

Завдання 6

Завдання 2: Система автодоріг

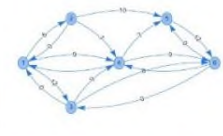
Система автодоріг, що проєктує мережу, може забезпечити пропускову спроможність, показану на малюнку (система автодоріг на малюнку).

Питання:

1. Який максимальний потік через цю систему (тис. автомобілів на годину)?

2. Скільки автомобілів на годину повинно проїхати по дорозі 5-6 щоб забезпечити максимальний потік?

Номер варіанту (0-30):



Максимальний потік (тис. автомобілів на годину):

Потік по дорозі 5-6 (тис. автомобілів на годину):

Вірно! Обидві відповіді вірні.

Завдання 3: Телефонна мережа

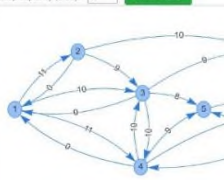
Телефонна компанія використовує підземну кабельну мережу для забезпечення високої якості зв'язку між великими містами (вузли 1 і 7 мережі). Переговори здійснюються через серію кабельних гілок і з'єднують їх вузли мережі, як це показано на малюнку. На малюнку показано також число телефонних переговорів (тис.), яке допускається одночасно в будь-якій точці часу.

Питання:

1. Яка максимальна кількість телефонних переговорів між двома містами, яка може бути допущена одночасно (тис. шт.)?

2. Яке число телефонних переговорів має забезпечуватися кабелем 4-7?

Номер варіанту (0-30):



Максимальний потік (тис. переговорів):

Потік по кабелю 4-7 (тис. переговорів):

Вірно! Обидві відповіді вірні.

Завдання 4: Мережа нафтопроводів

Нафтова компанія вводить мережу нафтопроводів, через яку нафта перекачується від родовищ до нафтозастав. Частина цієї мережі проєктована на малюнку (пропускова здатність нафтопроводів показана в тис. т/год).

Питання:

1. Яка компанія може поставити нафту в склади 7 і наскільки використовує пропускову здатність системи, то скільки часу займе поставка 10 тис. т. нафти?

2. Який максимальний потік може забезпечити лінія 2-3?

Номер варіанту (0-30):



Час поставки 10 тис. т. нафти:

Потік по лінії 2-3 (тис. т/год):

Вірно! Обидві відповіді вірні.

Завдання 5: Система автодоріг

Система автодоріг, що проєктує мережу, може забезпечити пропускову спроможність, показану на малюнку (система автодоріг на малюнку).

Питання:

1. Чому дорівнює максимальний потік автомобілів (кількість автомобілів на годину) для системи автодоріг, представленої на малюнку?

2. Який максимальний потік може забезпечити дорога 3-4?

Номер варіанту (0-30):



Максимальний потік (тис. автомобілів на годину):

Потік по дорозі 3-4 (тис. автомобілів на годину):

Вірно! Обидві відповіді вірні.

Для кожного з 30 варіантів генерується індивідуальна мережа, а відповіді перевіряються одразу, що допомагає студентам правильно оцінити отримані навички.

Вибір алгоритму максимального потоку безпосередньо впливає на швидкість обчислень, використання пам'яті та точність отриманих результатів у складних мережах. Тому, на лекціях пояснюється, що алгоритм Форда-Фалкерсона є одним із класичних методів знаходження максимального потоку, але існує кілька інших алгоритмів, які можуть бути ефективнішими в певних ситуаціях. Ось коротке порівняння.

Таблиця 4

Порівняння алгоритмів максимального потоку

	Форд-Фалкерсон	Edmonds-Karp	Дініц	Push-Relabel
Підхід:	Заснований на пошуку збільшувальних шляхів у графі.	Використовує пошук у ширину (BFS) для вибору шляхів.	Використовує рівневий граф і пошук в глибину (DFS).	Працює без використання збільшувальних шляхів, використовує операції підйому і перенесення
Часова складність:	Залежить від вибору шляху (може бути експоненційною в гіршому випадку).	$O(V E^2)$.	$O(V^2 E)$.	$O(V^3)$.
Переваги	Простота реалізації, застосовність до будь-яких графів.	Детермінована поведінка та стабільні часові характеристики.	Ефективніший на щільних графах, має кращу оптимізацію.	Підходить для паралельної реалізації та великих графів.

Для конкретної задачі, вибір алгоритму залежить від розміру графа, структури зв'язків і обмежень реального середовища. Якщо потрібно працювати з динамічними потоками, слід враховувати модифікації алгоритмів для змінних пропускових спроможностей. Якщо граф змінюється під час виконання алгоритму (наприклад, у транспортних або комунікаційних мережах), можливо, доведеться застосовувати інші методи, такі як модифікований Push-Relabel.

Для реалізації використовується **function fordFulkerson(taskNumber, n)**

<pre>function fordFulkerson(taskNumber, n) { const edgesTemplate = edgesTemplates[taskNumber]; const numVertices = taskNumber === 1 ? 7 : (taskNumber === 2 taskNumber === 3 ? 8 : (taskNumber === 4 ? 7 : 10)); // Вершини + 1 для індексації с 1 const source = 1; const sink = taskNumber === 1 ? 6 : (taskNumber === 2 taskNumber === 3 ? 7 : (taskNumber === 4 ? 4 : 9)); const graph = Array(numVertices).fill().map(() => Array(numVertices).fill(0)); edgesTemplate.forEach(edge => { let capacity = edge.capacity; if (capacity.includes('n')) { capacity = eval(capacity.replace('n', n)); } graph[edge.from][edge.to] = capacity; if (edge.reverse) { let reverseCapacity = edge.reverse; if (reverseCapacity.includes('n')) { reverseCapacity = eval(reverseCapacity.replace('n', n)); } graph[edge.to][edge.from] = reverseCapacity; } }); }</pre>	<pre>const residualGraph = graph.map(row => [...row]); const parent = new Array(numVertices).fill(-1); let maxFlow = 0; const flow = Array(numVertices).fill().map(() => Array(numVertices).fill(0)); while (bfs(residualGraph, source, sink, parent)) { let pathFlow = Infinity; for (let v = sink; v !== source; v = parent[v]) { const u = parent[v]; pathFlow = Math.min(pathFlow, residualGraph[u][v]); } for (let v = sink; v !== source; v = parent[v]) { const u = parent[v]; residualGraph[u][v] -= pathFlow; residualGraph[v][u] += pathFlow; flow[u][v] += pathFlow; flow[v][u] -= pathFlow; } maxFlow += pathFlow; } const result = taskNumber === 3 ? 10 / maxFlow : maxFlow; const specificFlow = taskNumber === 2 ? flow[4][7] : (taskNumber === 3 ? flow[2][3] : (taskNumber === 4 ? flow[3][4] : (taskNumber === 5 ? flow[3][5] : flow[5][6]))); return { result, specificFlow }; }</pre>
---	--

Рис.4. Лістинг алгоритму Форда-Фалкерсона

Висновок: застосування скриптів JavaScript у викладанні дискретної математики значно покращує освітній процес, роблячи його більш інтерактивним, адаптивним і ефективним. Використання JavaScript у платформі Moodle сприяє автоматизації перевірки завдань, підвищенню об'єктивності оцінювання та візуалізації математичних концепцій. Це не лише економить час викладачів і студентів, але й сприяє глибшому розумінню складних тем, таких як алгоритм Форда-Фалкерсона для пошуку максимального потоку в графі.

Завдяки технологічним можливостям JavaScript студенти можуть взаємодіяти з алгоритмами, аналізувати великі обсяги математичних даних та отримувати миттєвий JavaScript зв'язок щодо своїх розрахунків. У результаті впровадження JavaScript у навчальний процес створює умови для ефективного засвоєння знань, підвищуючи рівень цифрової грамотності студентів та сприяючи інтеграції інноваційних підходів у викладанні математичних дисциплін.

References

1. Hong, Z., Xi, D., Zhang, K., & Yan, T. (2024, June). Research on the Water Level Optimization of the Great Lakes Based on Ford-Fulkerson Algorithm. In *Proceedings of the 2024 4th International Conference on Artificial Intelligence, Big Data and Algorithms* (pp. 824-828).
2. Wahyuningsih, S., Octoviana, L. T., & Qohar, A. (2024, September). Performance of the maximum flow algorithm and its application to traffic density. In *AIP Conference Proceedings* (Vol. 3235, No. 1). AIP Publishing.
3. Астаф'єва, М. М., Глушак, О. М., & Литвин, О. С. (2024). Using STACK to support adaptive mathematics learning in LMS Moodle. In *3L-Person 2024: IX International Workshop on Professional Retraining and Life-Long Learning using ICT: Person-oriented Approach, co-located with the 19th International Conference on ICT in Education, Research, and Industrial Applications (ICTERI 2024)* (pp. 30-41). Lviv, Ukraine.

DOI 10.34132/mspc2025.01.14.15

Maksym Salyutin
Inessa Kulakovska

ASPECTS OF THE APPLICATION OF INFORMATION AND COMMUNICATION TECHNOLOGIES USING THE EXAMPLE OF JAVASCRIPT SCRIPTS ON THE MOODLE PLATFORM

Annotation: This research explores the application of information and communication technologies in education, specifically the use of JavaScript scripts on the Moodle platform for teaching discrete mathematics.

Digital technologies enhance interactivity, adaptability, and automation in assessment. JavaScript enables the creation of web applications for visualizing mathematical concepts, automating assignment evaluation, and developing educational platforms. Special attention is given to the Ford-Fulkerson algorithm for finding the maximum flow in graphs and its practical implementation in modeling transport and communication networks. JavaScript libraries such as Vis.js and Chart.js effectively visualize graph structures and truth tables. Moodle allows for the integration of custom plugins for algorithmic computations, improving the quality of education. Implementing JavaScript in education not only saves time for instructors and students but also facilitates deeper comprehension of complex mathematical concepts and enhances students' digital literacy. By leveraging JavaScript, students can interact with algorithms, analyze large sets of mathematical data, and receive instant feedback on their calculations, fostering a more engaging and efficient learning experience.

Key words: information and communication technologies, JavaScript, Moodle, discrete mathematics, graph visualization, Ford-Fulkerson algorithm, digital literacy

Відомості про автора / Information about the Author

Максим Салютін, аспірант, Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: salyutin1@gmail.com, ORCID: <https://orcid.org/0009-0003-0788-0012>

Maksym Salyutin, postgraduate student, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine, 54003, E-mail: salyutin1@gmail.com, ORCID: <https://orcid.org/0009-0003-0788-0012>

Інесса Кулаковська, кандидат фізико-математичних наук, доцент кафедри Інтелектуальних інформаційних систем, факультет комп'ютерних наук, Чорноморського національного університету імені Петра Могили, м. Миколаїв, Україна. E-mail: inessa.kulakovska@chmmu.edu.ua, orcid: <http://orcid.org/0000-0002-8432-1850>

Inessa Kulakovska, PhD of Physical and Mathematical Sciences, Associate Professor, Department of Intelligent Information Systems, Faculty of Computer Science, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine, 54003, E-mail: Inessa.Kulakovska@chmmu.edu.ua, ORCID: <http://orcid.org/0000-0002-8432-1850>