

сенсора з частотним виходом у 3,2 рази порівняно з датчиком на одному ОПТ та стабілізує вихідну частоту датчика.

### **Література:**

1. Prasad D., Nath V. An ultra-low power high-performance CMOS temperature sensor with an inaccuracy of 2 for aerospace applications. *Microsystem Technologies*. 2017. Vol. 25. No. 5. P. 1553–1563.
2. Lai S., Viola F. A., Cosseddu P., Bonfiglio A. Floating Gate, Organic Field-Effect Transistor-Based Sensors towards Biomedical Applications Fabricated with Large-Area Processes over Flexible Substrates. *Sensors*. 2018. Vol. 18. P. 688–699.
3. Li J. et.al. Long-range Raman Distributed Fiber Temperature Sensor with Early Warning Model for Fire Detection and Prevention. *IEEE Sensors Journal*, 2019. No. 8629078. P. 4427201.
4. Moreira Mota L.T., et.al. Development of a Surface Temperature Sensor to Enhance Energy Efficiency Actions in Buildings. *Sensors*. 2018. Vol. 18. No. 9, P. 3046–3062.

DOI <https://doi.org/10.36059/978-966-397-357-9-118>

## **РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАДАЧІ ОПТИМАЛЬНОГО ПЕРЕМНОЖЕННЯ ПОСЛІДОВНОСТІ МАТРИЦЬ ЗА ДОПОМОГОЮ ДИНАМІЧНОГО ПРОГРАМУВАННЯ**

**Новочинський Є. Г.**

здобувач вищої освіти другого (магістерського) рівня  
за спеціальністю 121 – Інженерія програмного забезпечення  
Міжнародний гуманітарний університет  
Науковий керівник: **Григор'єва Т. І.**  
кандидат технічних наук, доцент,  
завідувачка кафедри інформаційних технологій  
Міжнародний гуманітарний університет  
м. Одеса, Україна

Метод динамічного програмування є потужним інструментом для оптимізації обчислень в різноманітних задачах [1, 2]. Цей метод має свої переваги та недоліки, як і будь-який інший алгоритм або метод. В роботі

проведено аналіз умов, переваг та недоліків використання динамічного програмування.

Метод динамічного програмування може бути застосований до широкого спектру задач [3]. Динамічне програмування може бути дуже ефективним для розв'язання задач, які можна розбити на підзадачі. Але при цьому динамічне програмування може бути витратним з точки зору обчислювальної об'ємності, особливо для складних задач, які розбиваються на велику кількість підзадач. Зберігання значень для всіх підзадач у пам'яті може вимагати значних ресурсів, що може бути проблематичним для об'ємних задач.

Оскільки, динамічне програмування призначено для підвищення ефективності обчислень при вирішенні задач шляхом їх декомпозиції на відносно прості підзадачі, цей метод є важливим інструментом у різних галузях для розв'язання складних задач оптимізації.

Метою роботи є оптимізація послідовності множення матриць за допомогою динамічного програмування. В процесі застосування парадигми динамічного програмування для розв'язання складної задачі оптимального перемноження послідовності матриць, були визначені ключові етапи та вирішені основні завдання. Для реалізації рішення задачі обрано програмне забезпечення Microsoft Visual Studio, лінійка продуктів компанії Microsoft, що включає інтегроване середовище розробки програмного забезпечення, модулі тестування, пошуку помилок і ряд інших інструментальних засобів.

В роботі наведені приклади коду та формули, які використовуються для ефективного розв'язання задачі оптимального перемноження послідовності матриць. Розглянуто етапи розв'язання цієї задачі.

Першим кроком було необхідно знайти оптимальну допоміжну підструктуру. Тобто потрібно було визначити підзадачі, з яких буде конструюватися розв'язок. Для задачі оптимального перемноження послідовності матриць оптимальна допоміжна підструктура полягала у розбитті задачі на підзадачі, де кожне оптимальне рішення містило в собі оптимальні рішення менших допоміжних завдань. Таким чином, було визначено структуру оптимізаційної задачі. Було обрано, які параметри задачі є змінними, які є константами, які обмеження існують на змінні.

Другим етапом було знаходження рекурсивного рішення, тобто формування рекурсивної формули. Необхідно було виразити розв'язання задачі через розв'язання дрібніших підзадач. Рекурсивна формула повинна бути коректною і мати властивість оптимальної підструктури. У випадку оптимального перемноження матриць, допоміжне завдання включало оптимальну розстановку дужок в підпослідовності матриць.

На третьому етапі були створені таблиці для зберігання результатів підзадач, де кожна комірка містила оптимальне рішення для відповідної підзадачі. Під час заповнення таблиці використовується рекурсивна формула.

Заповнення таблиці відбувалося в порядку, починаючи з найменших підзадач і поступово переходячи до більших.

Останній етап полягав в отриманні розв'язку вихідної задачі, яке міститься в останній комірці таблиці. Це оптимальне рішення включає в себе оптимальні рішення всіх підзадач, що дозволяє знайти глобально оптимальний спосіб перемноження послідовності матриць.

Отже, застосування парадигми динамічного програмування вирішує складні задачі шляхом розбиття їх на менші оптимальні підзадачі (процедура декомпозиції), що дозволяє ефективно знаходити глобально оптимальні рішення.

Динамічне програмування є потужним інструментом, який знаходить широке застосування, але його використання вимагає уважності та аналізу, особливо з огляду на обчислювальну складність та обсяг використаної пам'яті. Метод динамічного програмування продовжить залишатися важливим інструментом у різних галузях для розв'язання складних задач оптимізації. Однак, дослідження нових алгоритмів та методів є важливим завданням для подальшого вдосконалення ефективності та універсальності методу.

**Висновки:**

1. Проведено аналіз умов, переваг та недоліків використання динамічного програмування.

2. Розроблено програмне забезпечення для задач оптимального перемноження послідовності матриць за допомогою динамічного програмування.

3. Використання метода динамічного програмування дозволяє отримувати не тільки зменшення кількості операцій, а також швидкий засіб формування запису послідовності множення матриць за рахунок рекурентних відносин.

4. Результати досліджень дозволяють рекомендувати використання динамічного програмування для оптимізації послідовності множення матриць при вирішенні задач в математиці, у фізиці, в техніці, в економіці, інших областях науки і знань.

### **Література:**

1. Динамічне програмування. URL: <https://ua5.org/osnprog/1907-dynamichne-programuvannya.html>

2. A. Lew, H. Mauch. Dynamic Programming: A Computational Tool. Springer-Verlag Berlin Heidelberg. 2007. 378 p.

3. Міца О.В., Оришич С.С., Дуло В.В. Аналіз ефективності динамічного підходу розв'язання задачі комівояжера. / XI Міжнародна наукова Інтернет-конференція «Національна безпека у фокусі викликів глобалізаційних процесів в економіці». 7–8 грудня 2021 року, Ukraine-Slovakia. С. 20–22.

DOI <https://doi.org/10.36059/978-966-397-357-9-119>

## **РОЗРОБКА E-LEARNING WEB-ДОДАТКУ З МЕТОДИКАМИ ГЕЙМІФІКАЦІЇ**

***Пір'ян Ю. В.***

*здобувач вищої освіти другого (магістерського) рівня*

*за спеціальністю 123 – Комп'ютерна інженерія*

*Міжнародний гуманітарний університет*

*Науковий керівник: Григор'єва Т. І.*

*кандидат технічних наук, доцент,*

*завідувачка кафедри інформаційних технологій*

*Міжнародний гуманітарний університет*

*м. Одеса, Україна*

Сучасний світ вимагає від нас постійно набувати нові знання, постійно підвищувати кваліфікацію. Дуже актуальною на сьогоднішній день є самоосвіта, тобто сучасна освітня платформа має надавати можливість не тільки здобувати нові знання, а й здійснювати самоконтроль і стимулювати саморозвиток, вона має бути ефективною і функціональною.

В Україні стратегічно важливим є впровадження інноваційних технологій, зокрема електронного навчання (e-learning), як ключового інструменту для досягнення цілей сталого розвитку в освіті [1].

В сучасному світі існує велика кількість ресурсів для навчання, але не всі з них надають користувачам відмінний досвід та можливість дивитись свої успіхи та самому контролювати виконання модулів та уроків. Деякі платформи пропонують лише уроки без можливості відстеження прогресу, інші взагалі крім курсів більше не мають функціоналу.

Метою роботи є розробка e-learning web-додатку з методиками гейміфікації з можливістю отримання точної статистики успіхів студента та заохочення до навчання за допомогою методик гейміфікації.