

АДАПТАЦІЯ МЕТОДУ ЛЕЙТНЕРА ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИВЧЕННЯ АЛГОРИТМІВ ТА СТРУКТУР ДАНИХ В ПРОГРАМНІЙ ІНЖЕНЕРІЇ

Вінницький національний технічний університет

Анотація

У роботі висвітлюється модифікація методу Лейтнера для підвищення ефективності вивчення алгоритмів та структур даних в програмній інженерії. Засвоюваність знань та навичок підвищується за рахунок використання методу інтервального повторення Лейтнера.

Ключові слова: інтерактивні технології, програмна інженерія, алгоритми та структури даних.

Abstract

The thesis describes Leitner's method modification for improved efficiency of studying the algorithms and data structures in software engineering. Learning the ideas and skills improves because of usage of the Leitner's interval repetition method.

Keywords: interactive technologies, software engineering, algorithms and data structures.

Вступ

У 1885 році німецький психолог Герман Еббінгауз провів дослідження пам'яті, у результаті якого було виявлено залежність відтворюваності нової інформації від часу. Через годину після першого безпомилкового повторення інформації відсоток відтворюваності склав 40%. Через десять годин – 35%. Через 24 години – 30%. Графік залежності формує криву Еббінгауза (рисунок 1) [1].

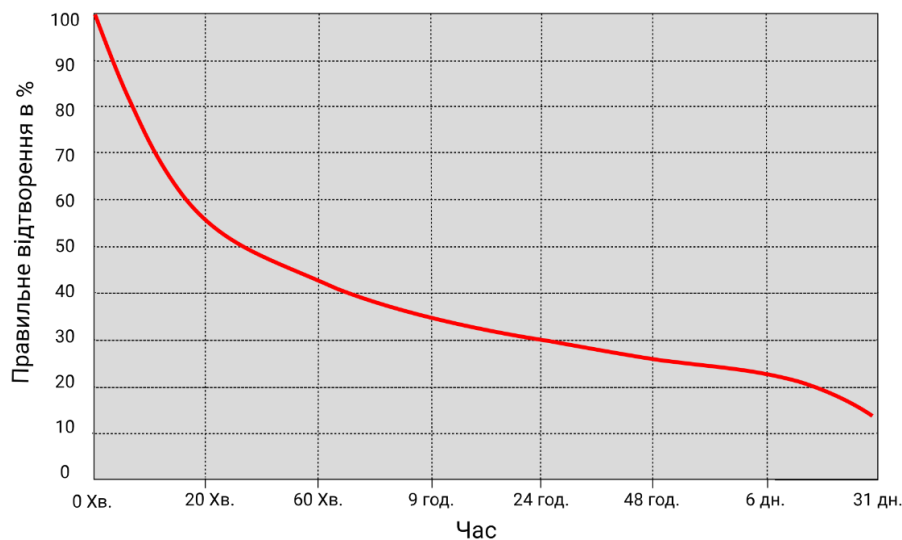


Рисунок 1 – Крива Еббінгауза

Дослідження Еббінгауза стало основою техніки інтервальних повторів, що полягає в повторенні засвоєного матеріалу через певні зростаючі інтервали часу. За таким принципом працює метод Лейтнера: фрагменти інформації записуються на картках і розбиваються на групи. Картки з першої групи демонструються учневі найчастіше, картки з другої групи – трохи рідше, і так далі. Якщо учень правильно пам'ятає інформацію з картки, вона переміщується у наступну групу. Якщо помиляється – картка переміщується у попередню або першу групу (рисунок 2) [2].

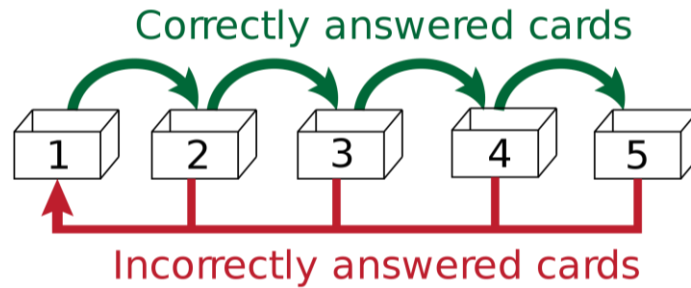


Рисунок 2 – Візуалізація методу Лейтнера

Метод Лейтнера широко використовується для вивчення іноземних мов. Але оригінальний метод Лейтнера не може бути використаний для формування навичок, адже концентрується на здатності відтворювати певну інформацію.

Розробка

У зв'язку з цим, є перспективною розробка модифікації методу Лейтнера, що може бути використана таким чином. Така модифікація описана нижче.

У цій модифікації картки – це алгоритми, структури даних, техніки розв'язання задач – наприклад, пошук у глибину [3], дерево відрізків [4], динамічне програмування [5] тощо.

Картки розбиваються на десять груп. У першій групі знаходяться найменш засвоєні картки. У десятій групі – найкраще засвоєні картки. У випадку, якщо учень проходить тестування успішно, картка переміщується до наступної по порядку групи. Якщо результат тестування незадовільний, картка переміщується до попередньої групи. Якщо учень не проходить тестування для картки протягом 7 днів, картка переміщується до попередньої групи.

Частота повторень для групи характеризується послідовністю Фібоначчі і наведена у таблиці 1.

Таблиця 1 – Частота повторень для групи

Номер групи	Частота повторень (дні)
1	1
2	2
3	3
4	5
5	8
6	13
7	21
8	34
9	55
10	89

Наприклад, картки з першої групи пропонуються для тестування щодня. Картки з десятої групи – раз на три місяці.

Тестування полягає у розв'язуванні певного набору задач, пов'язаних з темою картки. Підбір задач є ключовим фактором успішності застосування методу.

Для кожної задачі визначається числова характеристика складності. Складність задачі залежить від:

1. кількості людей, що успішно розв'язали задачу;
2. кількості людей, що зробили щонайменше одну спробу розв'язку;

3. середньої кількості спроб, необхідних для розв'язку задачі.

Числова характеристика складності – це ціле невід'ємне число. Значення 0 характеризує максимально просту задачу – задачу, яку з першого разу розв'язують усі, хто намагається. Максимальне значення складності не обмежується.

Запропоновано формулу:

$$X = \left(\frac{A * R^2}{S} - 1 \right) * 100$$

X – числова характеристика складності задачі;

S – кількість осіб, що розв'язали задачу;

A – кількість осіб, що намагались розв'язати задачу;

R – середня кількість спроб, за яку було досягнуто правильне рішення.

Для підвищення точності, для тренування не використовуються задачі, які намагались розв'язати менше 100 осіб.

Задачі розбиваються на десять груп за рівнем складності. Для цього вони сортуються за складністю і діляться на десять рівних груп. При тестуванні використовуються задачі з відповідним рівнем складності. Оскільки задача може бути пов'язана з декількома картками, для тестування відбираються задачі, які користувач може розв'язати. Наприклад, у користувача є картка «дерево (теорія графів)», що відноситься до 5 групи, і картка «алгоритм Крускала» [6], що відноситься до 1 групи. Таким чином, для тестування не буде використано задачу 5 групи складності, що пов'язана з обома вказаними картками, оскільки учень не в достатній мірі володіє темою «алгоритм Крускала».

Перевірка правильності розв'язку задачі може бути виконана за допомогою різних онлайн-ресурсів, що надають архів задач з програмування, а також автоматизовану тестову систему, що перевіряє правильність розв'язку задачі, а також відповідність заданим вимогам до споживання програмою системних ресурсів – таких як оперативна пам'ять та час виконання. Прикладами таких сервісів є Codeforces [7], E-Olymp, Timus Online Judge тощо.

Оскільки на початку тренувань учневі потрібно розв'язувати велику кількість задач, може виникати ситуація, у якій протягом одного дня учень не встигає пройти тестування для всіх запланованих карток. Для цього учневі потрібно визначити пріоритет кожної картки – таким чином, в першу чергу на тестування подаються картки з вищим пріоритетом.

Висновки

Подальший розвиток ідеї потребує експериментального дослідження. Існує необхідність розробки програмної системи, що забезпечить впровадження та дослідження ефективності розробленої модифікації методу Лейтнера.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Крива забування Еббінгауза – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Крива_забування_Еббінгауза
2. Leitner system – Wikipedia [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Leitner_system
3. Пошук у глибину – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Пошук_у_глибину
4. MAXimal :: algo :: Дерево отрезков [Електронний ресурс]. – Режим доступу: https://e-maxx.ru/algo/segment_tree
5. Задача о рюкзаке – Викиконспекты [Електронний ресурс]. – Режим доступу: https://neerc.ifmo.ru/wiki/index.php?title=Задача_о_рюкзаке
6. Алгоритм Крускала – Вікіпедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Алгоритм_Крускала
7. Help – Codeforces [Електронний ресурс]. – Режим доступу: <https://codeforces.com/help#q1>

Хом'юк Ірина Володимирівна – д. пед. н., професор, професор кафедри вищої математики, Вінницький національний технічний університет, м. Вінниця, e-mail: vikiravvh@gmail.com;

Кавка Олексій Олександрович – аспірант кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: oleksii.kavka@gmail.com.

Khomyuk Irina V. – Doctor of Science (Ped.), Professor of Higher Mathematics Department, Vinnytsia National Technical University, Vinnytsia, e-mail: vikiravvh@gmail.com;

Kavka Oleksii O. – Graduate Student of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: oleksii.kavka@gmail.com.