

Список використаних джерел

1. JavaScript Підручник. Основи вебпрограмування. URL: <https://w3schoolsua.github.io/js/index.html#gsc.tab=0> (дата звернення: 11.05.2024).
2. Meta Platforms. Посібник: знайомство з React. 2024. URL: <https://uk.legacy.reactjs.org/tutorial/tutorial.html> (дата звернення: 11.05.2024).
3. Використання Vite для швидкого розгортання веб-додатків на основі Vue та React. *IT Рейтинг України*. 20.04.2024. URL: <https://cutt.ly/AeeT3IDb> (дата звернення: 11.05.2024).
1. Що таке npm (Node Package Manager). Як встановити та розмістити пакети / А. Денисенко (автор, розробник і перекладач). *mc.today*. 18.04.2023. URL: <https://highload.today/uk/npm/> (дата звернення: 11.05.2024).

УДК 004.422.63

Козачок А. О., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки, Ветров О. С., старший викладач кафедри інформаційних технологій

ВИКОРИСТАННЯ ХЕШ-ТАБЛИЦЬ У СТРУКТУРАХ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Хеш-таблиці відіграють важливу роль у сучасних структурах даних, забезпечуючи швидкий і ефективний доступ до інформації. Це потужний інструмент, який використовується в багатьох областях інформатики, включно з базами даних, комп'ютерними мережами, компіляторами, а також системами управління контентом і іншими програмами, де потрібен швидкий доступ до даних. Завдяки своїй здатності виконувати операції пошуку, вставки та видалення за постійний час хеш-таблиці часто є ключовим складником сучасних алгоритмів та систем [1].

Основа хеш-таблиці – це хеш-функція, яка перетворює ключ у індекс, за яким зберігається відповідне значення. Хеш-функція має бути достатньо ефективною, щоб уникнути колізій, а також швидкою, щоб не уповільнювати операції. Хеш-таблиця зазвичай представляється у вигляді масиву чи списку, де індекси визначаються хеш-функцією.

Однією з основних проблем, пов'язаних із хеш-таблицями, є колізії – випадки, коли два різні ключі мають однаковий хеш-індекс. Для вирішення цієї проблеми використовуються різні методи. Метод ланцюжків передбачає, що в кожному індексі може зберігатися список (ланцюжок) елементів, які мають однаковий індекс. Інший підхід – відкрита адресація, де в разі колізії хеш-таблиця шукає інше місце для зберігання, застосовуючи різні стратегії, як-от лінійне або квадратичне пробування. Подвійне хешування використовує другу хеш-функцію, щоб зменшити ймовірність колізій [2].

Переваги хеш-таблиць у швидкості та ефективності. Операції пошуку, вставки та видалення відбуваються за дуже короткий час, оскільки не вимагають лінійного пошуку. До того ж хеш-таблиці є гнучкими і можуть використовуватися для зберігання різноманітних типів даних, від простих чисел до складних об'єктів [3].

Однак хеш-таблиці також мають свої недоліки. Колізії можуть уповільнити операції і потребують додаткових механізмів вирішення. Хеш-таблиці можуть споживати багато пам'яті, особливо коли прагнуть мінімізувати колізії. Також у хеш-таблицях неможливо зберігати порядок елементів, що може бути критичним у деяких застосуваннях.

Хеш-таблиці широко використовуються в різних галузях. У базах даних вони слугують для індексації та швидкого пошуку. У комп'ютерних мережах вони використовуються для кешування та маршрутизації. У мовах програмування вони є основою для реалізації словників, карт і асоціативних масивів.

Отже, хеш-таблиці є важливим інструментом у структурі даних. Їх використання дає змогу забезпечити високу ефективність, особливо під час роботи з великими обсягами інформації, де швидкість доступу є критичною. Водночас необхідно враховувати можливі недоліки і застосовувати відповідні стратегії для вирішення проблем, пов'язаних із колізіями та використанням пам'яті.

Список використаних джерел

1. Михалько В. Г. Адаптивна індексна структура бази даних для точкового пошуку за допомогою підходів машинного навчання: магістерська дисертація. *Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»*. 2018. URL: <https://ela.kpi.ua/server/api/core/bitstreams/af870de7-5956-4265-9b4b-0b91d588e666/content>
2. Hash Table Data Structure. *geeksforgeeks.org*. URL: <https://www.geeksforgeeks.org/hash-table-data-structure/> (дата звернення: 20.05.2024).
3. Розподілені хеш-таблиці. *Medium*. Oct. 30, 2020. URL: <http://surl.li/suowl> (дата звернення: 20.05.24).

УДК 004.06

Яценко В. В., здобувач 2 курсу спеціальності 122 Комп'ютерні науки, Вєтров О. С., старший викладач кафедри інформаційних технологій

ОГЛЯД АЛГОРИТМІВ ЧИТАННЯ ТА ЗАПИСУ В РЕЛЯЦІЙНИХ БАЗАХ ДАНИХ ТА СПОСОБИ ЇХ ОПТИМІЗАЦІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

Методи аналізу і обробки певних даних, упорядкованих і неупорядкованих, для добування знань називається наукою про дані [1]. Упорядкована інформація, об'єднана між собою за певними критеріями, називається базою даних [2]. Правила представлення, організації даних і їх зв'язків називаються моделлю даних [3]. Існують реляційна, об'єктно-орієнтована, графова, мережева, ієрархічна моделі даних. Найпопулярніша модель – реляційна – представляється у вигляді двовимірних таблиць. Для роботи з реляційними базами даних використовується SQL [4]. SQL – це мова структурованих запитів, що дає змогу користувачу взаємодіяти з даними у реляційних базах даних.

Ресурси knowledgehut, gigaspaces досліджують метод аналізу даних у реальному часі [5–6]. Згідно з їх дослідженням, сучасні компанії все частіше викорис-