

УДК 004.021

*Гуцуляк Д. В., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних
технологій*

АНАЛІЗ ЕФЕКТИВНОСТІ АЛГОРИТМІВ ПОШУКУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритми та структури даних у сучасному світі відіграють ключову роль у розвитку програмного забезпечення. Багато науковців приділяють увагу історії створення самих алгоритмів пошуку. Якби люди зберігали сотні записів, щоб мати легкий доступ до будь-якого з них пізніше, то для ефективного пошуку цих записів їх необхідно було б зберігати та індексувати. Ця проблема була успадкована, коли перші дослідники почали переміщувати записи на комп'ютери. Галузь пошукових алгоритмів добре вивчена, має довгу історію, яка почалася в математиці і швидко розширилася після винаходу комп'ютерів. Це почалося скромно з простими алгоритмами, як-от лінійний і двійковий пошук, але вони швидко зростали і незабаром їх стало більше [1].

Насамперед розглянемо основні алгоритми пошуку, як-от лінійний пошук, лінійний пошук з бар'єром, бінарний пошук. Кожен із цих алгоритмів має свої переваги та недоліки залежно від контексту застосування та обсягу даних. Ідея лінійного пошуку – проглядати по чергові елементи масиву, доки не буде знайдено шуканий елемент або не буде досягнуто кінця масиву. Псевдокод (рис. 1):

<p style="text-align: center;">$i=0$, поки ($i < N$ і $a[i] \neq \text{Key}$), $\{ i=i+1 \}$, якщо $i=N$, то елемент відсутній, інакше елемент має номер i</p>

Рисунок 1. Псевдокод лінійного пошуку

Лінійний пошук з бар'єром. Ідея – як і у попередньому випадку, елементи проглядаються по черзі, але для зменшення кількості порівнянь після останнього елемента додається елемент із ключем, що дорівнює шуканому значенню. Псевдокод наведено на рис. 2.

Бінарний пошук. Ідея – пошук, що реалізується в упорядкованому масиві. Шукане значення треба порівняти з ключем середнього елемента, внаслідок

цього порівняння визначити, у якій половині масиву знаходиться шуканий ключ, і знову застосувати ту саму процедуру до половини масиву. Процес припиняється, коли буде знайдено елемент або коли «довжина» таблиці стане менше 1 [2]. Псевдокод (рис. 3).

```
a[N] = Key; i = 0,  
поки (a[i] != Key),  
  { i = i+1 },  
якщо i = N,  
то елемент відсутній,  
інакше елемент має номер i
```

Рисунок 2. Псевдокод лінійного пошуку з бар'єром

```
L = 0; R = N-1,  
поки L < R,  
  {  
    m = (L+R) / 2,  
    якщо a[m] < Key,  
      то L = m+1,  
    інакше R = m,  
  }  
якщо a[R] = Key,  
то елемент має номер R,  
інакше елемент відсутній
```

Рисунок 3. Псевдокод бінарного пошуку

Із цього огляду можемо зробити висновок, що лінійний пошук простий, але неефективний для великих обсягів даних; лінійний пошук із бар'єром зменшує кількість порівнянь, а бінарний пошук ефективний лише для відсортованих даних, проте швидко знаходить шукане значення у відсортованому масиві.

Одним із поширених алгоритмів пошуку є пошук на основі хеш-таблиць. Хеш-таблиці, забезпечуючи швидкий доступ за ключем, використовуються для реалізації асоціативних масивів [3].

Порівняння ефективності різних алгоритмів пошуку у контексті різних структур даних розкриває ключові особливості кожного методу залежно від обсягу та властивостей даних. Лінійний пошук простий, але має обмежену ефективність для великих наборів даних. Лінійний пошук із бар'єром зменшує кількість порівнянь, проте не є універсальним рішенням для обробки великих обсягів даних.

Водночас бінарний пошук ефективний у відсортованих наборах, проте його ефективність залежить від ступеня відсортованості даних. Під час порівняння з хеш-таблицями, які забезпечують швидкий доступ за ключем, виявлено, що вони ефективні для асоціативних масивів та операцій з великими обсягами даних. Такий аналіз підкреслює важливість вибору конкретного алгоритму у поєднанні зі структурою даних, зокрема враховуючи обсяг і характеристики даних для оптимального використання в конкретних умовах [4].

Отже, вибір алгоритму пошуку має критичне значення для оптимального використання ресурсів та швидкості обробки інформації. Аналіз лінійного пошуку, лінійного пошуку з бар'єром та бінарного пошуку в контексті різних структур даних підкреслює їх переваги та обмеження. Для великих обсягів даних бінарний пошук впритул до відсортованих масивів є ефективним, тоді як хеш-таблиці виявляються корисними для швидкого доступу за ключем.

Список використаних джерел

1. Tudor Octavian Pocola. The evolution of search algorithms over time. Delft University of Technology. URL: https://filelist.tudelft.nl/Websections/Honours%20Exhibition/Scientific%20Writing/The_evolution_of_search_algorithms_over_time.pdf
2. Пошукові алгоритми. Алгоритми і структури даних: вебсайт. URL: http://elcat.pnpu.edu.ua/docs/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B8%20%D1%96%20%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B8%20%D0%B4%D0%B0%D0%BD%D0%B8%D1%85/lab2_search.html
3. Поняття структур даних, масив, список, словник, стек, черга, хеш-таблиця. Структури даних: вебсайт. URL: <https://ua5.org/struktury-danyh/1621-ponyattya-struktur-danyh-masyv-spysok-slovnyk-stek-chergha-hesh-tablyczya.html>
4. Introduction to Algorithms. 3rd Edition – Н. Thomas, E. Charles, L. Ronald, C. Stein. *MIT PRESS*, 2009. 1292 p.

УДК 004.07

Диркач Х. М., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки, Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій

СОРТУВАННЯ ТА ЙОГО ВПЛИВ НА АРХІТЕКТУРУ ПАМ'ЯТІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритм сортування визначається як алгоритм, що розміщує елементи якоїсь однотипної послідовності даних (масив, список, файл) у певному по-