

Лабораторна робота №4. Послідовний пошук

Мета: розробити програмні реалізації алгоритмів послідовного й швидкого послідовного пошуку. Провести аналіз алгоритмів.

Теоретичні відомості

Метод грубої сили розробки алгоритмів являє собою прямий підхід до рішення задачі, зазвичай, заснований безпосередньо на формулюванні задачі і на визначенні тих концепцій, що використовуються при постановці задачі.

У задачах пошуку будемо припускати, що є набір із n записів і завдання полягає в знаходженні однієї з них. Як і у випадку сортування, ми вважаємо, що кожний запис включає спеціальне поле – ключове поле. Набір усіх записів іменується таблицею або файлом.

Алгоритм пошуку має так званий *аргумент* K , і задача полягає в знаходженні запису, для якого K служить ключем.

Послідовний пошук. Почати з початку й продовжувати, поки не буде знайдений шуканий ключ, потім зупинитися.

Алгоритм S (Послідовний пошук, Sequential search)

// Дана таблиця записів $R_1, R_2, \dots, R_N$ з ключами $K_1, K_2, \dots, K_N$ відповідно.

// Алгоритм шукає запис з ключем K .

// Вихідні дані: позиція у вихідній послідовності запису, для якого ключ співпадає з K

Крок S1. Присвоїти змінній $i=1$.

Крок S2. Якщо $K=K_i$ повернути i та закінчити роботу.

Крок S3. Збільшити i на 1.

Крок S4. Якщо $i \leq N$, перейти на крок S2. У іншому випадку повернути повідомлення про відсутність шуканого запису та закінчити роботу.

Алгоритм S можна покращити наступним чином.

Алгоритм Q (Швидкий послідовний пошук, Quick sequential search)

// Дана таблиця записів $R_1, R_2, \dots, R_N$ з ключами $K_1, K_2, \dots, K_N$ відповідно.

// Алгоритм шукає запис з ключем K .

// Вихідні дані: позиція у вихідній послідовності запису, для якого ключ співпадає з K

Крок Q1. Присвоїти змінній $i=1$ та $K_{N+1}=K$.

Крок Q2. Якщо $K=K_i$ перейти на крок Q4.

Крок Q3. Збільшити i на 1 і перейти на крок Q2.

Крок Q4. Якщо $i \leq N$, повернути i та закінчити роботу. У іншому випадку повернути повідомлення про відсутність шуканого запису та закінчити роботу ($i=N+1$).

Математичний аналіз нерекурсивних алгоритмів.

1. Виберіть параметр (або параметри), за яким буде оцінюватися розмір вхідних даних алгоритму.
2. Визначте основну операцію алгоритму. (Як правило, вона перебуває в найбільше глибоко вкладеному внутрішньому циклі алгоритму.)
3. Перевірте, чи залежить число виконуваних основних операцій тільки від розміру вхідних даних. Якщо воно залежить і від інших факторів, розгляньте при необхідності, як змінюється ефективність алгоритму для найгіршого, середнього й найкращого випадків.
4. **Запишіть суму, що виражає кількість виконуваних основних операцій алгоритму.**
5. Використовуючи стандартні формули й правила підсумовування, спростить отриману формулу для кількості основних операцій алгоритму. Якщо це неможливо, визначите хоча б їхній порядок росту.

Завдання до лабораторної роботи

Для виконання поставленої мети, необхідно вирішити наступні задачі:

1. Реалізувати алгоритм S.
2. Реалізувати алгоритм Q.
3. Реалізації алгоритмів виконують пошук першого входження аргументу пошуку в послідовності випадкових чисел заданого об'єму.
4. Провести математичний аналіз алгоритмів S і Q.
5. Провести емпіричний аналіз алгоритмів. Зрівняти ефективності алгоритмів на вхідних даних різної структури.
6. Зрівняти ефективність алгоритмів.
7. Реалізувати інтерфейс користувача. Функціональні можливості інтерфейсу:
 - Виводити запрошення на введення об'єму випадкової послідовності й аргументу пошуку.
 - Виводити позицію шуканого елемента в списку.
 - Передбачити можливість виводу елемента випадкової послідовності по його позиції для перевірки результатів пошуку.

Контрольні запитання

1. Що таке метод розробки (проекування) алгоритмів?

2. У чому полягає особливість методу грубої сили?
3. Сформулюйте задачу пошуку в загальному виді.
4. Назвіть задачі, для яких ефективні алгоритми рішення можуть бути розроблені методом грубої сили.