МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ   
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ   
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

НАВЧАЛЬНА ДИСЦИПЛІНА  
Алгоритми та методи обчислень

ЗВІТ

З ПРАКТИЧНОЇ РОБОТИ №4

Виконала:

студентка групи КН-23-1

Морозова Ю.О.

Кременчук 2024

**Тема.** Алгоритми пошуку та їх складність

**Мета:** опанувати основні алгоритми сортування та навчитись методам

аналізу їх асимптотичної складності.

**Хід роботи**

**Варіант 1**

**1. Оцінити асимптотичну складність алгоритму лінійного пошуку у 𝑂- нотації в найгіршому і в найкращому випадку. Як можна покращити алгоритм лінійного пошуку?**

Найркащий випадок – O(1)

Найгірший випадок – O(n)

Можна покращити алгоритм лінійного пошуку, використовуючи оптимізацію, яка зупиняє пошук, якщо шуканий елемент знайдено, що зменшує кількість зайвих ітерацій. Також використання впорядкованих даних дозволяє застосувати бінарний пошук, що зменшує кількість порівнянь.

**2. Оцінити асимптотичну складність алгоритму бінарного пошуку у 𝑂-**

**нотації в найгіршому і в найкращому випадку.**

Найкращий випадок – О(1)

Найгірший випадок – О(logn)

**3. Побудувати алгоритм тернарного пошуку і оцінити його асимптотичну складність алгоритму у 𝑂-нотації в найгіршому і в найкращому випадку. Який з алгоритмів є оптимальнішим: бінарний, чи тернарний? Обґрунтувати відповідь відповідними обчисленнями**

Найкращий випадок – O(log\_3 n)

Найгірший випадок – O(log\_3 n)

Бінарний пошук може бути оптимальнішим за практичною швидкодією, особливо для великих масивів даних.

def binary\_search(arr, x):

left = 0

right = len(arr) - 1

comparisons = 0

while left <= right:

mid = (left + right) // 2

comparisons += 1

if arr[mid] == x:

return comparisons

elif arr[mid] < x:

left = mid + 1

else:

right = mid - 1

return comparisons

def ternary\_search(arr, x):

left = 0

right = len(arr) - 1

comparisons = 0

while left <= right:

mid1 = left + (right - left) // 3

mid2 = right - (right - left) // 3

if arr[mid1] == x:

return comparisons + 1

elif arr[mid2] == x:

return comparisons + 1

elif arr[mid1] > x:

right = mid1 - 1

elif arr[mid2] < x:

left = mid2 + 1

else:

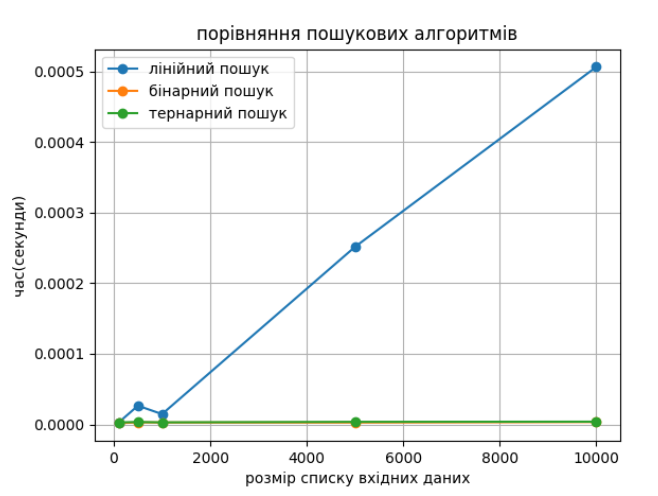
left = mid1 + 1

right = mid2 - 1

comparisons += 2

return comparisons

**4. Порівняти ефективність алгоритмів лінійного, бінарного та тернарного пошуку для різних розмірів вхідного списку. Для цього провести 30 експериментальне дослідження та побудувати графіки залежності часу виконання алгоритму від розміру вхідного списку**

****

**5. Порівняти алгоритми пошуку за їхньою здатністю працювати з**

**відсортованими та не відсортованими списками. Провести аналіз впливу**

**відсортованості списку на час виконання кожного алгоритму.**

Бінарний пошук найефективніший для відсортованих списків, але він не працює для невідсортованих. Лінійний пошук працює для обох типів списків, але менш ефективний, особливо для великих списків.

**Контрольні питання**

1. Що таке алгоритм пошуку і чому він важливий у контексті комп'ютерних наук?

Алгоритм пошуку - це процедура, що визначає метод пошуку конкретного елемента в масиві даних чи структурі даних. В контексті комп'ютерних наук алгоритми пошуку є важливою складовою, оскільки вони дозволяють ефективно знаходити необхідні дані в різних областях комп'ютерної науки

1. Які основні критерії оцінки ефективності алгоритмів пошуку?

Час виконання, складність, стабільність

1. Що таке лінійний пошук, і як він працює?

Лінійний пошук - це алгоритм, який перебирає елементи послідовно від початку до кінця списку, порівнюючи кожен елемент з шуканим значенням, доки не знайде його або не дійде до кінця списку.

1. Які умови повинні бути виконані для успішного застосування бінарного пошуку?

Відсортований масив, унікальні елементи, індексація елементів

1. Які переваги та недоліки використання бінарного пошуку порівняно з іншими алгоритмами пошуку?

Переваги: швидкість, ефективність, простота реалізації.

Недоліки: вимога унікальності, складність вставки та видалення елементів

1. Що таке тернарний пошук, і в чому його відмінність від бінарного пошуку?

Відмінність між тернарним і бінарним пошуками полягає в тому, що тернарний пошук розділяє масив на три частини, а бінарний - на дві. Тобто, в тернарному пошуку потрібно робити два порівняння з медіаною масиву, а в бінарному - одне.