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 ПРАКТИЧНИХ РОБІТ

Виконав:

студент групи КН-23-1

Ярковий Т.С.

Кременчук 2023

**Практична робота № 4**

**Тема. Алгоритми пошуку та їх складність**

**Мета:** опанувати основні алгоритми сортування та навчитись методам аналізу їх асимптотичної складності.

**Завдання**

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

Асимптотична складність в найкращому випадку O(1), в найгіршому O(n). Для покращення алгоритму необхідно попередньо впорядкувати масив.

2. Оцінити асимптотичну складність алгоритму бінарного пошуку у 𝑂- нотації в найгіршому і в найкращому випадку.

Асимптотична складність в найкращому випадку O(1), в найгіршому O(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.

int ternarySearch(int val, int a[], int left, int right) {

if (left >= right) return a[left] == val? left: -1;

int mid1 = (left\*2+ right)/3;

int mid2 (left + right\*2)/3;

if (val< a[midl]) {

return ternarySearch(val, a, left, midl 1);

else if (val == a[mid1]) {

return midl;

} else if (a < a[mid2]) {

return ternarySearch(val, a, mid11, mid2 - 1);

} else if (a == mid2) {

return mid2;

} else {

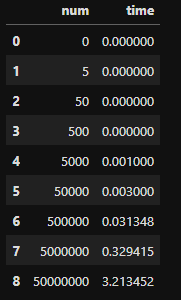
return ternarySearch(val, a, mid2+1, right);

}}

Бінарний пошук є більш ефективним. Ми ділимо на 3 частини, але кількість перевірок також збільшилась. За кількістю рекурсивних викликів виграш log3/log2=1,58. А кількість порівнянь збільшилась з 3 до 5 — програш = 1,67.

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.

Час лінійного пошуку



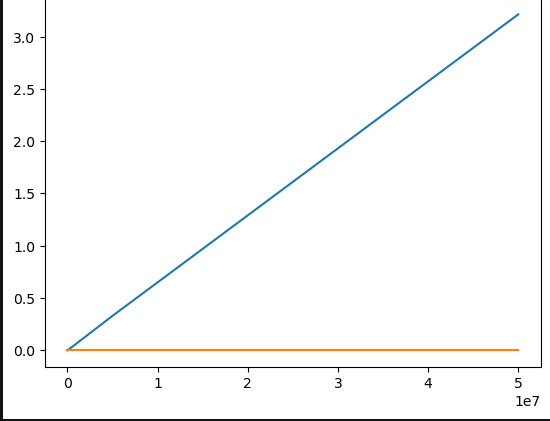
Час бінарного пошука



Час тернарного пошуку



Графіки



(Час бінарного і тернарного подібний тому вони зливаються в нижній лінії. Через велику швидкість пошуку графік бінарного і тернарного пошуку виглядає як пряма, насправді це log).

5. Порівняти алгоритми пошуку за їхньою здатністю працювати з відсортованими та не відсортованими списками. Провести аналіз впливу відсортованості списку на час виконання кожного алгоритму.

Лінійний пошук може працювати як в відсортованому масиві, так і в невідсортованому, відсортованість масиву пришвидшить роботу алгоритму на рівні константи. Бінарний і тернарний алгоритми можуть працювати лише в відсортованому масиві.

6. Розглянути сценарії використання кожного з алгоритмів пошуку у практичних задачах і обґрунтувати вибір кожного алгоритму в конкретному випадку.

Лінійний пошук потрібно використовувати в задачах пошуку в невідсортованому масиві, в невеликому масиві, масиві, що містить повтор елементів. Бінарний пошук варто використовувати в відсортованих масивах, особливо коли вони мають великий розмір. Тернарний пошук програє бінарному, тому краще використовувати бінарний.

**Контрольні питання**

1. Що таке алгоритм пошуку і чому він важливий у контексті комп'ютерних наук?   
Це алгоритм, що вирішує задачу знаходження певного елемента в контейнері. Алгоритми пошуку допомагаючи знаходити шукану інформацію швидко і ефективно в різних областях застосування.

2. Які основні критерії оцінки ефективності алгоритмів пошуку?

Основної характеристикою оцінки ефективності є часова складність.

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

Лінійний пошук — це пошук методом перебору. Пошук починають з першого елементу масиву. Якщо поточний елемент списку не дорівнює шуканому значенню, то здійснюється перехід до наступного елементу. Таким чином, у результаті кожної перевірки область пошуку зменшується на один елемент.

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

Масив має бути відсортованим.

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

Недоліки: масив має бути впорядкованим, більша складність операції insert(O(n)). Переваги: менший час пошуку find(O(logn)).

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

Тернарний пошук — це пошук у впорядкованому масиві, він йде аналогічно бінарному пошуку, різниця в тому, що масив ділять не на 2 частини, а на 3.