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 структури даних**»

Звіт

З Практичної роботи №4

Виконав

студент групи КН-24-1

Михайлик М. О.

Перевірив

доцент кафедри АІС

Сидоренко В. М.

Кременчук 2024

|  |  |
| --- | --- |
| Тема: | Алгоритми пошуку та їх складність. |
| Мета: | Опанувати основні алгоритми сортування та навчитись методам аналізу їх асимптотичної складності. |
|  |  |

**Хід роботи**

*Ознайомлення з короткими теоритичними відомостями*

*Виконати завдання.*

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

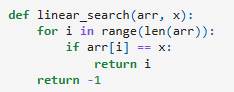


Рисунок 1 – «Лінійний пошук»

Якщо шуканий елемент знаходиться на початку масиву, він буде знайдений за першу ж ітерацію, отже в найкращому випадку складність – О(1).

Якщо елемент знаходиться в кінці масиву, або відсутній - треба переглянути всі елементи, отже складність О(n).

Для пришвидшення алгоритму можна створити хеш тфблицю для індексування елементів.

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

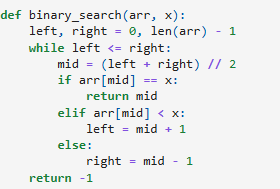


Рисунок 2 – «Бінарний пошук»

Якщо шуканий елемент знаходиться точно посередині на першій перевірці, то складність алгоритму О(1).

В найгіршому випадку алгоритмічна складність дорівнює O(log2 n) - в цьому випадку елемент або відсутній, або він знаходиться на кінцях масиву.

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. (Рис.3)

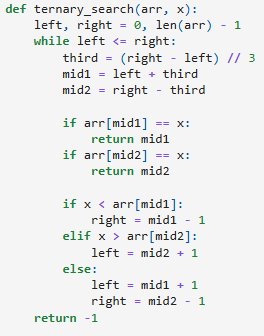


Рисунок 3 – «Тернарний пошук»

Якщо елемент знайдено на одному з перших порівнянь - наприклад, він збігається з mid1 або mid2, то складність алгоритму дорівнює О(1).

В найгіршому випадку алгоритмічна складність пошуку дорівнюватиме O(log3 n), оскільки масив ділиться на 3 частини та перевіряє одначасно два елементи, найгіршими випадками будуть або відсутність, або присутність на кінцях масиву.

Якщо привести до однієї основи логарифми, то ми зможемо побачити, що: 2×((log2 n)/1.58)≈1.26log2 n. З цього слідує, що асимптотична складність тернарного алгоритму пошуку вища за бінарний пошук. Порівнявши їх асимптотичну складність можемо побачити, що бінарний алгоритм пошуку більш оптимальний для проведення пошуку. На перший погляд може здатися, що тернарний алгоритм вигідніший, оскільки він має менше ітерацій розтину, але в цих ітераціях більше порівнянь, отже на кожну ітерацію втирачається більше часу.

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. (Рис. 4-7)

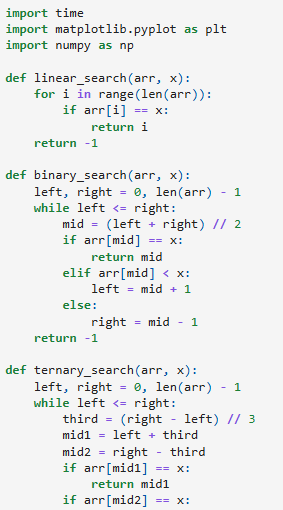


Рисунок 4 - «Код для графіку»

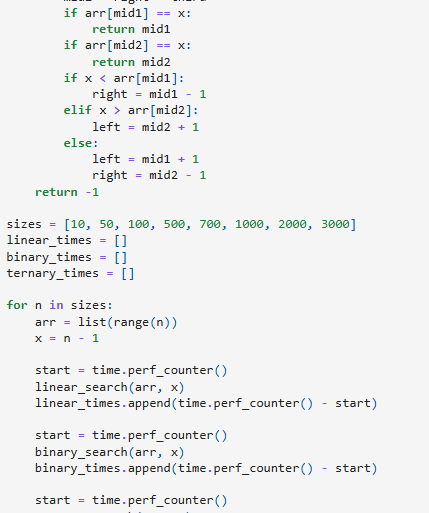


Рисунок 5 – «Ще код для графіку»

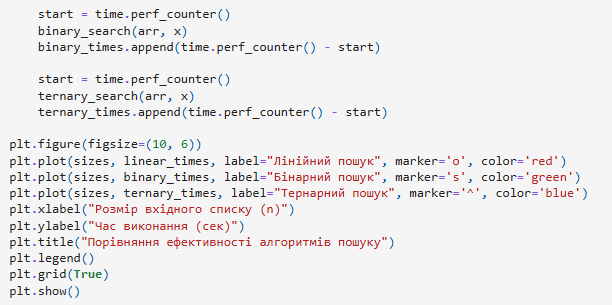


Рисунок 6 – «І ще код для графіку»

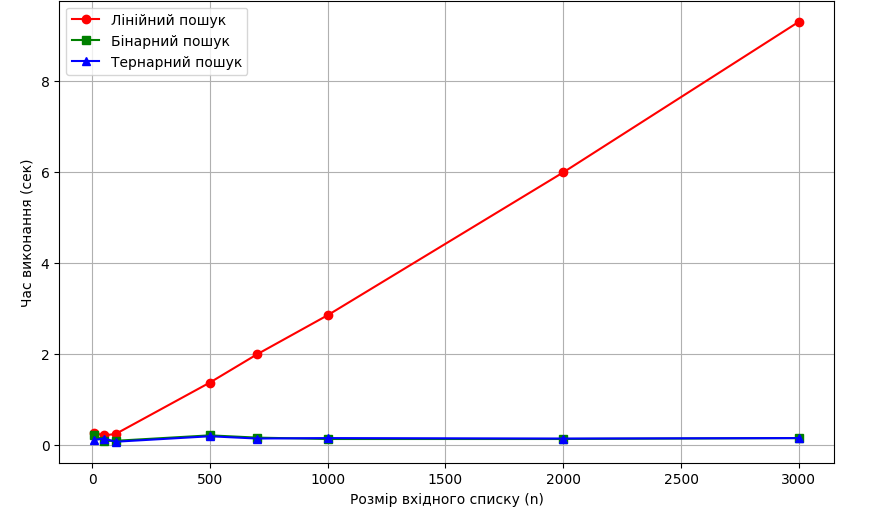


Рисунок 7 – «Порівняння еффективності алгоритмів пошуку»

Звідси слідує – те що тернарний та бінарний пошуки набагато швидші за лінійний, хоча й трохи відрізняются між собою.

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

Алгоритм лінійного пошуку може працювати як з відсортованими списками, так і з не відсортованими, але його асимптотична складність - O(n), тому він доволі повільний. В той час, як алгоритми бінарного та тернарного пошуку можуть працювати лише з відсортованими списками, з асимптотичною складністю - O(log n).

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

1. Лінійний пошук: наприклад, пошук потрібного товару в кошику покупця, пошук імені в списку користувачів.

2. Бінарний пошук: наприклад, автозаповнення в браузері, каталоги, словники, рейтинги.

3. Тетрарний пошук: наприклад, вибір найкращого ходу в обмеженому просторі, знайти найкращу точку на графіку.

*Додаткові питання*

1. Алгоритм пошуку - алгоритм, який дозволяє знайти заданий елемент у структурі даних (масиві, списку, дереві). У комп’ютерних науках пошук - одна з найбільш поширених задач, оскільки дані потрібно часто швидко знаходити у базах даних, пам’яті, файлах тощо.
2. Часова складність, просторова складність, наявність попередніх умов, стійкість до змін даних, середній/найгірший випадок виконання.
3. Лінійний пошук - алгоритм, який послідовно перевіряє кожен елемент списку до знаходження шуканого або до завершення перегляду.
4. Список повинен бути попередньо відсортованим.
5. Переваги: Швидкість (O(log n)) яка не втрачається з великою кількістю даних. Недоліки: потрібно попередньо сортувати список, більш складна реалізація, а також якщо мала кількість даних доречніше буде використати інший алгоритм пошуку.
6. Тернарний пошук — це алгоритм, який ділить діапазон на три частини замість двох, як у бінарному. Він також використовується у відсортованих списках.

*Висновки***:** В ході цієї роботи ми опанували основні алгоритми сортування та навчились методам аналізу їх асимптотичної складності.