Київський національний університет імені Тараса Шевченка

Кафедра програмних систем і технологій

Звіт про виконання лабораторної роботи з дисципліни

"Структури даних, аналіз і алгоритми комп'ютерної обробки інформації"

Виконав: студент 1 курсу ФІТ

Група ІПЗ-14

Мельничук Дмитро Олегович

Викладач: Бичков Олексій Сергійович

Київ-2022

1. Умова задачі

Написати програму мовою C# з можливістю вибору різних алгоритмів пошуку. Продемонструвати роботу (ефективність, час виконання) програм на різних структурах даних (масив, лінійний зв’язаний список), з різними умовами, що забезпечують зменшення часу виконання. Навести аналіз отриманих результатів.

Реалізувати алгоритми:

- пошуку перебором елемента масиву, що дорівнює заданому значенню.
- пошуку з бар'єром елемента масиву, що дорівнює заданому значенню.
- бінарного пошуку елемента масиву рівного заданому значенню.
- бінарного пошуку елемента масиву, рівного заданому значенню, в якій нове значення індексу m визначалося б не як середнє значення між L і R, а згідно з правилом золотого перерізу.

2. Аналіз задачі

Проаналізувавши умову задачі, ми вирішуємо використовувати знання та алгоритми, наведених у лекціях та завданнях на практику. Для двох типів даних буде ідентичний алгоритм пошуку, щоб побачити різницю у виконанні. Також для наглядності будемо використовувати таймер, щоб побачити за скільки часу буде виконуватись певний алгоритм.

3. Структура основних вхідних та вихідних даних

На вхід ми будемо подавати цілочисельні типи даних. Для масивів та лінійних зв’язних списків будемо вводити кількість елементів та діапазон значень цих елементів. А для алгоритмів пошуку будуть надходити готовий масив/лінійний зв’язний список та число, яке потрібно перевірити на наявність та було введене з клавіатури. На виході ми отримуємо інформацію на наявність елемента, а для массива ще і його індекс. Також буде виводитись час виконання алгоритму пошуку.

4. Алгоритм розв’язання задачі

Пошук перебором. Для реалізації цього алгоритму, нам потрібно реалізувати грубу силу, а саме - повний перебір масиву чи лінійного зв’язного списка. Переглядаємо кожен елемент, порівнюємо його із шуканим значенням і, якщо елемент дорівнює шуканому значенню, то ми зберігаємо індекс цього елемента. Алгоритм завершується тоді, коли ми знайшли шуканий елемент, або, коли весь масив/список пройдено і збігу не виявлено.

Пошук із бар’єром. Для цього алгоритму ми удосконалюємо попередній пошук перебором. А саме, ми спрощуємо логічну умову, зробивши бар’єм додатковий елемент, який буде мати шукане значення. Алгоритм буде швидшим, якщо існує гарантія , що співпадіння рано чи пізно відбудеться.

Бінарний пошук. Для цього алгоритму у нас буде додаткова інформація, а саме впорядковані дані. Сам алгоритм заключається в тому, щоб розділити елементи навпіл і, якщо середній елемент буде дорівнювати шуканому значенню, то пошук закінчується. Якщо ж ні, то порівнюємо це середнє значення із шуканим і, якщо середнє значення менше, то воно стає лівою границею, а якщо більше, то стає правою границею. Ці дії повторюються доти, поки не буде знайдено шуканий елемент, або коли цього значення не буде знайдено.

Бінарний пошук згідно з правилом золотого перерізу. Цей алгоритм відрізняється від стандартного бінарного пошуку тільки тим, що ми не ділимо елементи навпіл, а ділимо на золоте число, яке має значення $(1+\sqrt{5})/2$, що приблизно дорівнює ⅛. Алгоритм теж завершиться, коли число, отримане при ділені, буде дорівнювати шуканому, або коли цього значення не буде знайдено.

5. Текст програми

Текст готової програми буде викладений на GitHub.

Посилання: https://github.com/goldi4ek/me

6. Набір тестів

Для початку, проведемо тестування для масиву.

Згенеруємо один масив довжиною 1000 елементів в діапазоні значень від 0 до 1000. Виводиться 10 елементів з середини масива для зручності.

Почнемо пошуки:

Пошук перебором.

```
5216, 3005, 8851, 2781, 9591, 3504, 4142, 5862, 6083, 6931,
Find method choose:
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Search with gold cut - 4
1
Write elemetn which you need to find:
6931
Index of 6931 = 509
Time Spent for find: 00:00:00.0001582
Time Spent for find: 1582 miliseconds
```

Пошук із бар'єром.

```
5216, 3005, 8851, 2781, 9591, 3504, 4142, 5862, 6083, 6931,
Find method choose:
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Search with gold cut - 4
2

Write element which you need to find:
6931
Index of 6931 = 509
Time Spent for find: 00:00:00.0000018
Time Spent for find: 18 miliseconds
```

Бінарний пошук.

```
5216, 3005, 8851, 2781, 9591, 3504, 4142, 5862, 6083, 6931,
Find method choose:
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Search with gold cut - 4
3

Write elemetn which you need to find:
6931
Index of 6931 = 686
Time Spent for find: 00:00:00.0002201
Time Spent for find: 2201 miliseconds
```

Бінарний пошук згідно з правилом золотого перерізу.

```
5216, 3005, 8851, 2781, 9591, 3504, 4142, 5862, 6083, 6931,
Find method choose:
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Search with gold cut - 4
4

Write element which you need to find:
6931
Index of 6931 = 686
Time Spent for find: 00:00:00.0002723
Time Spent for find: 2723 miliseconds
```

Збільшимо кількість елементів у масиві до 100000

Пошук перебором.

```
5871, 6479, 5117, 219, 9093, 1651, 153, 4992, 9433, 7394,
Find method choose:
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Search with gold cut - 4
1
Write elemetn which you need to find:
153
Index of 153 = 10896
Time Spent for find: 00:00:00.0001874
Time Spent for find: 1874 miliseconds
```

Пошук із бар’єром.

```
5871, 6479, 5117, 219, 9093, 1651, 153, 4992, 9433, 7394,
Find method choose:
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Search with gold cut - 4
2

Write element which you need to find:
153
Index of 153 = 10896
Time Spent for find: 00:00:00.0000256
Time Spent for find: 256 miliseconds
```

Бінарний пошук.

```
5871, 6479, 5117, 219, 9093, 1651, 153, 4992, 9433, 7394,
Find method choose:
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Search with gold cut - 4
3

Write elemetn which you need to find:
153
Index of 153 = 1591
Time Spent for find: 00:00:00.0003133
Time Spent for find: 3133 miliseconds
```

Бінарний пошук згідно з правилом золотого перерізу.

```
5871, 6479, 5117, 219, 9093, 1651, 153, 4992, 9433, 7394,
Find method choose:
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Search with gold cut - 4
4

Write element which you need to find:
153
Index of 153 = 1592
Time Spent for find: 00:00:00.0002426
Time Spent for find: 2426 miliseconds
```

Тепер проведемо тестування для лінійних зв’язних списків. Тут також згенеруємо список з елементами та будемо шукати одне і те ж саме значення.

Кількість елементів – 1000, діапазон значень 1000

Пошук перебором.

```
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Binare search with gold cut - 4
1

Enumeration search

Enter search element
74
Found !
Time Spent for find: 00:00:00.0000081
Time Spent for find: 81 miliseconds
```

Пошук із бар’єром.

```
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Binare search with gold cut - 4
2

Search with barrier

Enter search element
74
Found !
Time Spent for find: 00:00:00.0000024
Time Spent for find: 24 miliseconds
```

Бінарний пошук.

```
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Binare search with gold cut - 4
3

Binare search

Enter search element
74
Found !
Time Spent for find: 00:00:00.0002194
Time Spent for find: 2194 miliseconds
```

Бінарний пошук згідно з правилом золотого перерізу.

```
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Binare search with gold cut - 4
4

Binare search with gold cut

Enter search element
74
Found !
Time Spent for find: 00:00:00.0002802
Time Spent for find: 2802 miliseconds
```

Збільшимо кількість елементів до 100000 та діапазон до 1000

Пошук перебором.

```
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Binare search with gold cut - 4
1

Enumeration search

Enter search element
7
Found !
Time Spent for find: 00:00:00.0000050
Time Spent for find: 50 miliseconds
```

Пошук із бар’єром.

```
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Binare search with gold cut - 4
2

Search with barrier

Enter search element
7
Found !
Time Spent for find: 00:00:00.0000006
Time Spent for find: 6 miliseconds
```

Бінарний пошук.

```
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Binare search with gold cut - 4
3

Binare search

Enter search element
7
Found !
Time Spent for find: 00:00:00.0003021
Time Spent for find: 3021 miliseconds
```

Бінарний пошук згідно з правилом золотого перерізу.

```
Enumeration search   - 1
Search with barrier  - 2
Binare search        - 3
Binare search with gold cut - 4
4

Binare search with gold cut

Enter search element
7
Found !
Time Spent for find: 00:00:00.0003271
Time Spent for find: 3271 miliseconds
```

7. Результати тестування програми та аналіз отриманих помилок.

Виконавши тестування для двох типів даних ми можемо зробити висновок, що алгоритми пошуку для лінійного зв'язного списку є швидшим за пошук у масиві. Але це залежить напряму від діапазону значень та довжини рядка або масиву.

Провівши тестування для масивів ми бачимо, що для малої кількості елементів найшвидшим є пошук з бар'єром, а на другому місці – пошук перебором. А коли збільшили кількості елементів то бачимо, що пошук з бар'єром є знов найшвидшим, а на другому – знову пошук перебором. Тому можна вважати, що для масиву найшвидшими є саме ці 2 методи, хоча це більше залежить від місця розташування шуканого елементу. Порівнюючи пошук з золотим перерізом та звичайний бінарний у випадку з маленькою кількістю елементів у масиві перший виявися більш ефективний, при більшій кількості елементів навпаки.

Тепер переходимо до лінійних зв'язних списків. Спостерігаємо, що для малої кількості елементів у нас найшвидшим знову є пошук з бар'єром. Збільшивши розмір рядка знову найефективнішим є пошук з бар'єром. Порівнюючи пошук з золотим перерізом та

звичайний бінарний для пошуку у списку ефективнішим виявився бінарний пошук, хоча з золотим перерізом від нього не сильно відставав.

При тестуванні програми виникала помилка при пошуку елементу методом золотого перерізу у структурі даних масив.