МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Структури даних, аналіз і алгоритми комп'ютерної обробки інформації»

Звіт до «Лабораторна робота № 4»

Виконала:	Кузнецова Дар’я Дмитрівна
Група	ІПЗ-11(2)
Форма навчання	денна
Спеціальність	121

2022

1. Умова завдання

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 завдання

Для роботи із цим завданням потрібно попередньо провести дослідження зв’язних списків. Це було прооблено на лекціях, але для щоб краще осягнути зміст цієї структури даних потрібно прочитати документацію вже наявного класу «Лінійні списки» у мові C#.

Найкраще для цього завдання використовувати тип даних цілих чисел (int), тому генерація масиву та лінійного списку відбувається саме з цілих чисел.

Основний текст програми буде написано у класі **Asd1,** а саме функція Main (точка входу програми), меню для вибору завдання, самі завдання, додаткові функції створення та виводу масиву, функції алгоритмів пошуку. Окремо для створення класу **LinkedList** є файл класу, у якому прописана структура цього класу та допоміжного класу **Node**. Також у класі **LinkedList** прописані функції цього класу, такі як: Додати елемент до списку на останню позицію, Видалити останній елемент зі списку, Повернути значення кількості елементів у списку, Відсортувати лінійний зв’язний список, Повернути створений з масиву список, Вивести список.

3. Структура основних вхідних та вихідних даних

Основними вхідними даними є цілі числа, для правильності вводу з консолі стоять перевірки. У функції основні вхідні дані – числа, пошук яких здійснюється. Вихідні дані це відсортовані масив та список, знайдені числа та час роботи алгоритму.

4. Алгоритм розв'язання задачі

Лінійний пошук:

- *У масиві:*
-

```
bool flag = false;
int index = 0;
stopWatch.Start();
while ((index < arr.Length) && !flag)
{
    if (arr[index] == key)
    {
        flag = true;
        Console.WriteLine("element " + key + " index is " + index);
    }
    index++;
}
if (!flag)
    Console.WriteLine("element wasn't found.");
```

- *У лінійному зв’язному списку:*

```
int index = 0;
bool flag = false;
Node iterator = linklist.head;

stopWatch.Start();
while ((iterator != null) && !flag)
{
    if (iterator.value == key)
    {
        flag = true;
        Console.WriteLine("element " + key + " index is " + --index);
    }
    index++;
    iterator = iterator.next;
}

if (!flag)
    Console.WriteLine("element wasn't found.");
```

Пошук з бар'єром:

- *У масиві:*

```
int index = 0;
bool flag = false;
int[] arrCopy = new int[arr.Length + 1];
Array.Copy(arr, arrCopy, arr.Length);
arrCopy[arrCopy.Length - 1] = key;

stopWatch.Start();
while (!flag)
{
    if (arrCopy[index] == key)
        flag = true;
    index++;
}
if (index == arrCopy.Length)
    Console.WriteLine("element wasn't found.");
else
    Console.WriteLine("element " + key + " index is " + --index);
```

- *У лінійному зв'язному списку:*

```
int index = -1;
bool flag = false;
Node iterator = linklist.head;
linklist.AddLast(key);

stopWatch.Start();
while (!flag)
{
    if (iterator.value == key)
        flag = true;
    index++;
    iterator = iterator.next;
}

if (index == linklist.ListLength(linklist))
    Console.WriteLine("element wasn't found.");
else
    Console.WriteLine("element " + key + " index is " + --index);
```

Бінарний пошук:

- *У масиві:*

```
int left = 0, right = (arr.Length - 1), mid = 0;
bool flag = false;
int[] arrSorted = arr;
Array.Sort(arrSorted);
Console.WriteLine("array sorted!");
OutputArray(arrSorted);

stopWatch.Start();
while (!flag && left < (right - 1))
{
    mid = (left + right) / 2;
    if (arrSorted[mid] == key)
        flag = true;
    else if (key > arrSorted[mid])
        left = mid + 1;
    else
        right = mid - 1;
}

if (flag)
    Console.WriteLine("element " + key + " index is " + mid);
else
    Console.WriteLine("element wasn't found.");
```

- *У лінійному зв'язному списку:*

```
Node iterator = listSorted.head;
int index = -1, left = 0, right = (listSorted.ListLength(listSorted) - 1);
int mid = (left + right) / 2;
bool flag = false;

stopWatch.Start();

while (iterator != null && !flag)
{
    if (index == mid)
    {
        if (key == iterator.value)
            flag = true;
        else if (key > iterator.value)
        {
            left = mid + 1;
            index++;
            iterator = iterator.next;
        }
        else
        {
            right = mid - 1;
            index = -1;
            iterator = listSorted.head;
        }
        mid = (left + right) / 2;
    }
    else
    {
        index++;
        iterator = iterator.next;
    }
}

if (flag)
    Console.WriteLine("element " + key + " index is " + index);
else
    Console.WriteLine("element wasn't found.");
```

Бінарний пошук з золотим перерізом:

- *У масиві:*

```
// GoldenRatio = (1 + Math.Sqrt(5)) / 2;
const double GoldenRatio = 1.6180339887498948482045868343 6; // golden number

int left = 0, right = (arr.Length - 1), mid = 0;
bool flag = false;
int[] arrSorted = arr;
Array.Sort(arrSorted);
Console.WriteLine("array sorted!");
OutputArray(arrSorted);

stopWatch.Start();
while (!flag && left < (right - 1))
{
    mid = (int)((left + right) / GoldenRatio);
    if (arrSorted[mid] == key)
        flag = true;
    else if (key > arrSorted[mid])
        left = mid + 1;
    else
        right = mid - 1;
}

if (flag)
    Console.WriteLine("element " + key + " index is " + mid);
else
    Console.WriteLine("element wasn't found.");
```

- *У лінійному зв'язному списку:*

```
const double GoldenRatio = 1.6180339887498948482045868343 6; // golden number
LinkedList listSorted = new LinkedList();
listSorted = LinkedList.Sort(arr);
listSorted.OutputList();

Node iterator = listSorted.head;
int index = -1, left = 0, right = (listSorted.ListLength(listSorted) - 1);
int mid = (int)((left + right) / GoldenRatio);
bool flag = false;

stopWatch.Start();
while (iterator != null && !flag)
{
    if (index == mid)
    {
        if (key == iterator.value)
            flag = true;
        else if (key > iterator.value)
        {
            left = mid + 1;
            index++;
            iterator = iterator.next;
        }
        else
        {
            right = mid - 1;
            index = -1;
            iterator = listSorted.head;
        }
        mid = (int)((left + right) / GoldenRatio);
    }
    else
    {
        index++;
        iterator = iterator.next;
    }
}
```

5. Текст програми

https://github.com/dxria/university_labs/tree/main/data%20structure/Asd1

6. Набір тестів

Для тестування програми ввожу спочатку менше значення розміру масиву, потім більше.

Це допоможе подивитись корректність роботи програми при великому обсязі інформації та меншому, а також порівняти час виконання алгоритму.

7. Результати тестування програми та аналіз отриманих помилок

```
=============== menu ==================
1. create an array.
2. linear search.
3. barrier search.
4. binary search.
5. golden ratio search.
0. stop the program.
=======================================
---------------------------------------
select command to execute.
1
enter array size: 14
array created!
array:    28    17     17     7      -7     -17    -44    -44    -11    27     -13    -8     -22    -1
---------------------------------------
select command to execute.
2
enter value of element to search: -1
element -1 index is 13
elapsed time (array): 140500 ns.

element -1 index is 13
elapsed time (list): 61900 ns.
---------------------------------------
select command to execute.
3
enter value of element to search: -1
element -1 index is 13
elapsed time (array): 119300 ns.

element -1 index is 13
elapsed time (list): 171300 ns.
---------------------------------------
select command to execute.
4
enter value of element to search: -1
array sorted!
array:    -44   -44    -22    -17    -13    -11    -8     -7     -1     7      17     17     27     28
element -1 index is 8
elapsed time (array): 436700 ns.

list:    -44   -44    -22    -17    -13    -11    -8     -7     -1     7      17     17     27     28
element -1 index is 8
elapsed time (list): 567100 ns.
---------------------------------------
select command to execute.
5
enter value of element to search: -1
array sorted!
array:    -44   -44    -22    -17    -13    -11    -8     -7     -1     7      17     17     27     28
element -1 index is 8
elapsed time (array): 579300 ns.

list:    -44   -44    -22    -17    -13    -11    -8     -7     -1     7      17     17     27     28
element -1 index is 8
elapsed time (list): 526500 ns.
---------------------------------------
select command to execute.
0
press any key to restart or 0 to stop
0
```

```
============== menu ==================
1. create an array.
2. linear search.
3. barrier search.
4. binary search.
5. golden ratio search.
0. stop the program.
=======================================
---------------------------------------
select command to execute.
1
enter array size: 100
array created!
array:    20    38      -22     -31     45      -1      -30     28      29      46      24      31      25      21      17
       -39     -48     29      13      -16     11      -22     37      8       48      14      33      -27     -19     -24
      -10     12      2       -22     10      45      -17     37      -4      -8      -31     48      -48     39      -47
     47      -13     31      10      19      31      -42     -5      -36     -39     -8      -49     19      -29     -19
    16      47      -3      2       6       -39     -21     -4      -40     21      28      -35     -17     -29     -35
   12      -25     47      5       -49     -24     -48     12      11      -5      -18     -33     -12     -33     -32
  44      13      16      -36     26      -29     1       0       -17     42
---------------------------------------
select command to execute.
2
enter value of element to search: 31
element 31 index is 11
elapsed time (array): 204200 ns.

element 31 index is 11
elapsed time (list): 71200 ns.
---------------------------------------
select command to execute.
3
enter value of element to search: 31
element 31 index is 11
elapsed time (array): 909100 ns.

element 31 index is 11
elapsed time (list): 227100 ns.
---------------------------------------
select command to execute.
4
enter value of element to search: 31
array sorted!
array:    -49   -49     -48     -48     -48     -47     -42     -40     -39     -39     -39     -36     -36     -35     -3
5       -33     -33     -32     -31     -31     -30     -29     -29     -29     -27     -25     -24     -24     -22     -2
2       -22     -21     -19     -19     -18     -17     -17     -17     -16     -13     -12     -10     -8      -8      -5
-5      -4      -4      -3      -1      0       1       2       2       5       6       8       10      10      11      11
12      12      12      13      13      14      16      16      17      19      19      20      21      21      24      25
26      28      28      29      29      31      31      31      33      37      37      38      39      42      44      45
45      46      47      47      47      48      48
element 31 index is 83
elapsed time (array): 465100 ns.

list:    -49    -49     -48     -48     -48     -47     -42     -40     -39     -39     -39     -36     -36     -35     -3
5       -33     -33     -32     -31     -31     -30     -29     -29     -29     -27     -25     -24     -24     -22     -2
2       -22     -21     -19     -19     -18     -17     -17     -17     -16     -13     -12     -10     -8      -8      -5
-5      -4      -4      -3      -1      0       1       2       2       5       6       8       10      10      11      11
12      12      12      13      13      14      16      16      17      19      19      20      21      21      24      25
26      28      28      29      29      31      31      31      33      37      37      38      39      42      44      45
45      46      47      47      47      48      48
element 31 index is 83
elapsed time (list): 430300 ns.
---------------------------------------
select command to execute.
5
enter value of element to search: 31
array sorted!
array:    -49   -49     -48     -48     -48     -47     -42     -40     -39     -39     -39     -36     -36     -35     -3
5       -33     -33     -32     -31     -31     -30     -29     -29     -29     -27     -25     -24     -24     -22     -2
2       -22     -21     -19     -19     -18     -17     -17     -17     -16     -13     -12     -10     -8      -8      -5
-5      -4      -4      -3      -1      0       1       2       2       5       6       8       10      10      11      11
12      12      12      13      13      14      16      16      17      19      19      20      21      21      24      25
26      28      28      29      29      31      31      31      33      37      37      38      39      42      44      45
45      46      47      47      47      48      48
element 31 index is 83
elapsed time (array): 398200 ns.

list:    -49    -49     -48     -48     -48     -47     -42     -40     -39     -39     -39     -36     -36     -35     -3
5       -33     -33     -32     -31     -31     -30     -29     -29     -29     -27     -25     -24     -24     -22     -2
2       -22     -21     -19     -19     -18     -17     -17     -17     -16     -13     -12     -10     -8      -8      -5
-5      -4      -4      -3      -1      0       1       2       2       5       6       8       10      10      11      11
12      12      12      13      13      14      16      16      17      19      19      20      21      21      24      25
26      28      28      29      29      31      31      31      33      37      37      38      39      42      44      45
45      46      47      47      47      48      48
element 31 index is 83
elapsed time (list): 332500 ns.
---------------------------------------
select command to execute.
0
press any key to restart or 0 to stop
0
```

Робимо висновок, що програма працює коректно.