

Лабораторна робота №1

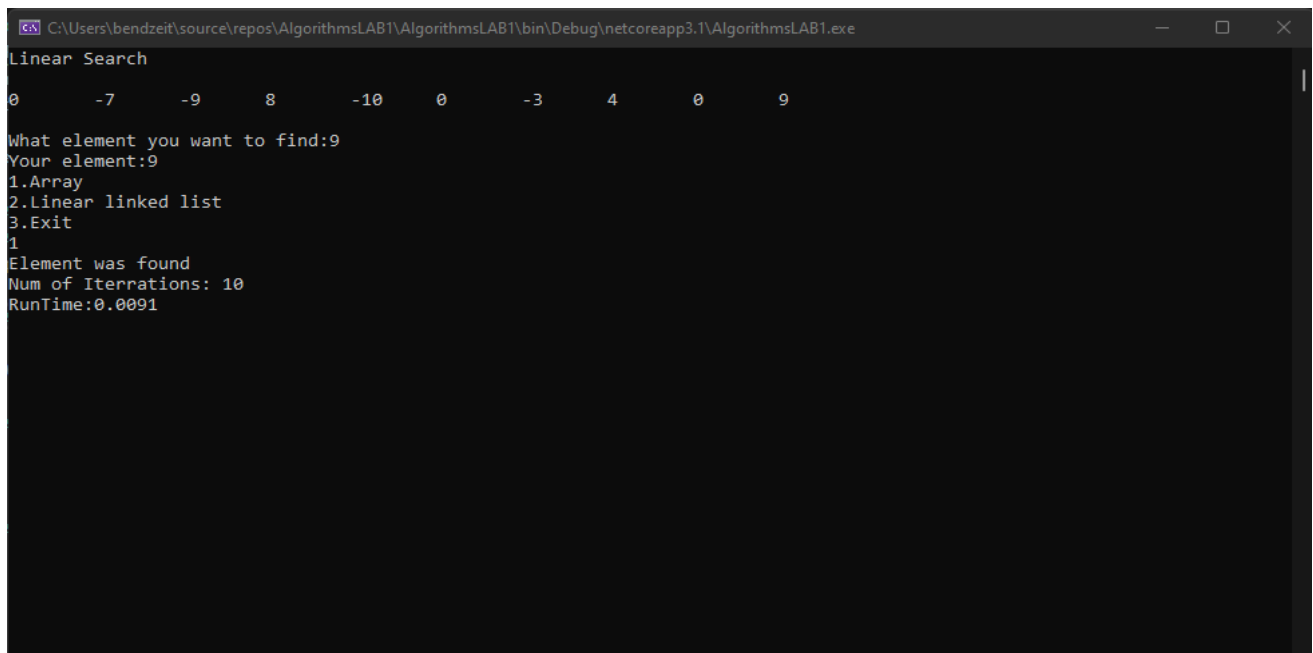
- пошуку перебором елемента масиву, що дорівнює заданому значенню

Продемонструвати роботу ефективність (час виконання) програм на різних структурах даних (масив, лінійний зв'язаний список)

Час виконання зміряється класом **StopWatch**

1. Масив

Маємо масив з 10 елементів, вибравши останій елемент як шуканий, маємо результат виконання нашої програми :



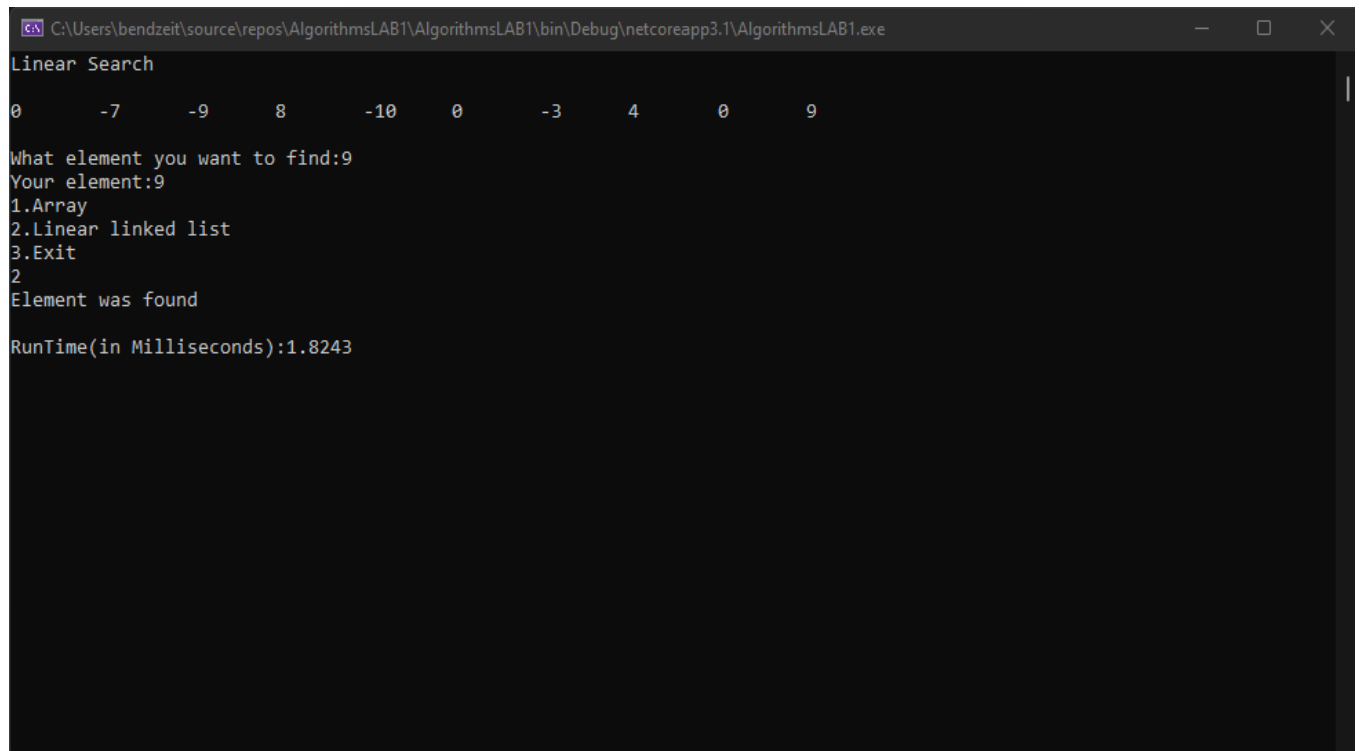
```
C:\Users\bendzeit\source\repos\AlgorithmsLAB1\AlgorithmsLAB1\bin\Debug\netcoreapp3.1\AlgorithmsLAB1.exe
Linear Search
0      -7      -9      8      -10     0      -3      4      0      9
What element you want to find:9
Your element:9
1.Array
2.Linear linked list
3.Exit
1
Element was found
Num of Iterations: 10
RunTime:0.0091
```

Час виконання: **0.0091**

Кількість ітерацій: 10

2. Лінійний зв'язний список

Маємо ті ж самі елементи :



```
C:\Users\bendzeit\source\repos\AlgorithmsLAB1\AlgorithmsLAB1\bin\Debug\netcoreapp3.1\AlgorithmsLAB1.exe
Linear Search
0      -7      -9      8      -10     0      -3      4      0      9

What element you want to find:9
Your element:9
1.Array
2.Linear linked list
3.Exit
2
Element was found
RunTime(in Milliseconds):1.8243
```

Час виконання: **1.8243**

Як бачимо лінійний список знайшов елемент пізніше ніж масив на 1.8152 мілісекунди швидше, так як для масиву ми використовуємо індекси, а для зв'язного списку проходимо за допомогою foreach що і збільшило час виконання.

- пошуку з бар'єром елемента масиву, що дорівнює заданому значенню.

Кількість елементів: 100

1. Масив

```
What element you want to find:203
Your element:203
After adding barrier:
261 35 906 203 748 966 236 -75 389 350 -615 -855 392 -84 324 -
498 53 -90 389 -932 663 985 -171 399 958 549 -942 -638 799 -547 -
174 166 614 -676 -611 -754 -509 102 -718 27 -412 -81 -841 259 -534 7
9 478 -550 -567 315 -918 965 874 331 904 275 -795 -494 245 -802 9
07 -126 126 -65 -476 218 -469 67 -474 77 -989 950 -373 -639 480 -
256 826 570 -67 -122 -929 607 929 -517 -630 -139 291 -718 -580 114 -
960 984 857 -916 120 713 793 223 -266 247 203
1.Array
2.Linear linked list
3.Exit
1
Element was found
Num of Iterations: 4
RunTime:0.0051
```

У масиві елемент було знайдено за **0.0051** мс.

2. Лінійний зв'язний список

```
What element you want to find:203
Your element:203
After adding barrier:
261 35 906 203 748 966 236 -75 389 350 -615 -855 392 -84 324 -
498 53 -90 389 -932 663 985 -171 399 958 549 -942 -638 799 -547 -
174 166 614 -676 -611 -754 -509 102 -718 27 -412 -81 -841 259 -534 7
9 478 -550 -567 315 -918 965 874 331 904 275 -795 -494 245 -802 9
07 -126 126 -65 -476 218 -469 67 -474 77 -989 950 -373 -639 480 -
256 826 570 -67 -122 -929 607 929 -517 -630 -139 291 -718 -580 114 -
960 984 857 -916 120 713 793 223 -266 247 203
1.Array
2.Linear linked list
3.Exit
2
Element was found
RunTime(in Milliseconds):0.9945
```

У Лінійний зв'язному список елемент було знайдено за **0.9945** мс.

Можна зробити висновок, що для ітераційного пошуку швидшим буде масив, через те, що доступ до елемента за допомогою індексу працює швидше ніж foreach

- бінарного пошуку елемента масиву рівного заданому значенню.

1. Масив

```
Binary Search

Sorted Array:
-989  -960  -942  -932  -929  -918  -916  -855  -841  -802  -795  -754  -718  -718  -676  -
539   -638  -630  -615  -611  -580  -567  -550  -547  -534  -517  -509  -498  -494  -476  -
474   -469  -412  -373  -266  -256  -174  -171  -139  -126  -122  -90   -84   -81   -75   -
57    -65   27   35   53   67   77   79   102  114  120  126  166  203  218  2
23    236  245  247  259  261  275  291  315  324  331  350  389  389  392  3
99    478  480  549  570  607  614  663  713  748  793  799  826  857  874  9
94    906  907  929  950  958  965  966  984  985

What element you want to find:27
Your element:27

1.Array
2.Linear linked list
3.Exit
1
Element was found

RunTime(in Milliseconds):0.0014
```

Час затрачений на пошук елемента **0.0014** мс.

Що в порівнянні з ітераційним пошуком в декілька разів швидше.

2. Лінійний зв'язний список

```

Binary Search

Sorted Array:
-989  -960  -942  -932  -929  -918  -916  -855  -841  -802  -795  -754  -718  -718  -676  -
539  -638  -630  -615  -611  -580  -567  -550  -547  -534  -517  -509  -498  -494  -476  -
474  -469  -412  -373  -266  -256  -174  -171  -139  -126  -122  -90  -84  -81  -75  -
67  -65  27  35  53  67  77  79  102  114  120  126  166  203  218  2
23  236  245  247  259  261  275  291  315  324  331  350  389  389  392  3
99  478  480  549  570  607  614  663  713  748  793  799  826  857  874  9
04  906  907  929  950  958  965  966  984  985

What element you want to find:27
Your element:27

1.Array
2.Linear linked list
3.Exit
2
Element was found

RunTime(in Milliseconds):0.4511

```

Для знаходження того ж самого елемента, лінійний зв'язний список затратив **0.4511** млс. що в порівнянні з масивом є досить поганим результатом.

Масив знайшов елемент у 322 рази швидше ніж зв'язний список.

- 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

1. Масив

```

Binary Search with Golden Ratio

Sorted Array:
-989  -960  -942  -932  -929  -918  -916  -855  -841  -802  -795  -754  -718  -718  -676  -
539  -638  -630  -615  -611  -580  -567  -550  -547  -534  -517  -509  -498  -494  -476  -
474  -469  -412  -373  -266  -256  -174  -171  -139  -126  -122  -90  -84  -81  -75  -
67  -65  27  35  53  67  77  79  102  114  120  126  166  203  218  2
23  236  245  247  259  261  275  291  315  324  331  350  389  389  392  3
99  478  480  549  570  607  614  663  713  748  793  799  826  857  874  9
04  906  907  929  950  958  965  966  984  985

What element you want to find:67
Your element:67

1.Array
2.Linear linked list
3.Exit
1
Element was found

RunTime(in Milliseconds):0.0014

```

Час виконання: **0.0014млс.** що є ідентичним до бінарного пошуку в масиві.

2. Лінійний зв'язний список

```
Binary Search with Golden Ratio

Sorted Array:
-989  -960  -942  -932  -929  -918  -916  -855  -841  -802  -795  -754  -718  -718  -676
539   -638  -630  -615  -611  -580  -567  -550  -547  -534  -517  -509  -498  -494  -476
474   -469  -412  -373  -266  -256  -174  -171  -139  -126  -122  -90   -84   -81   -75
57    -65   27    35    53    67    77    79    102   114   120   126   166   203   218
23    236   245   247   259   261   275   291   315   324   331   350   389   389   392
99    478   480   549   570   607   614   663   713   748   793   799   826   857   874
04    906   907   929   950   958   965   966   984   985

What element you want to find:67
Your element:67

1.Array
2.Linear linked list
3.Exit
2
Element was found

RunTime(in Milliseconds):0.0094
```

Час виконання: **0.0094млс** , що являється майже у 7 разів довше ніж у масиві

Висновки:

Випробувавши різні методи пошуку з різними структурами даних, маємо такий результат:

Ітераційний

Масив : 0.0091

Зв'язний список : 1.8

Ітераційний з бар'єром

Масив : 0.0051

Зв'язний список : 0.9945

Бінарний

Масив: 0.0014

Зв'язний список : 0.4511

Бінарний із золотим перерізом

Масив: 0.0014

Зв'язний список : 0.0094

Можна побачити, що масив є більш ефективним ніж зв'язний список у всіх алгоритмах.