Звіт до лабораторної роботи 1

Алгоритми та структури даних

ІПЗ-11/1

Студент: Зганяйко Володимир Віталійович

Викладач: Бичков Олексій Сергійович

1. Умова завдання лабораторної роботи.

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 середенє значення між L I R, а згідно з правилом золотого перерізу.

1. Аналіз задачі.

Для зручності роботи з алгоритмами, будуть використані масиви та списки, що складаються з цілих чисел. Для усіх види пошуків, які необхідно здійснити, будуть створені окремі методи для кожного пошуку та для кожної структури даних. Тобто, у нас є чотири види пошуку та дві структури даних, виходить, що таких методів буде вісім. Усі вони будуть описані в основному класі Program.

Крім того, буде створено окремий клас Node, у якому будуть створені його поля, та описані методи, такі як додавання нового елементу до списку та виведення його на екран.

Для зручної роботи програми буде створене меню, за яким користувач зможе обрати з якою структурою даних він хоче працювати та який вид пошуку до неї здійснити.

Для перевірки ефективності кожного з методів, різні пошуки буде здійснено на одному й тому самому масиві чи списку. Буде здійснене виведення тривалості алгоритму та кількості затрачених ітерацій циклу. Тривалість роботи алгоритму буде обрахована за допомогою класу Stopwatch та його методами: Start(), Stop(), ElapsedMilliseconds().

1. Структура основних вхідних та вихідних даних.

Основними вхідними даними, які будуть введенні з клавіатури: довжина масиву, довжина списку, шуканий елемент.

Вихідні дані, які будуть отримані у результаті виконання програми: Індекс шуканого елементу, кількість ітерацій циклу, тривалість роботи алгоритму пошуку.

1. Алгоритм розв’язання задачі.

**Для масиву:**

**Генерація та виведення масиву:**

Random random = new Random();

for (int i = 0; i < array.Length; i++)

{

array[i] = random.Next(0, array.Length);

}

Console.WriteLine("Ваш масив:");

for (int i = 0; i < array.Length; i++)

{

Console.Write(array[i] + " ");

}

Console.WriteLine();

return array;

**Лінійний пошук:**

Duration.Start();

for (int i = 0; i < array.Length && !flag; i++)

{

iters++;

if (array[i] == key)

{

index = i;

flag = true;

Duration.Stop();

}

}

**Пошук з бар’єром:**

Duration.Start();

while (array1[index] != key)

{

iters++;

index++;

}

Duration.Stop();

**Бінарний пошук:**

Duration.Start();

while (left <= right && !flag)

{

iters++;

middle = (left + right) / 2;

if (key == array[middle])

{

flag = true;

Duration.Stop();

}

else if (key < array[middle])

{

right = middle - 1;

}

else

{

left = middle + 1;

}

}

Duration.Stop();

**Бінарний пошук з золотим перерізом:**

Duration.Start();

while (left <= right && !flag)

{

iters++;

middle = (int)(right - (right - left) / goldratio);

if (key == array[middle])

{

flag = true;

Duration.Stop();

}

else if (key < array[middle])

{

right = middle - 1;

}

else

{

left = middle + 1;

}

}

Duration.Stop();

**Лінійний зв’язний список:**

**Генерація та виведення списку:**

Random random = new Random();

for (int i = 1; i < length; i++)

{

head.AddToEnd(random.Next(0, length));

}

Console.WriteLine("Ваш список: ");

head.Print();

Console.WriteLine("");

**Знаходження поточного елементу списку:**

for (int i = 0; i < index; i++)

{

element = element.next;

}

return element.data;

**Сортування списку:**

Node element;

while (head != null)

{

element = head.next;

while (element != null)

{

if (head.data > element.data)

{

int temp = head.data;

head.data = element.data;

element.data = temp;

}

element = element.next;

}

head = head.next;

}

**Лінійний пошук:**

Duration.Start();

for (int i = 0; head != null && !flag; i++)

{

iters++;

if (head.data == key)

{

flag = true;

index = i;

Duration.Stop();

}

head = head.next;

}

Duration.Stop();

**Пошук з бар’єром:**

Duration.Start();

while(head.data != key)

{

iters++;

head = head.next;

index++;

}

Duration.Stop();

**Бінарний пошук:**

Duration.Start();

while (left <= right && !flag)

{

iters++;

middle = (int)((right + left) / 2);

if (key == CurrentElement(head, middle))

{

flag = true;

Duration.Stop();

}

else if (key < CurrentElement(head, middle))

{

right = middle - 1;

}

else

{

left = middle + 1;

}

}

Duration.Stop();

**Бінарний пошук з золотим перерізом:**

Duration.Start();

while (left <= right && !flag)

{

iters++;

middle = (int)((right + left) / 2);

if (key == CurrentElement(head, middle))

{

flag = true;

Duration.Stop();

}

else if (key < CurrentElement(head, middle))

{

right = middle - 1;

}

else

{

left = middle + 1;

}

}

Duration.Stop();

1. Текст програми.

Текст програми ви можете знайти за посиланням.

https://github.com/Vovan4ok/ASD

1. Набір тестів.

Набір тестів підібраний таким чином, що буде здійснений пошук елемента, який стоїть під одним і тим самим індексом.

**Масив**

Масив на 10 елементів: 6 12 10 45 12 5 38 48 23 33 Будемо шукати елемент, що знаходиться по середині масиву, на п’ятій позиції. Для звичайного масиву це буде число 5, для відсортованого – 23.

Масив на 100 елементів: 22 479 188 193 358 388 308 56 172 349 106 413 97 361 119 285 357 138 266 93 179 330 127 278 263 279 200 422 311 155 240 277 96 189 89 74 397 256 414 330 470 33 212 214 143 457 21 229 397 222 497 392 494 61 351 192 351 410 430 79 188 386 244 16 108 329 316 407 271 44 388 100 224 436 123 146 118 413 134 305 257 470 24 88 158 348 283 35 338 397 223 487 326 334 141 83 288 195 105 314

Для масиву на 100 елементів, буде здійснено пошук першого елемента: для звичайного масива – число 22, для відсортованого – 16.

Масив на 1000 елементів: 3702 4477 4924 415 4309 3198 4418 3758 3850 4646 3411 1087 152 2727 4373 4399 1177 1456 753 2871 748 4342 3960 2623 4893 1079 2681 2630 4261 3104 4252 2631 2319 3188 980 3243 4112 1806 1740 2889 104 3394 2457 3175 1045 2814 835 1464 4119 3153 3409 4034 2155 2928 2701 4359 516 2300 522 3229 516 1787 4496 745 393 780 3768 1964 1747 1129 286 4370 4716 2864 2767 2354 1885 785 1578 2078 244 1217 3510 1107 4695 217 476 4390 487 1620 2727 1812 1283 3510 2373 3316 3897 1711 2781 265 4046 3871 4717 2990 2866 4038 4318 4291 161 346 2474 4730 722 3444 2985 4298 3276 3388 1049 176 303 4377 1476 126 3401 3474 3086 1205 491 4450 3456 173 3004 1313 3032 1372 1499 520 3241 657 898 1184 4229 140 4145 2996 1090 2839 525 3074 39 509 662 2604 4961 4669 2394 4590 4588 4392 951 3113 3799 711 1890 2301 1725 4408 412 1612 2799 2755 147 392 4278 4118 148 1336 980 404 2384 247 679 2417 4410 2947 4511 400 1352 3363 3978 1908 932 3848 4705 2784 2385 3517 3250 1844 1271 1681 2427 3912 274 2283 361 269 3325 842 4521 1058 3609 4183 2008 704 2433 1382 1300 3943 2789 1325 3056 2049 2633 890 1823 1299 686 1493 1733 1630 3085 4136 4133 702 2820 1766 2142 2126 2585 4241 2074 510 3841 2920 3298 1749 1840 4001 350 1002 2216 4306 4054 4945 3625 377 1278 4384 460 4062 4583 1831 4108 2890 2972 4473 49 1305 2884 667 4240 4174 1357 3548 4251 2546 3207 4713 2591 73 4458 1685 1143 731 4413 3779 81 4188 2077 2442 487 2758 1666 3523 4658 242 1401 951 4948 3824 1700 534 1117 4683 1761 3042 2949 506 694 1079 2170 1565 1792 386 4603 2898 688 3376 3477 4193 4392 860 4228 441 179 1482 2508 2834 3890 4008 1144 2256 4764 3766 3373 3924 567 1459 3970 1370 829 4575 3494 998 271 3922 965 1279 3920 1759 4553 3024 2473 755 4432 839 1306 675 4504 279 534 115 1615 1685 4934 4914 1800 3025 2012 679 2330 4228 4406 2106 3363 4817 2366 3229 170 1256 517 1228 3913 2130 4454 3119 4782 4009 4334 2534 2617 4892 1955 3691 836 714 2959 1808 1063 357 2121 2939 4266 3241 4429 325 3618 367 2391 4614 3472 3076 505 3247 4762 4305 1201 4485 2231 1815 132 2791 3691 4478 3062 2437 2272 715 1270 2649 1858 558 2166 4812 4134 2577 1962 672 2700 4129 4500 4415 1617 4719 4062 4540 4387 3707 3929 1531 4512 3474 4577 4247 225 4330 4248 4788 178 1992 3052 2903 4097 4742 2756 2913 910 692 4112 2184 2343 528 1784 3101 2314 717 1173 3971 416 3521 3050 3565 1786 4738 3136 3384 981 2919 4587 4804 3329 2173 493 708 1077 1597 318 1874 1963 1148 3630 3694 4595 1744 4187 3983 1689 1475 1932 4507 3157 276 4372 1656 3941 4487 1116 4358 1606 4372 1932 2990 1104 4308 3855 169 35 1075 2024 716 399 4298 2008 4268 4891 4355 3970 41 551 4152 1694 4505 986 79 1647 3052 2801 3836 1766 1589 481 959 268 2591 4215 639 2589 286 2889 4018 3948 613 4451 1216 4108 3858 3267 104 1764 4585 517 1074 3807 2454 2677 2427 2004 1025 2661 802 2367 1198 4308 434 235 4439 4030 4417 52 4252 1766 3683 2152 1532 203 1080 54 4770 970 2788 4785 2697 2071 2181 1072 4406 2152 2813 4913 1788 3634 1564 2625 2086 1650 2750 1305 4016 980 4669 4827 3850 51 2512 2819 1833 3921 2274 2250 1597 2372 2234 54 4872 1017 4670 4126 2127 4362 828 2286 1216 4504 3264 617 201 1058 66 4881 2453 4775 1038 3789 1301 2961 934 2646 4559 4361 4238 484 4878 563 3315 4415 1400 1510 2752 1069 1980 3623 4177 4654 152 2383 3610 4346 1787 1894 2617 775 3855 2393 4796 1821 1334 3444 3753 1911 82 2024 4567 2766 123 343 2407 653 1188 3848 4216 3691 3305 3996 2900 3830 598 1384 3637 3917 4350 1588 858 2665 1743 3463 1629 2485 767 1493 3080 2956 2757 841 986 4956 4056 1411 4530 4808 4975 2957 3158 2938 2677 3926 2470 4858 4492 966 1223 4950 4807 4835 2561 3494 1165 1902 1022 1660 3714 4922 4885 4694 768 1260 933 2464 3010 2943 4773 4186 1853 3828 3942 1392 3430 2920 4987 2816 992 1771 2992 4800 270 3130 3149 2922 3280 2491 3791 2154 388 2870 1094 52 3071 3463 2170 1417 2993 6 1847 1548 1193 2036 3096 978 893 1168 64 3244 1914 3205 667 1943 1930 85 4424 2637 3111 3011 4183 518 4152 2619 3798 3983 2734 3889 3320 4966 750 3570 4823 986 4923 1443 3607 3233 33 2170 2028 2111 2427 547 239 2182 2202 4151 3122 2986 4039 4532 3305 4981 822 1329 2396 3574 3238 4113 3244 2003 2848 98 2493 3344 3382 4680 2019 486 1478 1191 2604 940 983 2072 3091 3604 2379 1616 1780 3583 767 333 926 1218 264 4842 164 3594 4047 33 4994 920 3926 24 4262 2328 3053 1895 3800 2522 2131 2635 1508 2848 1928 2365 3998 3749 3237 3791 1194 1622 2333 4660 1236 2514 4171 1275 3080 3539 4515 3425 1439 1445 1196 1684 2014 958 2810 762 550 484 2816 4257 1451 3131 3825 3078 4290 2899 844 1415 356 255 3838 3372 3586 4265 3788 1747 3157 1778 3814 260 3006 3705 1196 63 1651 244 350 3039 939 2475 3241 710 3806 3075 3209 3105 3689 1093 1984 181 2695 3100 3069 1184 2606 2824 3097 2748 2170 1062 2605 3771 1669 2555 1251 2746 1935 3762 1426 4046 2548 2804 476 2881 2014 3127 4566 511 1056 682 3058

Для масиву на 1000 елементів буде здійснено пошук останнього елемента. Для звичайного масиву – число 3058, для відсортованого – 4994.

Масив на 10000 елементів: 15152 49778 38263 13548 27944 45568 35478 49177 32562 42944 6738 28357 13714 21662 5418 20190 1009 17229 9680 44371 6176 43701 12979 800 5831 33154 3355 28322 42173 14954 31676 329 4123 288 17100 41052 2864 25396 584 46422 39219 30605 246 11164 42675 33296 28016 31684 7460 13245 27184 49338 48019 42931 36116 21450 36799 37463 7717 44789 42212 7156 7003 17608 11267 6409 24234 13426 4561 14365 17326 25612 16644 13257 5152 25571 43454 1814 8124 22534 5137 21671 20861 28927 37769 … …42067 26804 34879 35250 43378 47629 8844 26262 49424 43467 42622 47785 29470 35109 33584 26189 2372 19107 7402 15660 34414 28571 5927 22942 16894 40062 34617 45222 23254 7099 18162 9137 12037 20411 21632 5627 40507 42480 2883 15003 42384 24421 8800 10558 26148 29336 37057 28572 32715 9208 21026 320 49733 21266 11413 13496 20877 11937 18356 3316 13011 13621 3007 42324 25304 33485 35747 9059 13476 27956 35682 9892 16223 42399 23276 9992 19770 45369 46793 37558 3004 6045 12506 14045 36072 17841 9403 40771 8997 8136 34750 28570 24123 49566 1992 28762 21413 16476 35253 42662 43588 27998 15095 4759 23526 11134 34097 7334 47990 1420 43725 25507 15143 30798 312 6965 1114 38962 6252 7462 24082 44975 61 5340 43205 7112 35768 16657 40407 44513 38579 3294 10115 4130 43969 25006 40949 7747 40519 24937 33744 2069 42781 7576 14411 9242 13426 43324 49254 45026 27648 32450 10223 27791 18580 48612 27936 9755 11553 16413 25366 17439 16927 3476 12841 40430 15392 11012 36829 25305 16016 43367 48442 31314 23717 22012 2194 42485 40928 33962 43685 42443 17402 45381 16865 6128 43070 32323 35549 45356 47070 31194 46193 24105 49571 16304 35142 39304 44735 23981 43850 2413 6495 23949 29009 34281 34008 28909 4073 46568 46418 35450 18827 27591 22728 32922 37 21545… … 37393 9673 22159 35427 8723 11840 26945 21157 44958 42644 14540 22101 45787 41061 49874 42801 47200 14817 45001 33129 16212 33065 9294 1643 4898 27554 5040 7637 11831 47250 43706 38305 654 22979 40628 5571 29275 49469 45228 13254 41330 29853 26252 10507 38309 25095 48838 33852 33136 33976 2707 40798 483 41556 44193 4327 378 20515 30529 31169 6800 19308 9325 47707 48938 26235 21447 22808 432 44302 13526

Для масиву в 10000 елементів буде здійснено пошук елемента під 9166 індексом, для звичайного масиву – число 26804, для відсортованого – число 45770.

**Список.**

Для списку в 10 елементів: 10 23 15 39 0 31 49 26 24 16

Для списку в 10 елементів буде здійснено пошук елемента під п’ятим індексом. Для звичайного списку – число 31, для відсортованого – число 24.

Для списку в 100 елементів: 498 491 13 353 462 335 403 211 47 490 280 110 128 373 58 227 383 281 275 41 363 440 217 154 193 254 329 263 339 20 367 103 467 13 218 455 258 83 430 190 97 289 394 258 373 227 241 407 336 224 381 481 147 491 432 340 273 359 160 207 6 139 372 27 122 177 142 114 155 103 469 299 351 85 443 74 45 459 280 465 12 421 427 114 139 386 455 476 81 378 182 398 423 222 183 457 417 367 136 195

Для списку в 100 елементів буде здійснено пошук першого елемента. Для звичайного списку – число 498, для відсортованого – число 6.

Для списку в 1000 елементів: 864 2905 846 4729 2773 2871 4516 374 909 1446 1273 4874 4833 462 987 4037 1632 3356 555 2111 4449 183 1333 4961 2835 4111 2274 4566 4387 3095 4615 356 3954 2226 4510 429 2253 787 1388 3269 713 693 922 3815 4990 2221 680 1853 1612 4745 2559 3499 3809 2331 2836 3959 1572 884 1893 3661 597 4949 986 2814 1830 917 919 2606 952 558 1783 844 1968 2286 1398 3755 4261 2517 4971 614 3431 420 2954 4642 535 1116 1547 1623 4389 550 3857 1369 3893 2726 2671 763 4706 3108 1984 4073 1301 3073 901 1054 2961 1714 1531 1522 932 4080 4697 4054 913 1279 230 176 1995 1344 2278 714 4369 4296 3217 401 1700 413 1951 4241 4614 635 4049 1152 533 898 4312 357 4519 1899 1680 3821 4585 24 691 308 852 4802 455 2614 2496 2495 3768 3361 830 1270 4703 2005 4856 499 4353 2547 4763 2289 1907 297 31 3545 3521 15 1967 4872 657 95 4664 3457 1128 4344 3604 2908 4548 1897 4958 4336 1745 2119 1866 688 322 4262 1194 2306 501 4019 2546 4132 4057 2296 3116 394 277 2307 1281 440 646 2624 1838 3673 3697 2373 142 358 3400 1947 951 2456 2589 426 544 4788 3430 4343 3222 4258 3820 4660 4370 1637 2549 532 4399 3832 1228 3210 2630 2240 615 4517 3689 4121 280 3193 1990 2949 4120 836 3905 3554 3067 90 1542 3630 1751 3328 1963 934 4084 2023 1620 986 3253 200 1124 3164 2973 1310 4130 190 4316 3710 1840 3071 1737 1423 4507 237 2352 273 137 2983 755 815 3569 2459 3989 769 2080 2899 1247 1696 3155 3592 2897 2703 867 80 2068 3826 4976 2810 1706 3715 4237 4357 3249 3470 1893 328 3820 1725 3582 3811 1885 3636 230 2437 1630 3665 4175 2204 4229 1230 3942 2620 554 3247 174 4033 556 4427 3168 3526 297 2327 1277 2040 1578 4212 4209 518 3876 1752 4079 4521 3113 4344 1706 4261 2473 3429 3450 3403 4650 2772 3581 475 4772 1616 3802 1 3296 2859 4772 4392 3557 56 3514 4558 2359 3190 859 2417 4456 3656 3327 2476 2150 4421 4506 1210 1541 913 1560 2126 977 4764 3875 2525 3745 801 2267 4961 410 4207 2222 1016 1980 4686 964 3057 829 2147 1902 4283 2569 1032 3947 994 4444 1104 3325 351 2109 2591 3460 2383 1299 2645 3415 3792 1181 989 812 1557 922 898 2007 248 1434 2311 496 2463 3457 1449 380 4393 4010 2277 4556 4944 817 2880 3080 2641 2476 1916 2851 2818 747 4838 4716 4335 1271 2171 1876 4840 1335 344 3361 1671 4024 3698 4560 2132 608 861 1893 660 2211 4066 3372 4022 3088 3470 2975 3300 2908 3171 4081 4005 3046 4189 4501 1595 2594 1160 1192 1285 4572 540 3058 3666 3916 2885 920 2118 3320 948 2033 1614 23 2191 607 1680 1466 694 1246 2800 4196 3576 1931 3163 1263 4355 3625 4835 4197 2965 4537 4447 4669 608 1087 1670 1008 4706 106 203 2550 856 4980 1959 1138 2466 3982 855 3582 2821 129 1900 4914 3391 2089 996 3608 4894 2403 4560 4259 1085 2921 355 1410 2585 1945 4415 1103 3936 672 1760 588 1043 250 3339 3596 4971 2025 3796 373 2769 1252 1376 2835 2637 4533 1277 3518 90 3061 1113 2302 546 943 1465 2935 4624 549 3552 520 4567 4752 4645 2148 3368 1311 3870 3141 3749 2400 3636 2869 3606 4186 1490 4832 1545 2519 3772 3052 668 896 1012 875 4559 4457 41 100 3784 404 3971 4422 3472 3276 806 3017 1606 4227 4467 1325 662 3135 4769 2690 4424 3243 3696 1359 4452 1633 1390 2105 1776 500 4852 3670 3739 3770 2589 3911 1269 3770 4357 3345 3429 4214 4396 330 3379 3472 3225 2317 3052 2447 2389 2532 1127 3322 1450 1315 761 3682 647 2151 4014 1865 2646 2972 3423 2136 4278 2836 1637 555 55 1892 3778 1424 4260 210 3847 3365 2980 980 3407 4072 4053 4329 3111 2319 2543 416 2319 1274 3149 2587 3123 2205 4330 1563 1568 1423 1906 1243 4194 388 679 2496 2391 497 3753 4703 4062 1240 2467 2395 701 4667 3744 4941 2812 3316 4861 1103 4592 3861 516 363 2405 2467 3769 1573 2093 2697 3641 2423 1458 1736 1785 3018 3392 1557 1937 2613 3566 2840 1354 1079 3806 754 1885 3456 3461 4389 3751 3251 1562 803 1650 332 4872 316 90 4923 1727 2180 2609 1364 2598 44 937 3965 2881 725 1549 1255 1379 2247 2537 1752 2507 4437 1557 1651 581 313 3111 2704 3946 390 861 655 85 1452 3146 3076 1466 2013 885 1385 230 4989 3480 3762 4817 2920 574 2736 2839 2495 1872 4314 3265 4898 2825 434 3759 3439 4341 1414 4068 4905 2418 2208 4183 281 3879 1429 2579 3473 4788 4365 1361 1151 1522 2517 956 2794 1834 2661 4738 375 4864 1450 3518 1546 2389 187 3626 2711 4316 3027 2671 4471 2316 324 2570 1272 4579 4535 4041 4145 156 4366 2706 2506 2953 2114 3375 308 4478 964 1604 1680 1676 3692 41 968 3690 2636 2892 3691 2802 4867 4156 1761 1693 1890 3834 1197 4946 4684 3215 2299 3619 593 218 497 3743 1011 3592 275 1812 3318 3232 3352 1422 991 2795 3624 282 1602 2581 1942 1642 349 1343 288 210 945 812 1063 3279 2177 361 4793 2748 4305 3061 1082 3473 4544 2224 2678 4044 2617 1879 4483 1634 803 338 702 4095 210 914 3344 2102 1273 656 3269 4250 4930 286 2797 198 2529 1996 4634 2957 3439 603 2117 2930 1712 150 738 4377 4903 2898 4025 3470 1860 3653 4407 607 110 1967 3068 1262 2017 2691 1475 3649 4791 1832 3543 4257 4426 2479 1514 621 2245 1525

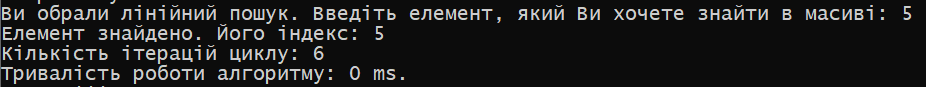
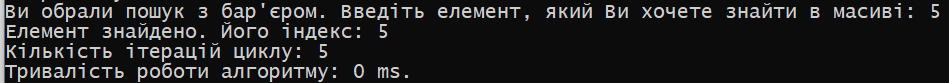
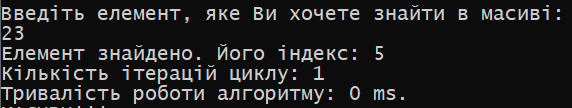
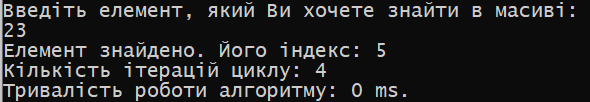
Для списку в 1000 елементів буде здійснено пошук останнього елемента. Для звичайного списку – число 1525, для відсортованого – число 4990.

Для списку в 10000 елементів: 48688 1783 26540 42929 47688 8166 48000 3482 17293 5283 5941 31762 35174 31754 22805 30627 18288 17318 36073 24154 7944 17804 42683 47786 32932 35486 41670 46335 32735 1708 17964 13708 37465 1322 4817 6471 31522 7439 44381 47363 6175 7197 9159 30400 41323 33789 43405 13392 24563 33636 32742 19197 11997 42715 32675 43038 9099 28754 9997 12202 16496 1664 20746 15585 37319 42233 44296 33851 26937 16333 49105 10849 22937 38709 17979 747 8644 12283 6462 49142 42080 28277 21772 49098 18966 48766 1710 44749 18647 11778 47372 2767 47441 32179 30867 4510 36450 43573 43080 49090 39492 9553 36455 8230 34531 21892 46260 23287 24735 31928 34393 46815 22291 10854 20121 38218 29892 21648 46619 38552 40522 49546 15204 15158 18961 46337 13408 40758 7841 13469 14297 15070 19203 7372 9650 32526 41822 13542 14566 47073 2506 28423 20014 36719 27384 556 30476 36586 12057 42649 24618 14801 46954 4527 8567 39945 44349 21296 39268 38194 8484 5501 15445 11266 17630 19322 27612 14919 1204 37595 46396 16350 7081 49545 36046 12099 29532 28485 37773 18405 15861 26821 28701 15192 38851 49495 18116 14675 48805 19704 38176 20525 24273 26372 38589 47004 12977 8747 19088 8061 22943 15556 35382 24461 46253 8267 7720 47409 18481 46468 10413 15864 33522 20863 22333 31663 26800 253 22415 18135 1206 12937 16114 31499 49418 25870 42076 30709…

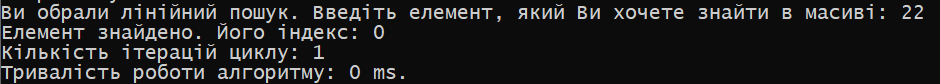
…48170 47258 46431 23667 43673 38578 41564 45608 31078 20984 41111 7872 3101 3038 3130 14571 16747 13497 27463 12404 41684 42368 32228 19205 44435 44741 34497 20888 19623 10321 154 18672 10135 3710 14413 8103 42625 34760 18703 30917 11865 11760 48081 33229 46337 19400 26772 37357 43657 11504 41429 26826 39929 26395 3601 5802 15030 27225 29231 48931 4080 20675 25984 20756 20829 22438 47736 49391 38624 45027 21946 31987 44794 46545 538 29924 44286 48999 22868 25035 17969 47140 27230 8119 18891 23327 28743 33739 109 8611 43072 15399 5528 19771 26837 41190 35775 27324 12399 23898 21664 27381 48733 48630 39557 9442 32032 43384 25857 23676 11516 16031 4357 4196 30962 6940 43444 17865 1864 47501 43695 13996 49281 30013 1954 6547 26458 25023 19708 9347 44148 16961 36599 48970 3371 40587 48406 28600 18561 9449 41295 35359 7882 26432 47094 27041 11041 1332 38808 19896 47745 17909 25459 14897 30202 7667 28100 18719 46675 33010 32984 7009 23675 16510 29528 44555 29431 5387 825 40374 8533 14843 49304 42415 6206 8336 6114 22848 32918 24913 45505 25125 36214 49811 11601 26238 41501 21702 18768 45703 12487 29686 31924 35551 26465 34286 11683 41371 46904 2539 47609 5654 507 48434 11363 32902 18605 33044 8690 21865 1553 5251 35801 21762 37505 7858 20794 23864 4908 3289 3053 7729 36619 5121 27887 28846 32918 13753 15950 21920 46653 14742 25943 30378 27303 39850 24618 37780 38448 28699 7633 8457 13011 46902 44150 7235 43885 10162 48045 18606 13491 37819 36463 43614 49486 39879 19035 45386 20546 32516 49983 4852 17094 36770 25212 36810 29308 5423 44458 47654 33240 33014 35415 26208 45656 44596 44718 39716 10970 20651 34961 22756 15707 47343 8428 8834 28279 32328 30892 37423 20814 29232 17233 5931 28716 2781 41363 26240 21689 7340 27927 38462 15703 390 35366 30476 2403 10254 47958 4889 45161 29319 34415 49894 47555 27226 39144 19750 28341 16378 8531 46979 24530 7035 26840 4007 15781 29483 47492 42875 3233 18477 18026 3630 42724 46498 7052 15316 11976 27951 6430 18024 34370 26002 42262 41495 44816 17338 8376 1489 13636 21612 47898 5311 48809 30948 13931 8668 23550 31359 14695 22920 12761 5082 1656 26683 11292 30784 7170 1056 20173 23827 7774 390 9947 40506 12608 48528 14772 35345 9190 48442 21106 46002 29238 31620 20579 12715 4821 11776 32567 48384 41766 30617 13255 33510 5262 29287 24345 15579 30203 14032 10168 7319 31315 39809 13838 47508 45364 8303 18339 15402 43896 38204 22168 16253 1813 16759 25844 20035 6104 48577 25963 45394 18489 21789 32060 27156 37134 26437 35243 33321 24183 49192 5142 45158 38273 13787 14686 39429 17782 23070 17350 46518 43436 17164 4488 3561 8449 47002 31696 38503 3443 4309 9475 31625 38069 14774 38830 9526 7748 36681 10374 18927 23059 35018 41219 44703 33062 27009 27979 38025 2072 36414 2253 33033 31226 29444 1957 1324 17300 28499 18707 40132 44740 46739 29877 19873 39717 23516 7089 23499 24957 5160 31681 31101 12696 48423 23458 8418 25945 9785 20498 31439 19022 43670 36431 17028 2056 26442 399 8624 12816 37505 42225 29249 17973 20241 24186 26320 5140 21345 4985 9546 19919 4479 13068 46912 4733 21151 20336 32802 5986 43539 25379 7515 8000 37267 21110 1069 10308 12849 17817 13275 23116 48465 10683 37451 12934 2432 13128 42454 24236 47138 3277 4599 4799 10952 11519 14542 30602 39518 12294 30904 6105 17596 2637 19277 12125 34710 21248 30705 49131 23117 16011 42291 3528 41709 36429 21454 43796 25616 33978 2301 8023 27882 8566 8848 40261 20780 2715 47047 25748 6567 20467 20789 554 36912 7170 5520 45827 41406 25326 28224 31183 32422 43322 1119 31126 10986 1070 13089 24522 40065 20745 4985 5540 9993 22881 28223 9030 43789 29015 41344 31994 24201 4957 42563 2650 45221 41736 16615 34539 30909 25626 2390 290 5753 21118 25600 34559 7446 27721 29274 19709 39625 22524 35682 35822 15481 15249 40561 14030 28947 46489 2038 12390 33981 46229 6981 27465 759 48468 35904 19249 34454 28550 43613 14438 18355 4695 49786 38874 47281 43664 1583 16067 49741 21634 42368 1676 19274 6741 37168 29972 1175 2585 5592 34419 23588 40000 16308 9524 14136 48135 33800 8978 41817 10025 35255 11075 28911 21244 17466 10786 15462 1686 16749 35282 44905 35971 12649 12347 3860 5305 8190 44017 11299 5932 18074 31869 22957 9193 40850 28355 38386 40262 24737 49146 9863 42605 24250 39715 36379 31293 22765 48030 39274 26382 14509 49489 35835 20309 39514 37616 27350 3960 5663 8831 39944 39782 47678 35884 41950 3385 38118 19718 30393 39111 22399 25200 26948 17603 25419 2300 11720 22933 25967 22567 32540 10160 4742 34917 41422 18585 46034 2647 19679 3233 1004 34426 34599 15906 9201 20080 44926 38365 47764 32994 43174 3046 17637 163 3982 39309 22540 18231 44890 37213 25895 4417 27993 33095 26291 29784 35040 12760 44462 23365 7351 35470 38 27159 38107 37973 40601 11042 8402 5338 7373 23355 25168 42973 29393 29493 42522 4579 30935 2113 46044 27802 7756 32465 27337 46587 6432 1503 39615 29416 35040 32166 43902 24399 25642 7703 3008 40477 12055 16009 48708 11497 9829 39552 29839 2540 29248 35019 3702 46797 37261 30460 31825 42349 27321 29548 27714 17572 49821 41519 31540 39097 21427 28985 20297 25207 29452 769 17330 21689 26485 2045 42523 14926 3405 34546 17973 18204 2626 24797 17338 21413 47801 42817 42155 4579 340 1553 47077 46094 29989 35435 40655 20535 34468 9610 40070 30844 19254 4632 23087 28479 17688 32012 20312 35216 37937 16899 38944 42774 11575 9509 14946 25024 24181 10127 41296 28609 36830 15717 24866 27898 3691 21236 41699 41050 11389 21987 30457 43794 44475 37129 48949 47994 1710 38859 3725 15788 22505 6939 16642 33441 12609 4303 34518 20480 20488 15631 46354 24341 18314 11461 44013 3536 29765 45188 24787 46451 40312 29262 23826 15949 36441 45149 48298 24446 10559 16951 23313 35321 6401 25507 6104 29890 49075 41425 15289 49388 36718 21219 20561 45758 25632 6116 25480 33014 43115 45412 18229 6521 14071 7273 25476 43242 33113 692 46999 17460 6004 10072 9920 3536 42317 11032 17939 42806 5356 14123 4461 38340 29899 25399 9733 19092 8701 28067 40316 30325 19668 15284 31331 15147 48722 16791 21250 49128 31 12861 46777 48382 44425 47829 43383 26645 11299 17024 3440 14599 38176 32673 27657 28992 40951 29888 26622 38672 47540 6383 34541 5045 10376 16674 47791 40720 28741 44773 4813 25525 39782 18811 42775 42494 17345 6078 43915 32069 32016 33087 7793 41676 24627 25716 42148 36995 37626 2339 24195 18834 40169 32577 1588 43647 28320 41732 38006 27750 37 2662 47904 16526 12210 27914 24817 19365 39898 35163 11646 34872 35973 44552 6656 14452 48589 42865 30418 34659 24526 10796 3094 26402 20577 35979 35356 7204 17223 49127 14174 25612 18072 16164 32031 29353 35183 41267 29466 46712 899 22782 47097 2463 40692 39322 32860 45616 25921 37135 45058 35455 11314 3346 3223 39241 49568 21502 45948 26230 42557 17612 2142 40770 20988 36033 16800 19809 12521 27302 29269 7322 13398 33705 33760 1744 13698 631 35709 31255 3119 39740 31283 30087 4069 28719 14298 14725 12941 42790 29785 13680 45319 3235 4154 33287 20640 6326 31474 4658 22522 13051 33095 48619 7866 37736 22056 27608 19586 1479 25542 48936 35792 14693 23111 2817 36329 22054 36700 42269 21735 2074 6867 19731 47517 15588 12002 10163 29551 472 31104 7372 19157 31051 8732 40068 6645 32664 22220 16333 6663 36690 45138 11461 17247 30849 27888 30626 30124 1337 46957 48585 19626 39205 32923 20447 6183 13363 1101 42278 25733 11892 48056 19113 20374 18170 29779 4741 5960 33043 46171 3665 49833 20367 35605 35044 6935 45405 2483 16668 37700 31376 30039 28213 3515 32519 37745 29951 48127 38284 33884 43281 31562 29942 40600 44770 38633 26024 41087 49076 1069 23146 24665 47081 5166 43292 48751 49259 3599 47878 13511 10777 10880 34433 4578 44356 31853 19843 15598 37855 30425 49827 6613 17676 49158 2889 22103 19891 29766 40835 46411 30911 4318 3407 15598 14553 6711 32958 23047 10222 33767 38486

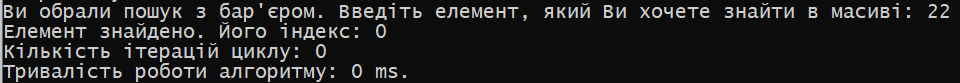
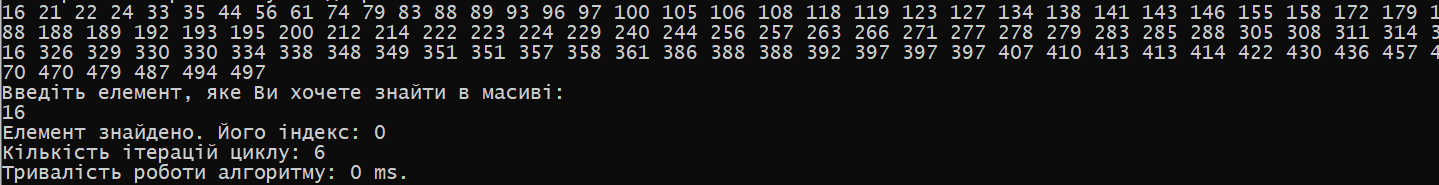
Для списку в 10000 елементів буде здійснено пошук елемента з індексом 9166. Для звичайного списку – число 31439, для відсортованого – число 46016.

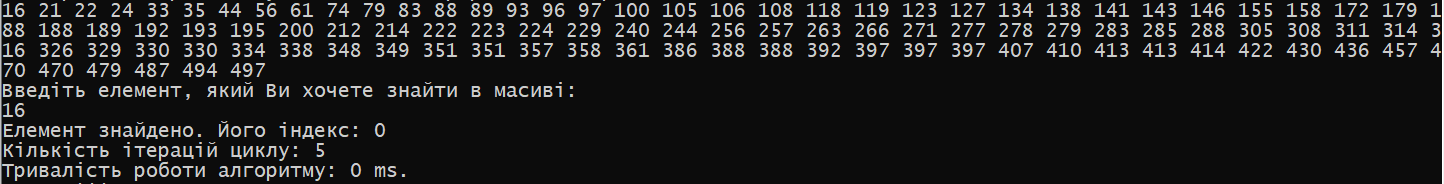
1. Результати тестування програми та аналіз отриманих помилок.

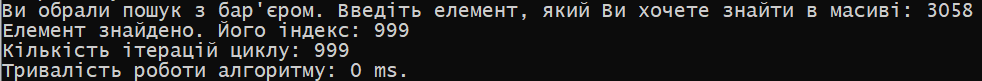
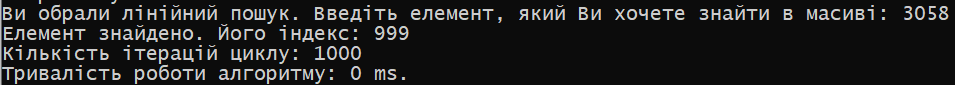
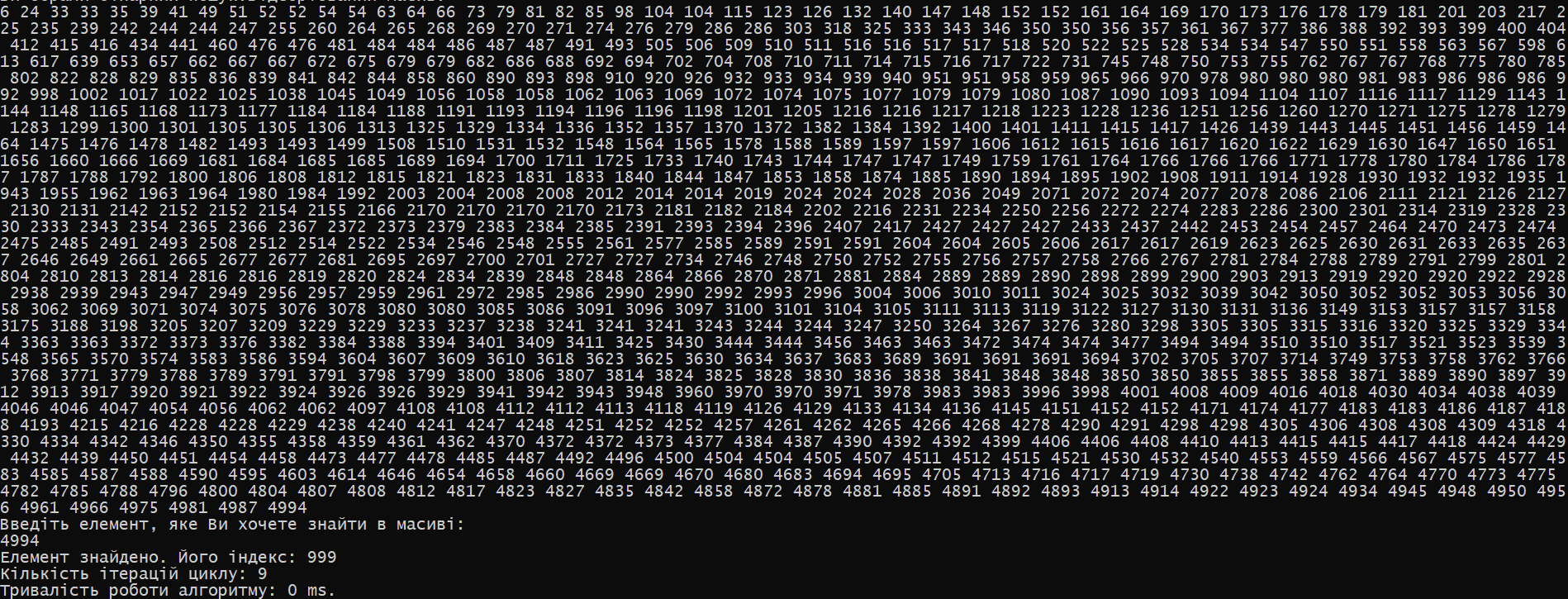
Для масиву в 10 елементів:

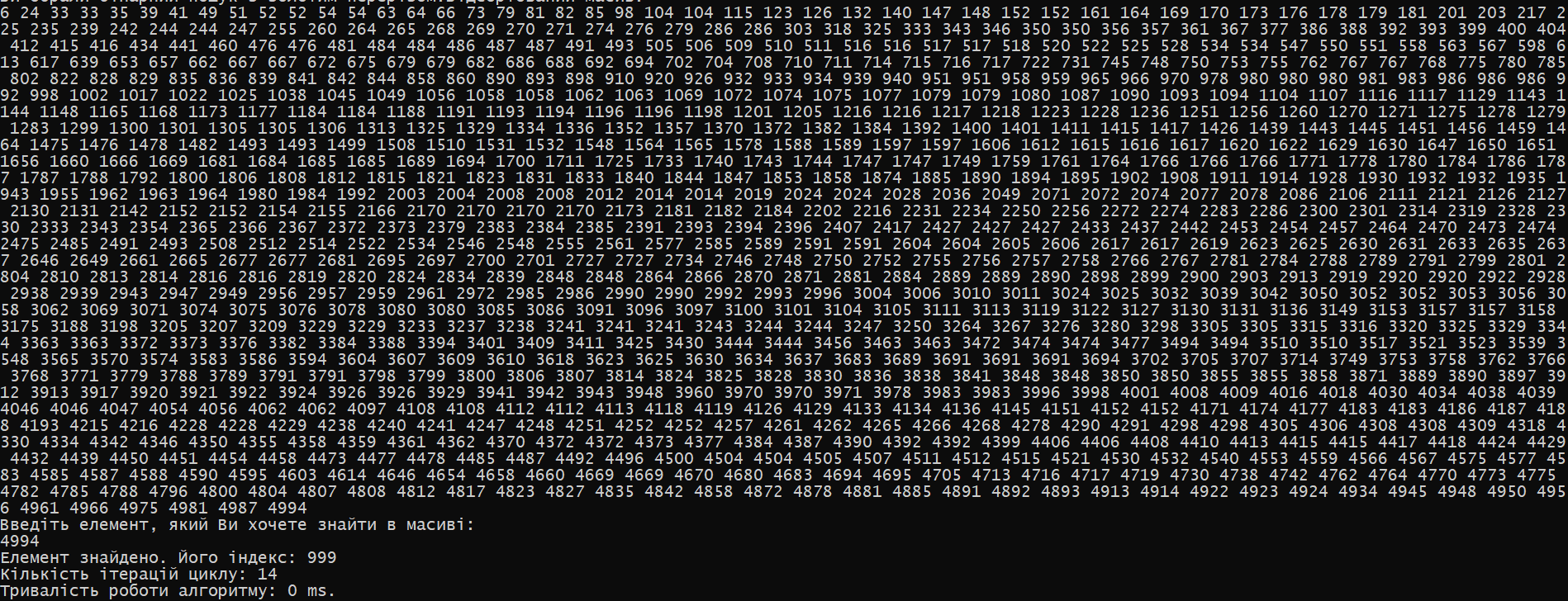
Для масиву в 100 елементів:



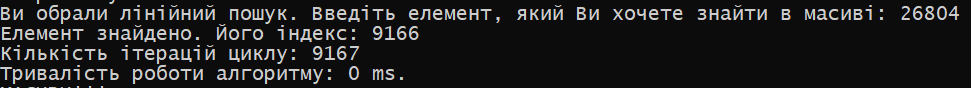
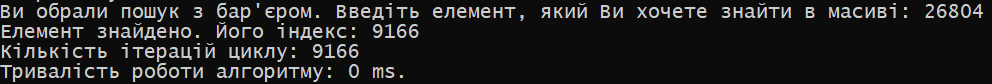


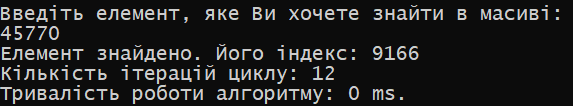


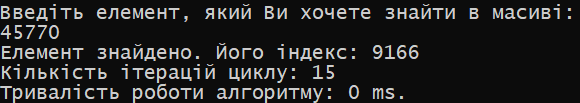
Для масиву в 1000 елементів:

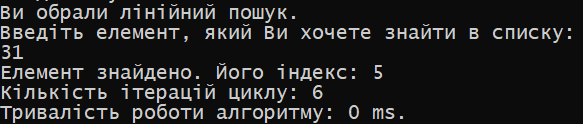


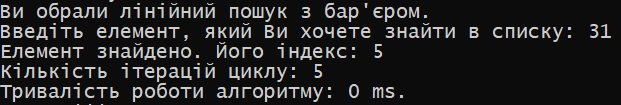
Для масиву в 10000 елементів:

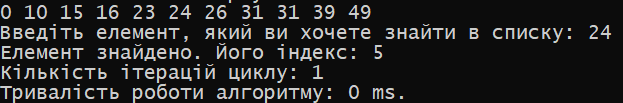


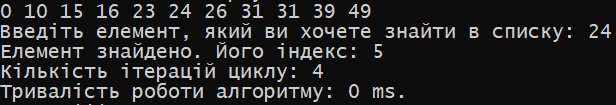




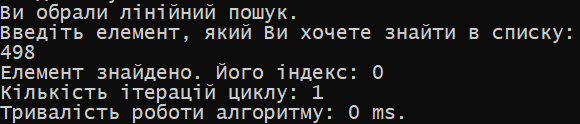
Для списку в 10 елементів:

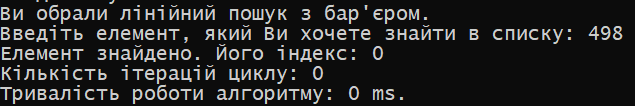


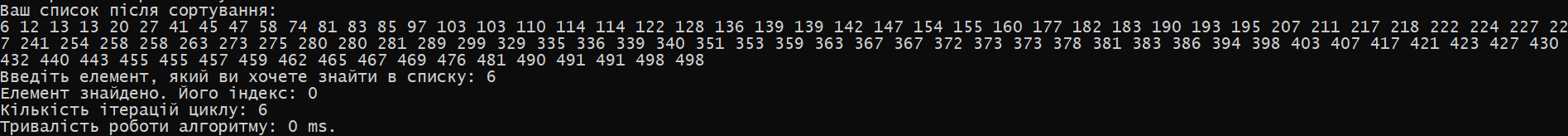


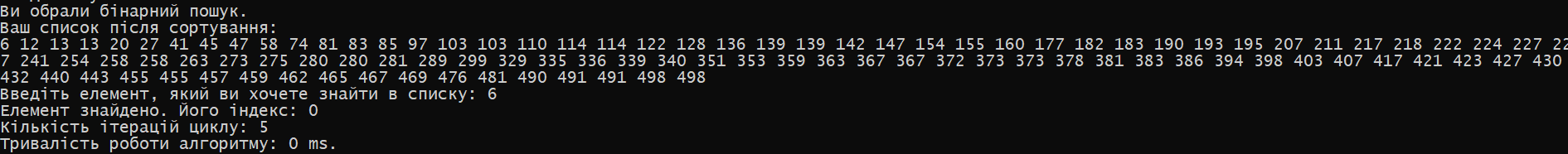


Для списку в 100 елементів:

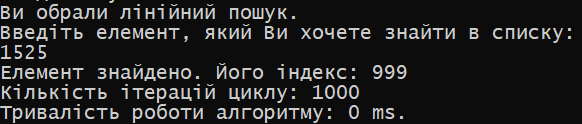


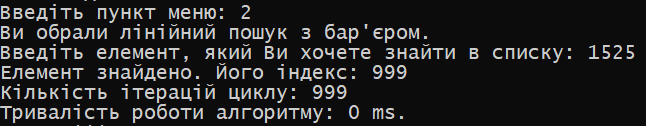


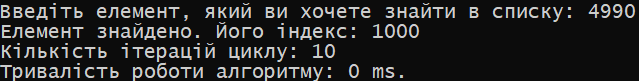


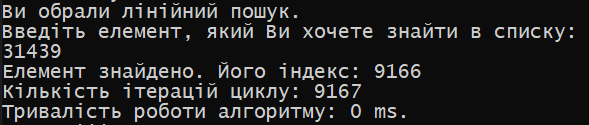


Для списку в 1000 елементів:

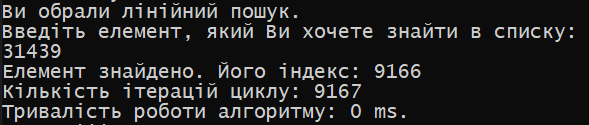


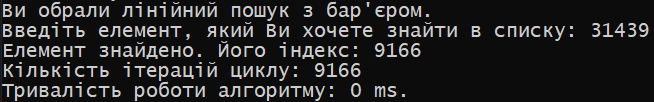


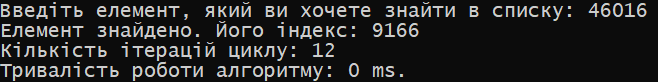


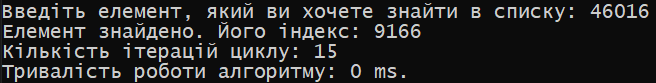


Для списку в 10000 елементів:









**Висновок та аналіз тестів:** Згідно з отриманими результатами виконання алгоритмів програмою можна зробити певні висновки. Найефективнішим алгоритмом пошуку є бінарний, адже він витрачає менше ітерацій на пошук елемента, який знаходиться **НЕ НА ПОЧАТКУ** масиву, проте повинна бути виконана основна його умова – сортування структури даних. Його високу ефективність можна пояснити тим, що після кожної ітерації циклу, масив звужується вдвічі. В порівнянні з бінарним пошуком із золотим перерізом, там масив звужується не в два рази, а в 1,6 раза. Якщо шуканий елемент знаходиться на початку масиву, то найефективнішими будуть пошук лінійний та з бар’єром. Насправді ці два пошуки майже не відрізняються між собою. Єдине, що в даній програмі пошук з бар’єром робив на одну ітерацію менше, аніж звичайний лінійний пошук. Так відбувалося через те, що в умові виконання циклу лінійного пошуку не було перевірки на те, чи поточний елемент дорівнює шуканому, вона була безпосередньо в самому циклі. Тому цикл спочатку робив ітерацію, а потім перевірку.

Крім того, можна зробити висновок, що структура даних ніяк не впливає на ефективність алгоритму. Можна побачити, що були отримані абсолютно однакові результати роботи програми як над масивами так і над списками.

Недоліки: при роботі із списком була виявлена помилка у виконанні програми, при використанні пошуку з бар’єром. Тобто, якщо ми спочатку виконаємо пошук з бар’єром, то суть даного алгоритму в тому, щоб додати шуканий елемент в кінець списку. Тому подальші алгоритми пошуку будуть видавати помилку, так як вони будуть знаходити саме цей останній доданий елемент, тому наприклад коли елементу в списку не було, то після пошуку з бар’єром він з’являється в кінці списку і інші алгоритми пошуку вже будуть його знаходити. Це можна “пофіксити”, створивши метод видалення останнього елемента списку.