Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра ІПІ

Звіт

з лабораторної роботи № 1 з дисципліни
«Алгоритми та структури даних 2. Структури даних»

„ **Проектування і аналіз алгоритмів внутрішнього сортування**”

Виконав(ла) *ІП-15 Плугатирьов Дмитро Валерійович*
(шифр, прізвище, ім'я, по батькові)

Перевірив *Соколовський Владислав Володимирович*
(прізвище, ім'я, по батькові)

Київ 2022

ЗМІСТ

Мета лабораторної роботи

Мета роботи – вивчити основні методи аналізу обчислювальної складності алгоритмів внутрішнього сортування і оцінити поріг їх ефективності.

Завдання

Виконати аналіз алгоритму внутрішнього сортування на відповідність наступним властивостям (таблиця 2.1):

– стійкість;

– «природність» поведінки (Adaptability);

– базуються на порівняннях;

– необхідність додаткової пам'яті (об'єму);

– необхідність в знаннях про структуру даних.

Записати алгоритм внутрішнього сортування за допомогою псевдокоду (чи іншого способу по вибору).

Провести аналіз часової складності в гіршому, кращому і середньому випадках та записати часову складність в асимптотичних оцінках.

Виконати програмну реалізацію алгоритму на будь-якій мові програмування з фіксацією часових характеристик оцінювання (кількість порівнянь, кількість перестановок, глибина рекурсивного поглиблення та інше в залежності від алгоритму).

Провести ряд випробувань алгоритму на масивах різної розмірності (10, 100, 1000, 5000, 10000, 20000, 50000 елементів) і різних наборів вхідних даних (впорядкований масив, зворотно упорядкований масив, масив випадкових чисел) і побудувати графіки залежності часових характеристик оцінювання від розмірності масиву, нанести на графік асимптотичну оцінку гіршого і кращого випадків для порівняння.

Зробити порівняльний аналіз двох алгоритмів.

Зробити узагальнений висновок з лабораторної роботи.

Таблиця 2.1 – Варіанти алгоритмів

№	**Алгоритм сортування**
1	Сортування бульбашкою
2	Сортування гребінцем («розчіскою»)

ВИКОНАННЯ

1.1 Аналіз алгоритму на відповідність властивостям

Аналіз алгоритму сортування бульбашкою на відповідність властивостям наведено в таблиці 3.1.

Таблиця 3.1 – Аналіз алгоритму на відповідність властивостям

Властивість	**Сортування бульбашкою**
Стійкість	Стійкий
«Природність» поведінки (Adaptability)	Природна
Базуються на порівняннях	Так
Необхідність в додатковій пам'яті (об'єм)	Так. В даному прикладі бралися 4 байти пам’яті на додаткову цілочисельну змінну
Необхідність в знаннях про структури даних	Так. В даному випадку використовувався звичайний масив

Аналіз алгоритму сортування гребінцем на відповідність властивостям наведено в таблиці 3.2.

Таблиця 3.2 – Аналіз алгоритму на відповідність властивостям

Властивість	**Сортування гребінцем**
Стійкість	Стійкий
«Природність» поведінки (Adaptability)	Природна
Базуються на порівняннях	Так
Необхідність в додатковій пам'яті (об'єм)	Так. В даному прикладі бралися 12 байтів пам’яті на додаткову цілочисельну та з фіксованою крапкою змінні.

Необхідність в знаннях про структури даних	Так. В даному випадку використовувався звичайний масив

1.2 Псевдокод алгоритму сортування «бульбашкою»

Крок 1. Визначити основні дії.

Крок 2. Прохід по всій послідовності арифметичним циклом певну кількість разів.

Крок 3. Прохід по всіх елементах послідовності.

Крок 4. Порівняння елементів, які розташовані поруч, та їх обмін місцями за певної умови.

Крок 1

початок

прохід по всій послідовності арифметичним циклом певну кількість разів

прохід по всіх елементах послідовності

порівняння елементів, які розташовані поруч, та їх обмін місцями за певної умови

кінець

Крок 2

початок

повторити для y **від** 0 **до** seq.size - 1

прохід по всіх елементах послідовності

порівняння елементів, які розташовані поруч, та їх обмін місцями за певної умови

все повторити

кінець

Крок 3

початок

повторити для y **від** 0 **до** seq.size - 1

повторити для i **від** 1 **до** seq.size

порівняння елементів, які розташовані поруч, та їх обмін місцями за певної умови

все повторити

все повторити

кінець

Крок 4

початок

повторити для y **від** 0 **до** seq.size - 1

повторити для i **від** 1 **до** seq.size

якщо seq[i - 1] > seq[i]

то

temp := seq[i]

seq[i] := seq[i - 1]

seq[i - 1] := temp

все якщо

все повторити

все повторити

кінець

Псевдокод алгоритму сортування «гребінцем»

Крок 1. Визначити основні дії.

Крок 2. Виконання ітерацій по всій послідовності алгоритмом «бульбашка» до моменту зменшення кроку до 1 або менше.

Крок 3. Перебір елементів послідовності для подальших дій з ними.

Крок 4. Порівняння елементів, які розташовані поруч, та їх обмін місцями за певної умови.

Крок 1

початок

виконання ітерацій по всій послідовності алгоритмом «бульбашка» до моменту зменшення кроку до 1 або менше

перебір елементів послідовності для подальших дій з ними порівняння елементів, які розташовані поруч, та їх обмін місцями за певної умови

кінець

Крок 2

початок

factor := 1.2473309

step := array.size – 1

повторити

поки step >= 1

перебір елементів послідовності для подальших дій з ними порівняння елементів, які розташовані поруч, та їх обмін місцями за певної умови

step := (int)(step / factor)

все повторити

кінець

Крок 3

початок

factor := 1.2473309

step := array.size – 1

повторити

поки step >= 1

для i **від** 0 **до** array.size

повторити

порівняння елементів, які розташовані поруч, та їх обмін місцями за певної умови

все повторити

step := (int)(step / factor)

все повторити

кінець

Крок 4

початок

factor := 1.2473309

step := array.size – 1

повторити

поки step >= 1

для i **від** 0 **до** array.size

повторити

якщо array[i] > array[i + step]

то

Swap(array[i], array[i + step])

все якщо

все повторити

step := (int)(step / factor)

все повторити

кінець

Підпрограми

Swap(value1, value2)

temp := value1

value1 := value2

value2 := temp

кінець

1.3 Аналіз часової складності

Метод сортування «бульбашкою»

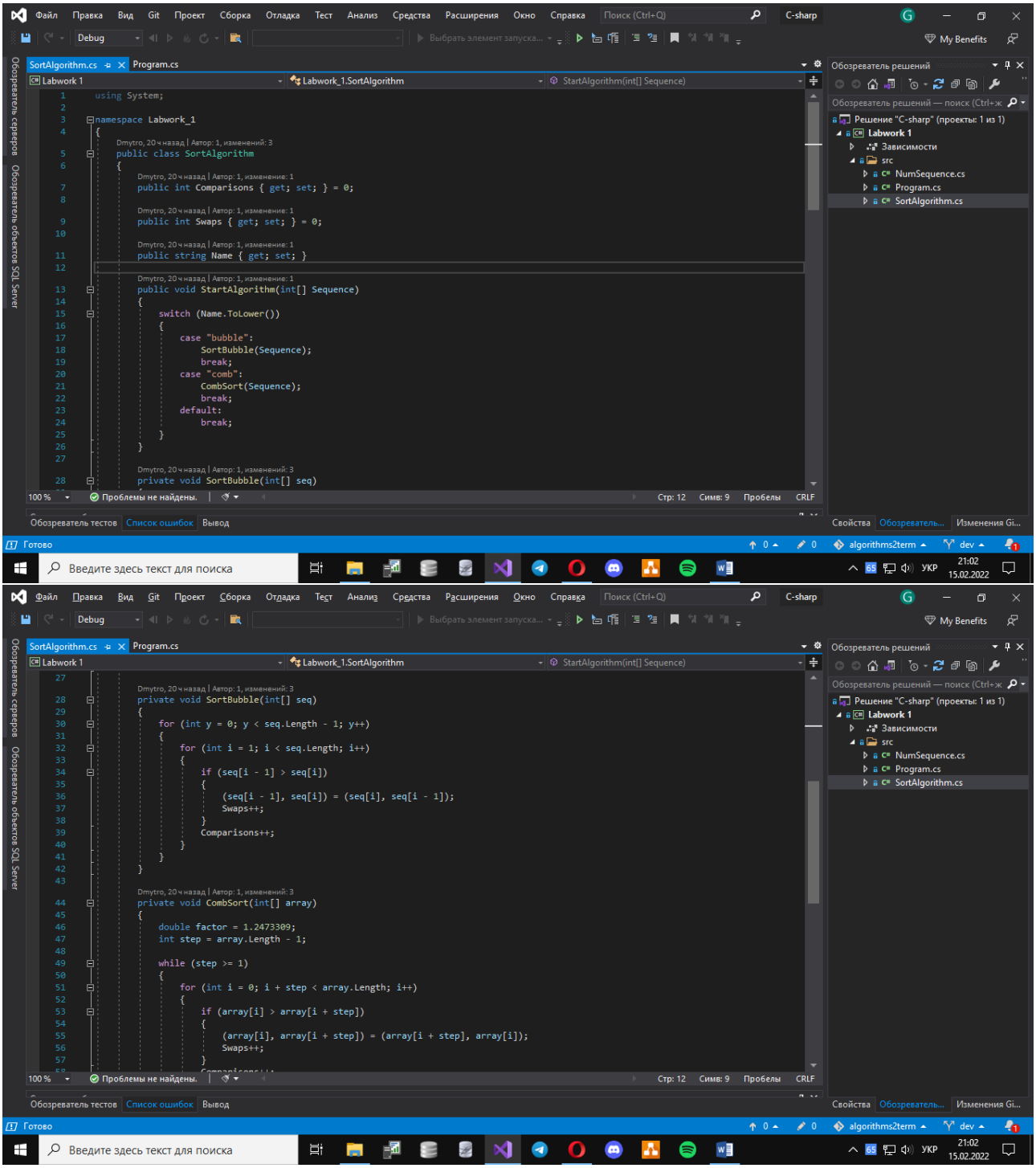
К-сть елементів	Час		
	Гірший $O(n^2)$	**Середньостатистичний** $O(n^2)$	**Кращий** $O(n)$
10	00:00:00.0000056	00:00:00.0000057	00:00:00.0000054
100	00:00:00.0000271	00:00:00.0000339	00:00:00.0000195
1000	00:00:00.0030748	00:00:00.0028864	00:00:00.0017776

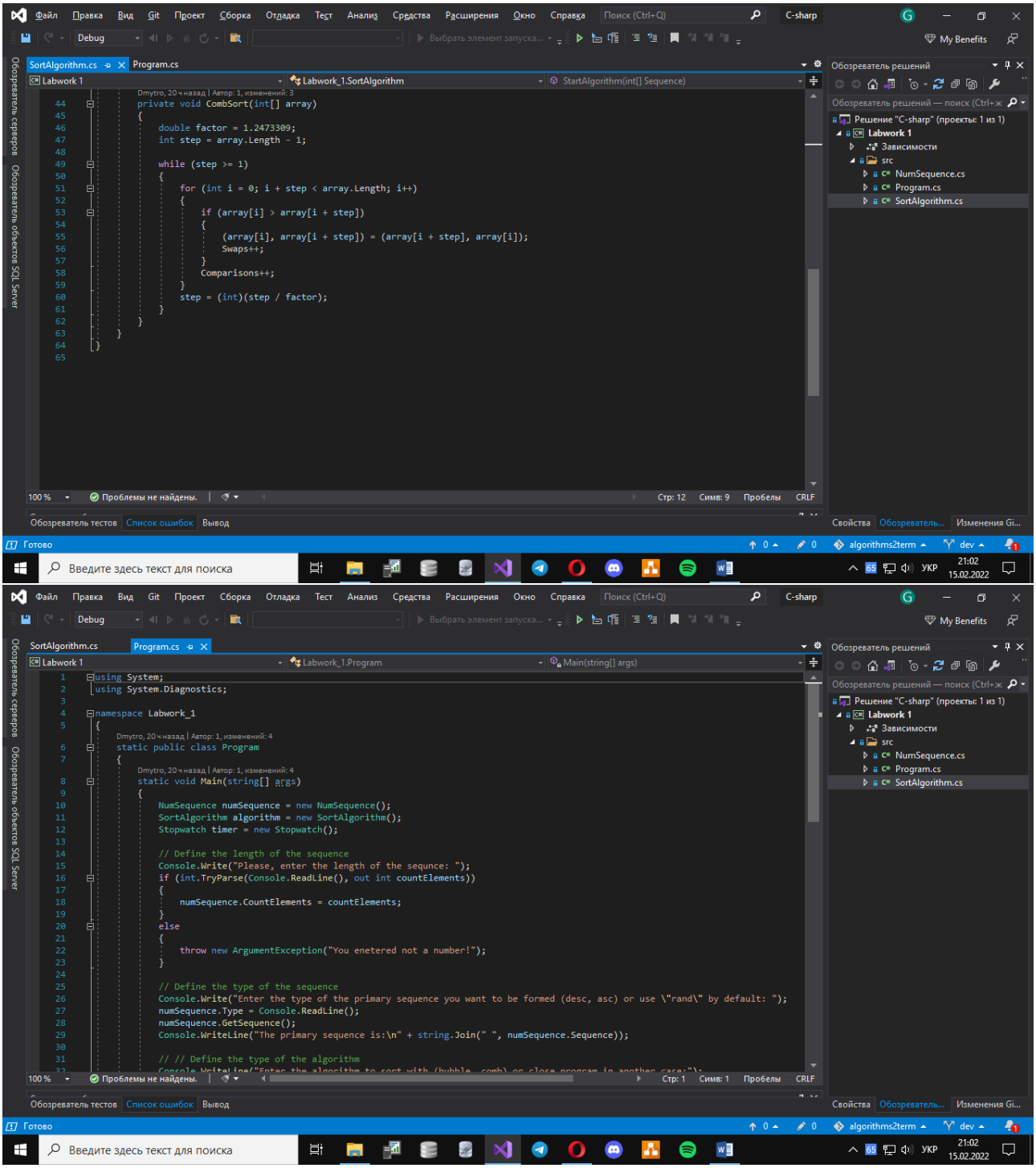
Метод сортування “гребінцем”

К-сть елементів	Час		
	Гірший $O(n^2)$	**Середньостатистичний** $\Theta(n_3 \setminus 3_b)$	**Кращий** $O(n \log n)$
10	00:00:00.0000083	00:00:00.0000087	00:00:00.0000054
100	00:00:00.0000280	00:00:00.0000116	00:00:00.0000074
1000	00:00:00.0030445	00:00:00.0001028	00:00:00.0000431

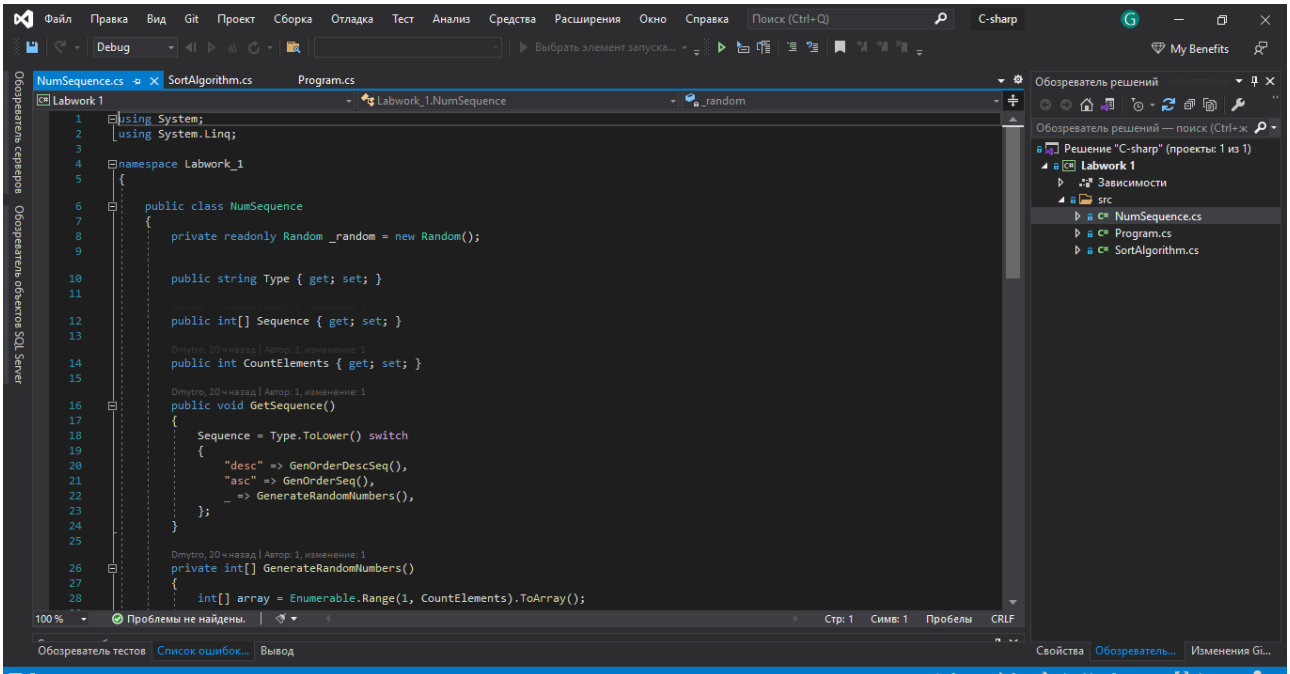
1.4 Програмна реалізація алгоритму

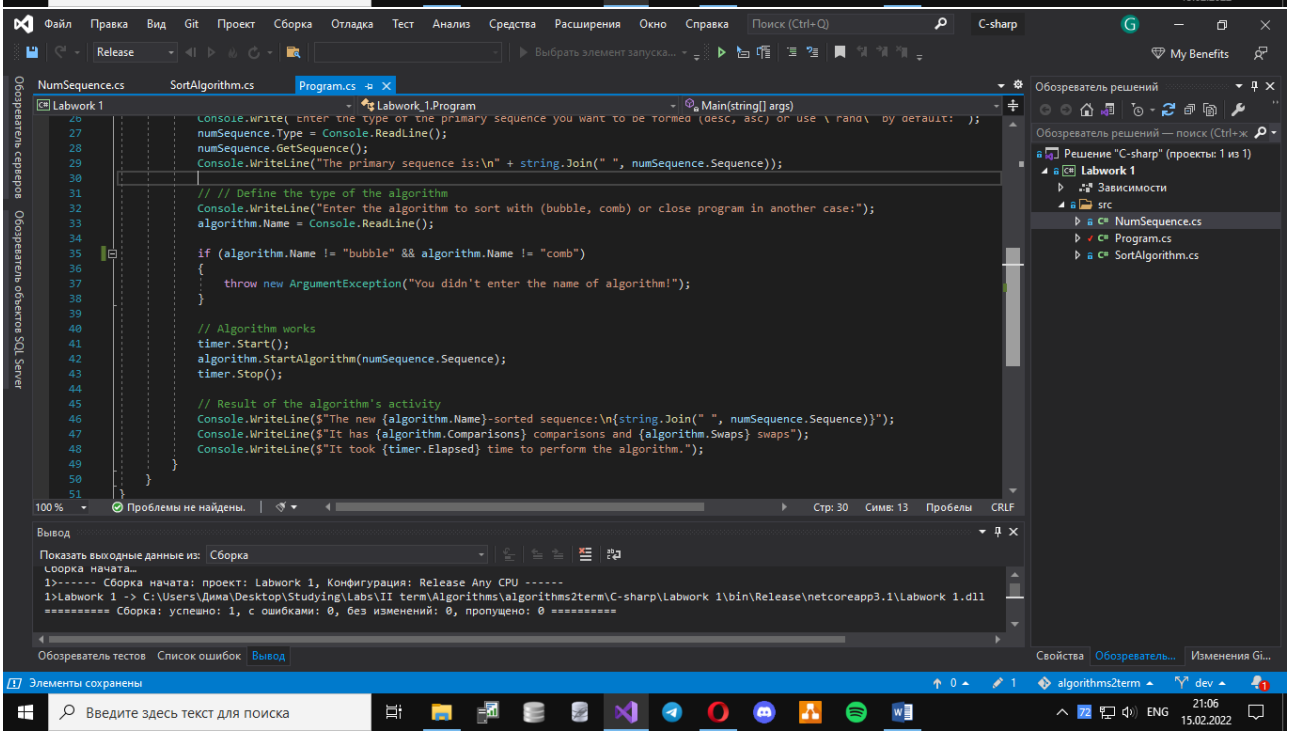
1.4.1 Вихідний код

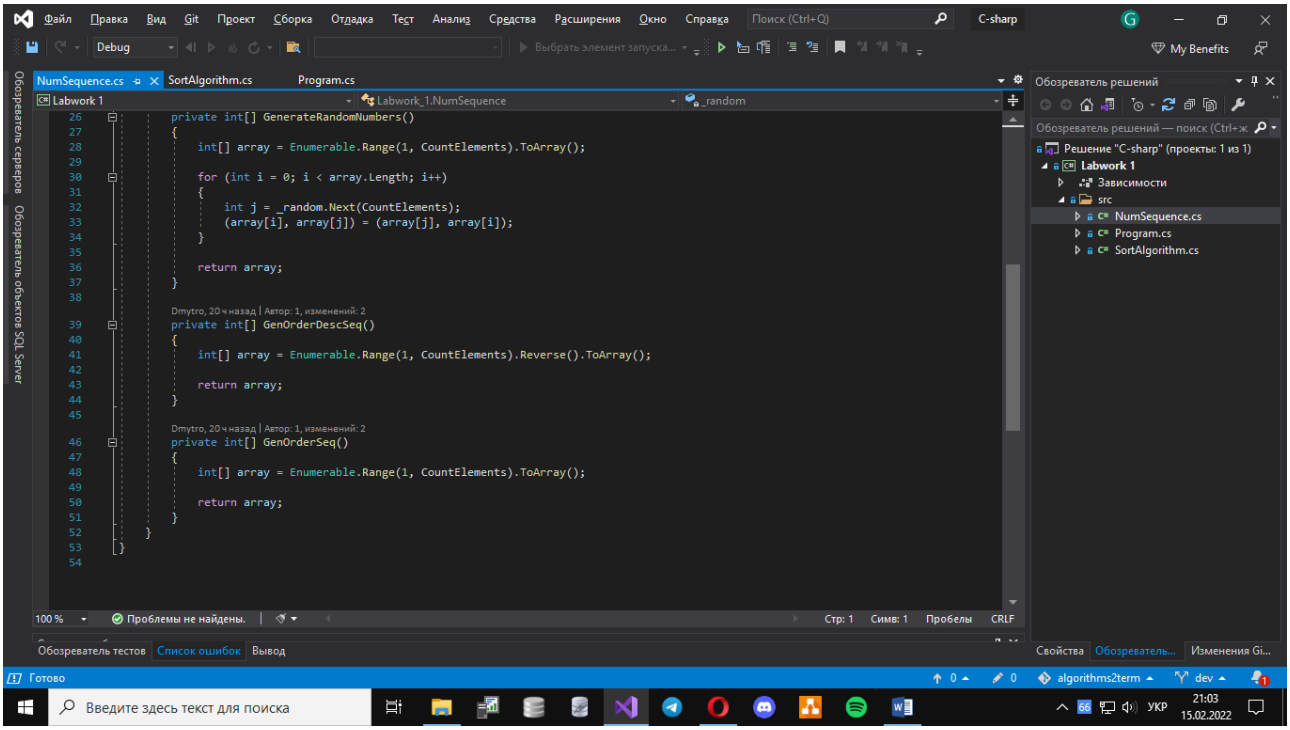



Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
C-sharp
Debug
My Benefits
SortAlgorithm.cs
Program.cs
Labwork 1
Labwork_1.SortAlgorithm
StartAlgorithm(int[] Sequence)
Dmytro, 20 ч назад | Автор: 1, изменений: 3
private void CombSort(int[] array)
{
double factor = 1.2473309;
int step = array.Length - 1;
while (step >= 1)
{
for (int i = 0; i + step < array.Length; i++)
{
if (array[i] > array[i + step])
{
(array[i], array[i + step]) = (array[i + step], array[i]);
Swaps++;
}
Comparisons++;
}
step = (int)(step / factor);
}
}
}
}
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "C-sharp" (проекты: 1 из 1)
Labwork 1
Зависимости
src
NumSequence.cs
Program.cs
SortAlgorithm.cs
100 %
Проблемы не найдены.
Стр: 12
Симв: 9
Пробелы
CRLF
Обозреватель тестов
Список ошибок
Вывод
Свойства
Обозреватель...
Изменения Gi...
Готово
algorithms2term
dev
Введите здесь текст для поиска
УКР
21:02
15.02.2022

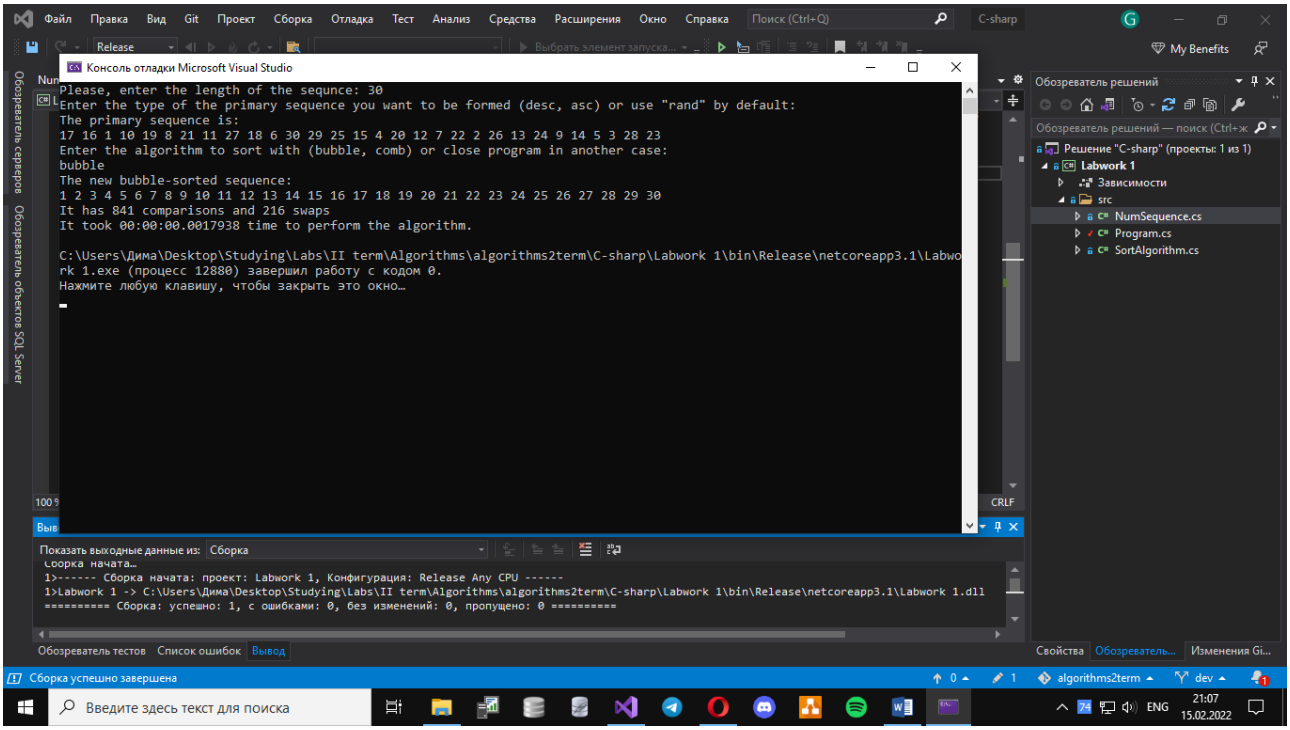
Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
C-sharp
Debug
My Benefits
SortAlgorithm.cs
Program.cs
Labwork 1
Labwork_1.Program
Main(string[] args)
using System;
using System.Diagnostics;
namespace Labwork_1
{
Dmytro, 20 ч назад | Автор: 1, изменений: 4
static public class Program
{
Dmytro, 20 ч назад | Автор: 1, изменений: 4
static void Main(string[] args)
{
NumSequence numSequence = new NumSequence();
SortAlgorithm algorithm = new SortAlgorithm();
Stopwatch timer = new Stopwatch();
// Define the length of the sequence
Console.Write("Please, enter the length of the sequnce: ");
if (int.TryParse(Console.ReadLine(), out int countElements))
{
numSequence.CountElements = countElements;
}
else
{
throw new ArgumentException("You enetered not a number!");
}
// Define the type of the sequence
Console.Write("Enter the type of the primary sequence you want to be formed (desc, asc) or use \"rand\" by default: ");
numSequence.Type = Console.ReadLine();
numSequence.GetSequence();
Console.WriteLine("The primary sequence is:\n" + string.Join(" ", numSequence.Sequence));
// // Define the type of the algorithm
Обозреватель решений
Обозреватель решений — поиск (Ctrl+ж)
Решение "C-sharp" (проекты: 1 из 1)
Labwork 1
Зависимости
src
NumSequence.cs
Program.cs
SortAlgorithm.cs
100 %
Проблемы не найдены.
Стр: 1
Симв: 1
Пробелы
CRLF
Обозреватель тестов
Список ошибок
Вывод
Свойства
Обозреватель...
Изменения Gi...
Готово
algorithms2term
dev
Введите здесь текст для поиска
УКР
21:02
15.02.2022

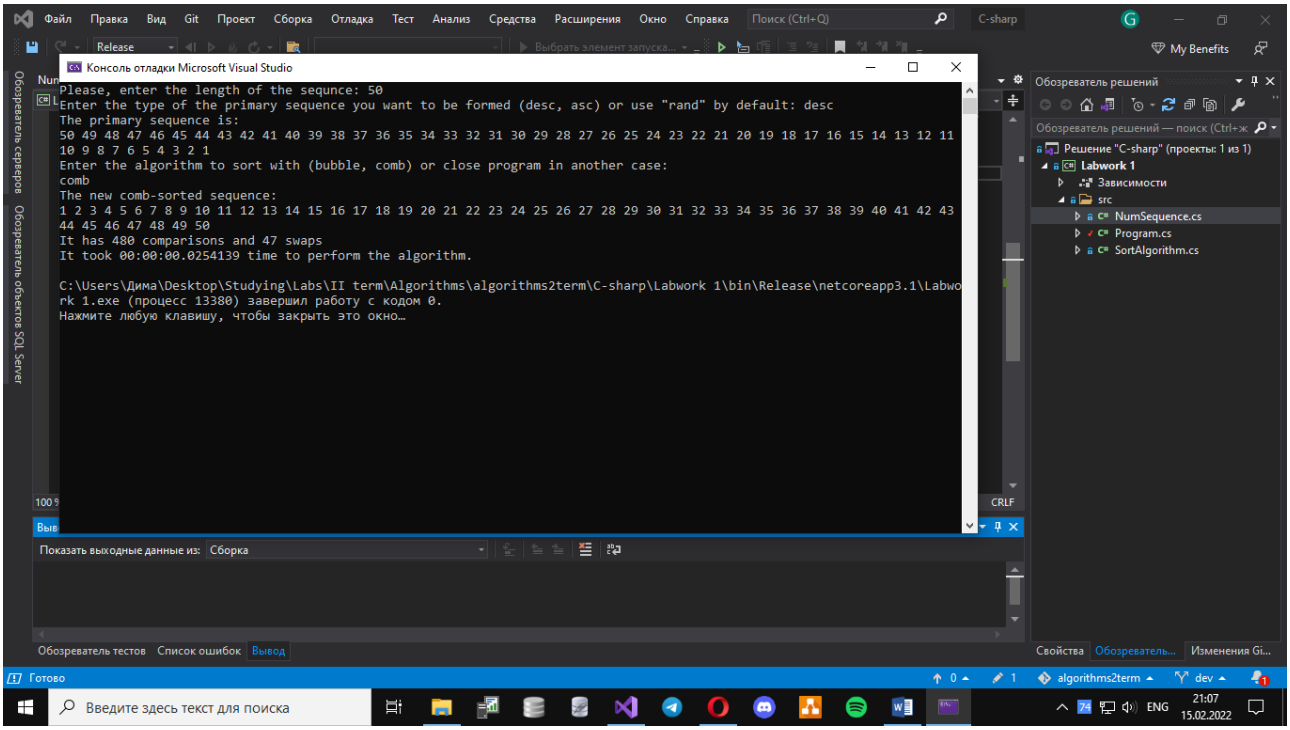

Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
C-sharp
Debug
My Benefits
NumSequence.cs SortAlgorithm.cs Program.cs
Labwork 1
Labwork_1.NumSequence
_random
using System;
using System.Linq;
namespace Labwork_1
{
public class NumSequence
{
private readonly Random _random = new Random();
public string Type { get; set; }
public int[] Sequence { get; set; }
public int CountElements { get; set; }
public void GetSequence()
{
Sequence = Type.ToLower() switch
{
"desc" => GenOrderDescSeq(),
"asc" => GenOrderSeq(),
_ => GenerateRandomNumbers(),
};
}
private int[] GenerateRandomNumbers()
{
int[] array = Enumerable.Range(1, CountElements).ToArray();
Обозреватель решений
Решение "C-sharp" (проекты: 1 из 1)
Labwork 1
Зависимости
src
NumSequence.cs
Program.cs
SortAlgorithm.cs
Проблемы не найдены.
Стр: 1 Симв: 1 Пробелы CRLF
Обозреватель тестов Список ошибок... Вывод
Свойства Обозреватель... Изменения Gi...
Готово
algorithms2term dev
Введите здесь текст для поиска
УКР
21:03
15.02.2022


Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
C-sharp
Release
My Benefits
NumSequence.cs SortAlgorithm.cs Program.cs
Labwork 1
Labwork_1.Program
Main(string[] args)
Console.WriteLine("The primary sequence is:\n" + string.Join(" ", numSequence.Sequence));
// // Define the type of the algorithm
Console.WriteLine("Enter the algorithm to sort with (bubble, comb) or close program in another case:");
algorithm.Name = Console.ReadLine();
if (algorithm.Name != "bubble" && algorithm.Name != "comb")
{
throw new ArgumentException("You didn't enter the name of algorithm!");
}
// Algorithm works
timer.Start();
algorithm.StartAlgorithm(numSequence.Sequence);
timer.Stop();
// Result of the algorithm's activity
Console.WriteLine($"The new {algorithm.Name}-sorted sequence:\n{string.Join(" ", numSequence.Sequence)}");
Console.WriteLine($"It has {algorithm.Comparisons} comparisons and {algorithm.Swaps} swaps");
Console.WriteLine($"It took {timer.Elapsed} time to perform the algorithm.");
Обозреватель решений
Решение "C-sharp" (проекты: 1 из 1)
Labwork 1
Зависимости
src
NumSequence.cs
Program.cs
SortAlgorithm.cs
Проблемы не найдены.
Стр: 30 Симв: 13 Пробелы CRLF
Вывод
Показать выходные данные из: Сборка
Сборка начата...
1>------ Сборка начата: проект: Labwork 1, Конфигурация: Release Any CPU ------
1>Labwork 1 -> C:\Users\Дима\Desktop\Studying\Labs\II term\Algorithms\algorithms2term\C-sharp\Labwork 1\bin\Release\netcoreapp3.1\Labwork 1.dll
========== Сборка: успешно: 1, с ошибками: 0, без изменений: 0, пропущено: 0 ==========
Обозреватель тестов Список ошибок Вывод
Свойства Обозреватель... Изменения Gi...
Элементы сохранены
algorithms2term dev
Введите здесь текст для поиска
ENG
21:06
15.02.2022



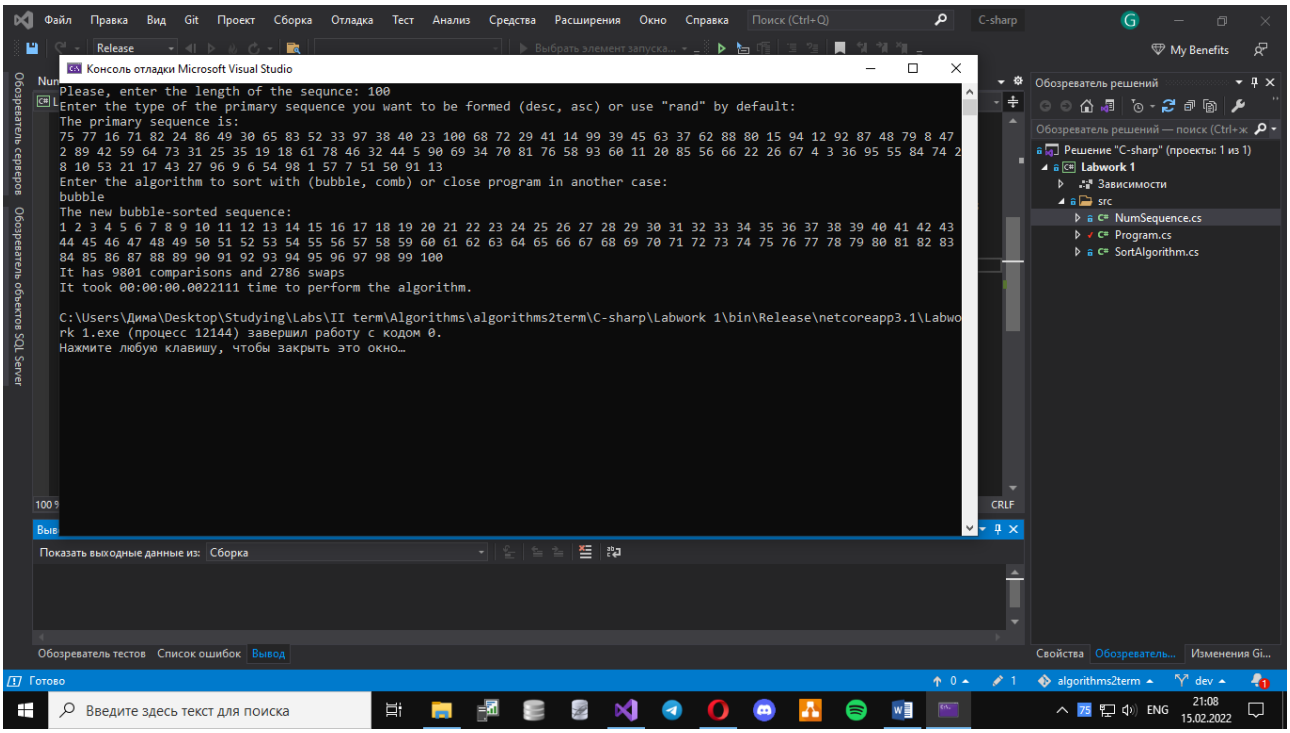
1.4.2 Приклад роботи





На рисунках 3.1 і 3.2 показані приклади роботи програми сортування масивів на 100 і 1000 елементів відповідно.

Рисунок 3.1 – Сортування масиву на 100 елементів «Бульбашка»



«Гребінець»

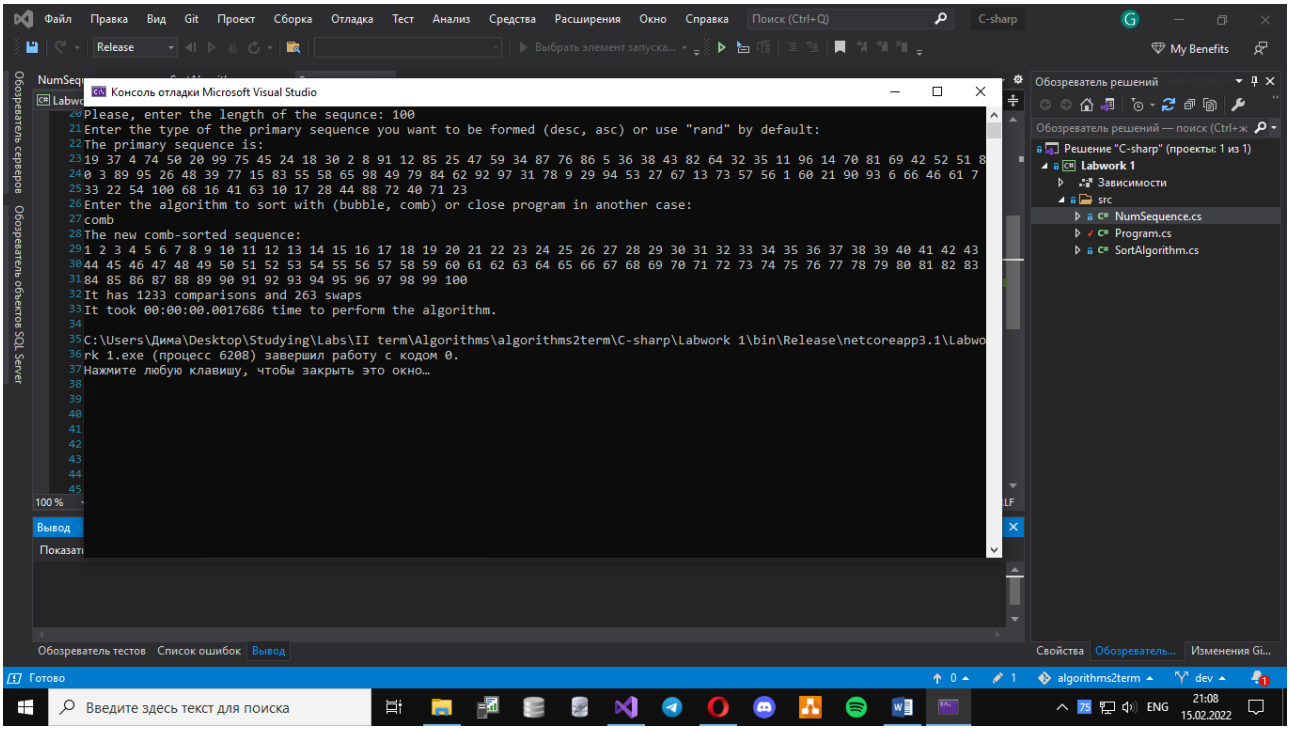
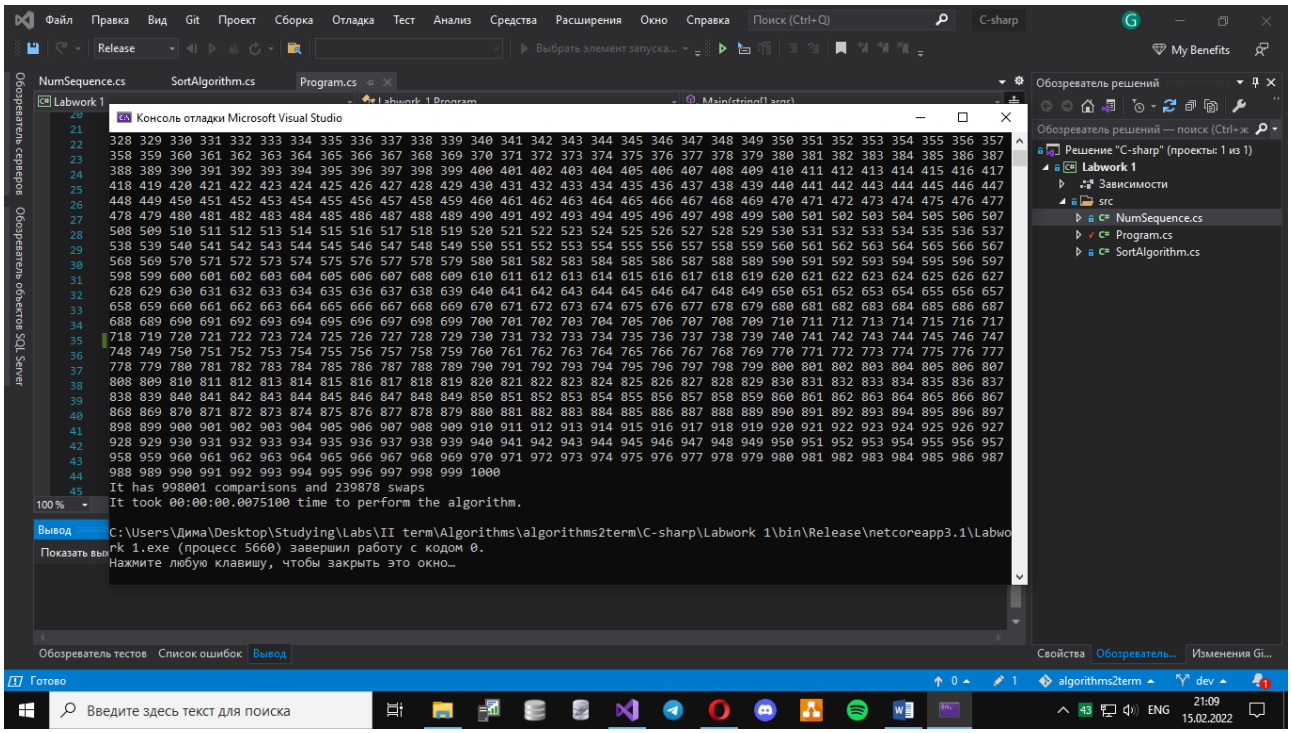
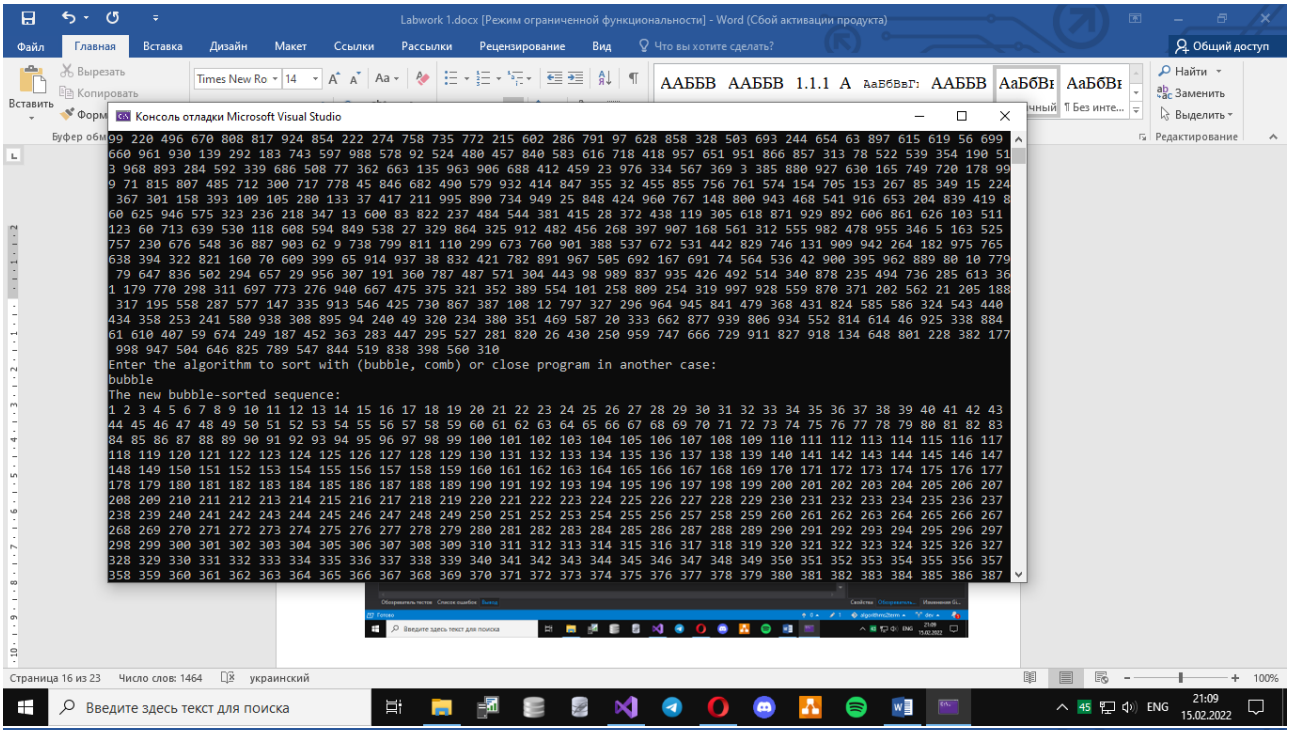
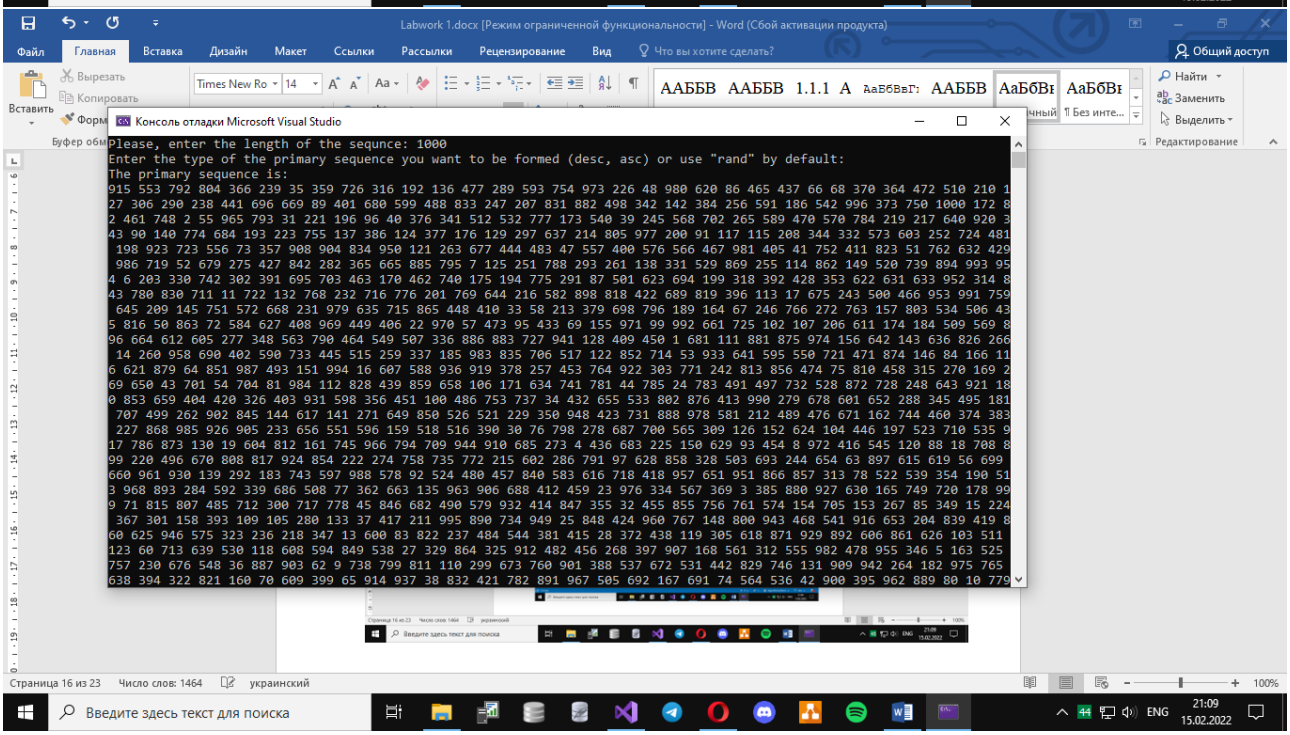


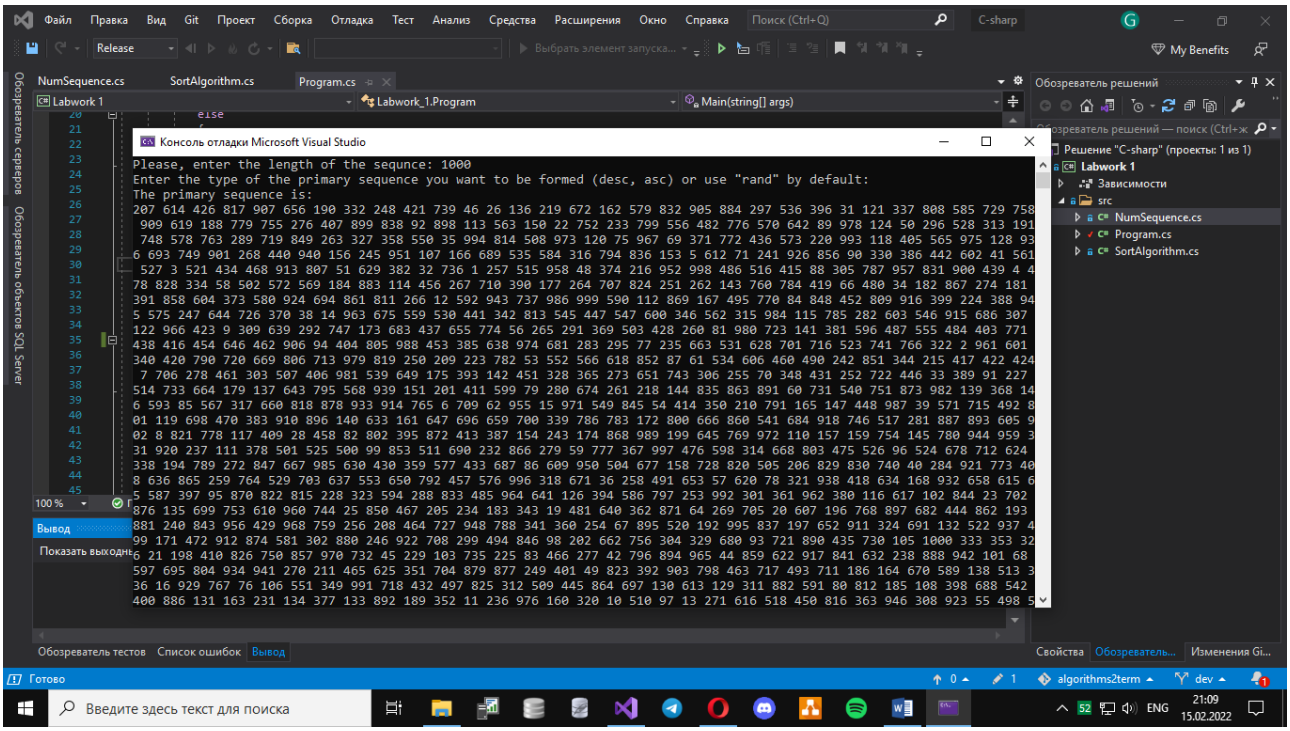
Рисунок 3.2 – Сортування масиву на 1000 елементів «Бульбашка»

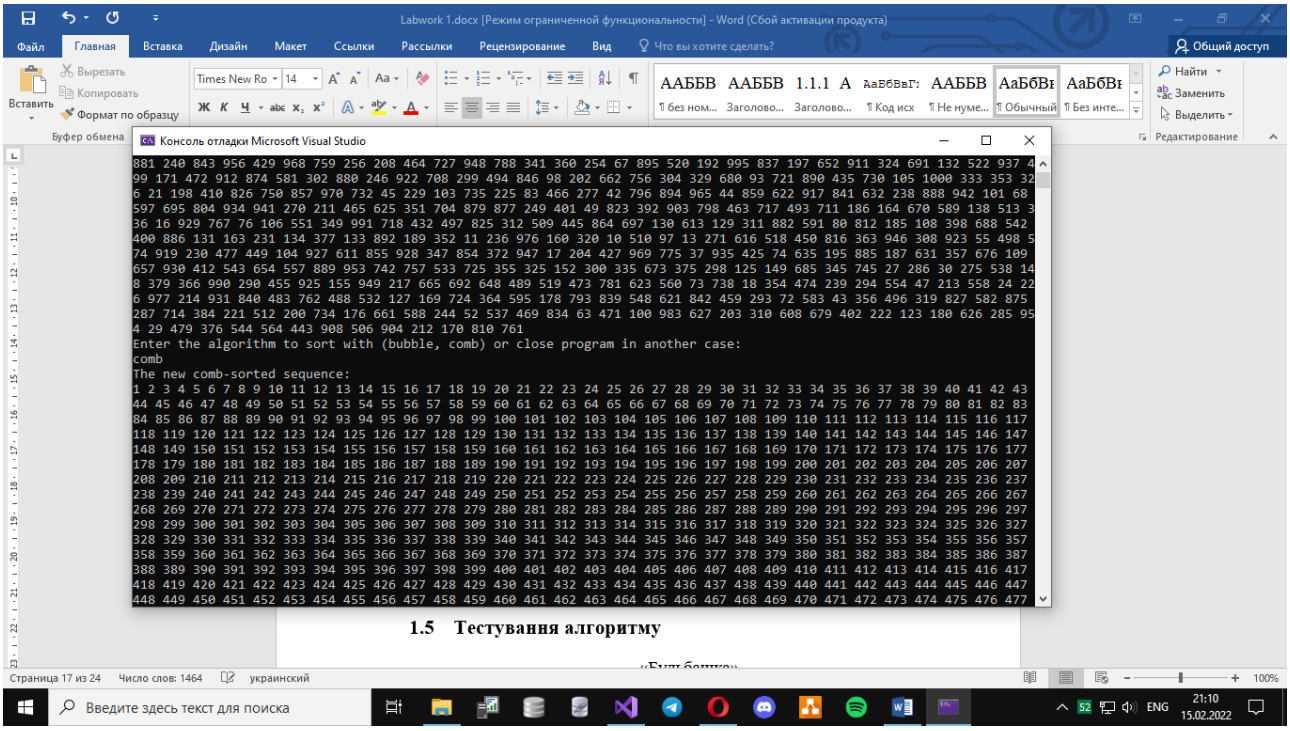


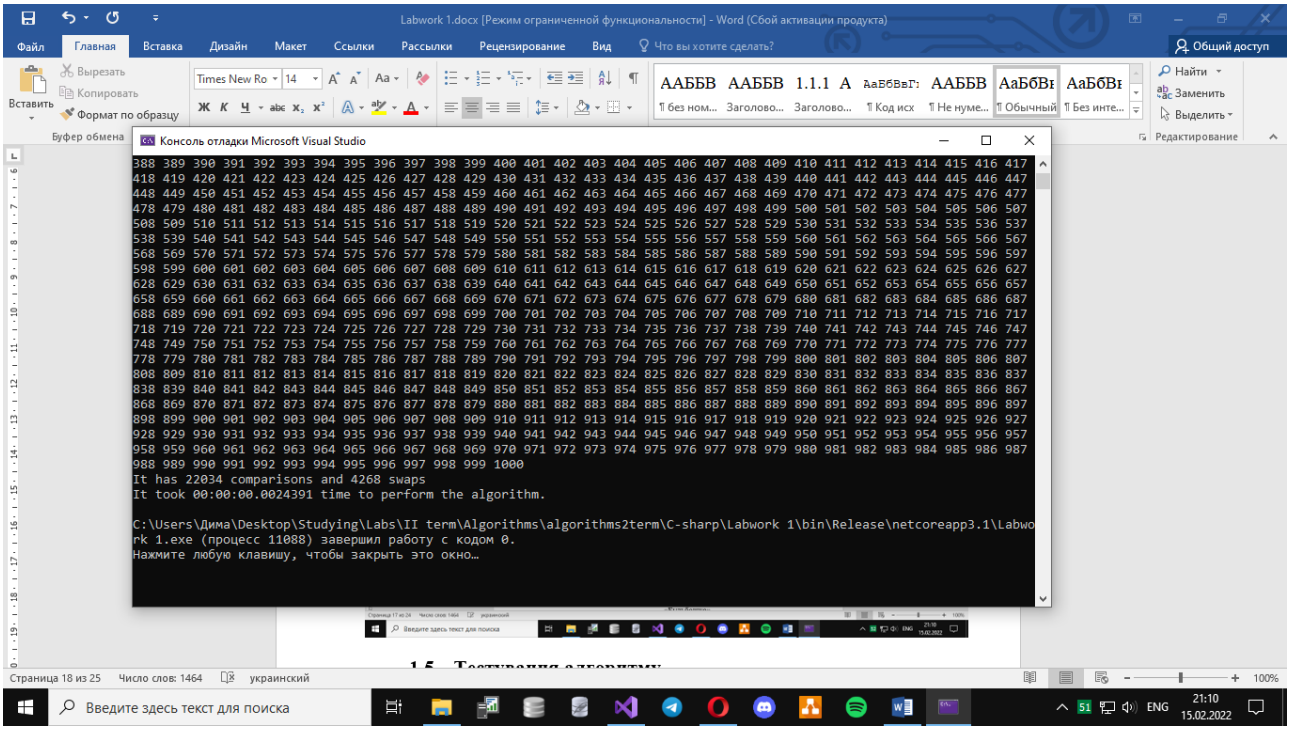

Labwork 1.docx [Режим ограниченной функциональности] - Word (Сбой активации продукта)
Консоль отладки Microsoft Visual Studio
99 220 496 670 808 817 924 854 222 274 758 735 772 215 602 286 791 97 628 858 328 503 693 244 654 63 897 615 619 56 699
660 961 930 139 292 183 743 597 988 578 92 524 480 457 840 583 616 718 418 957 651 951 866 857 313 78 522 539 354 190 51
3 968 893 284 592 339 686 508 77 362 663 135 963 906 688 412 459 23 976 334 567 369 3 385 880 927 630 165 749 720 178 99
9 71 815 807 485 712 300 717 778 45 846 682 490 579 932 414 847 355 32 455 855 756 761 574 154 705 153 267 85 349 15 224
367 301 158 393 109 105 280 133 37 417 211 995 890 734 949 25 848 424 960 767 148 800 943 468 541 916 653 204 839 419 8
60 625 946 575 323 236 218 347 13 600 83 822 237 484 544 381 415 28 372 438 119 305 618 871 929 892 606 861 626 103 511
123 60 713 639 530 118 608 594 849 538 27 329 864 325 912 482 456 268 397 907 168 561 312 555 982 478 955 346 5 163 525
757 230 676 548 36 887 903 62 9 738 799 811 110 299 673 760 901 388 537 672 531 442 829 746 131 909 942 264 182 975 765
638 394 322 821 160 70 609 399 65 914 937 38 832 421 782 891 967 505 692 167 691 74 564 536 42 900 395 962 889 80 10 779
79 647 836 502 294 657 29 956 307 191 360 787 487 571 304 443 98 989 837 935 426 492 514 340 878 235 494 736 285 613 36
1 179 770 298 311 697 773 276 940 667 475 375 321 352 389 554 101 258 809 254 319 997 928 559 870 371 202 562 21 205 188
317 195 558 287 577 147 335 913 546 425 730 867 387 108 12 797 327 296 964 945 841 479 368 431 824 585 586 324 543 440
434 358 253 241 580 938 308 895 94 240 49 320 234 380 351 469 587 20 333 662 877 939 806 934 552 814 614 46 925 338 884
61 610 407 59 674 249 187 452 363 283 447 295 527 281 820 26 430 250 959 747 666 729 911 827 918 134 648 801 228 382 177
998 947 504 646 825 789 547 844 519 838 398 560 310
Enter the algorithm to sort with (bubble, comb) or close program in another case:
bubble
The new bubble-sorted sequence:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117
118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177
178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237
238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267
268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297
298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327
328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357
358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387
Страница 16 из 23
Число слов: 1464
украинский
Введите здесь текст для поиска
ENG
21:09
15.02.2022


Labwork 1.docx [Режим ограниченной функциональности] - Word (Сбой активации продукта)
Консоль отладки Microsoft Visual Studio
Please, enter the length of the sequnce: 1000
Enter the type of the primary sequence you want to be formed (desc, asc) or use "rand" by default:
The primary sequence is:
915 553 792 804 366 239 35 359 726 316 192 136 477 289 593 754 973 226 48 980 620 86 465 437 66 68 370 364 472 510 210 1
27 306 290 238 441 696 669 89 401 680 599 488 833 247 207 831 882 498 342 142 384 256 591 186 542 996 373 750 1000 172 8
2 461 748 2 55 965 793 31 221 196 96 40 376 341 512 532 777 173 540 39 245 568 702 265 589 470 570 784 219 217 640 920 3
43 90 140 774 684 193 223 755 137 386 124 377 176 129 297 637 214 805 977 200 91 117 115 208 344 332 573 603 252 724 481
198 923 723 556 73 357 908 904 834 950 121 263 677 444 483 47 557 400 576 566 467 981 405 41 752 411 823 51 762 632 429
986 719 52 679 275 427 842 282 365 665 885 795 7 125 251 788 293 261 138 331 529 869 255 114 862 149 520 739 894 993 95
4 6 203 330 742 302 391 695 703 463 170 462 740 175 194 775 291 87 501 623 694 199 318 392 428 353 622 631 633 952 314 8
43 780 830 711 11 722 132 768 232 716 776 201 769 644 216 582 898 818 422 689 819 396 113 17 675 243 500 466 953 991 759
645 209 145 751 572 668 231 979 635 715 865 448 410 33 58 213 379 698 796 189 164 67 246 766 272 763 157 803 534 506 43
5 816 50 863 72 584 627 408 969 449 406 22 970 57 473 95 433 69 155 971 99 992 661 725 102 107 206 611 174 184 509 569 8
96 664 612 605 277 348 563 790 464 549 507 336 886 883 727 941 128 409 450 1 681 111 881 875 974 156 642 143 636 826 266
14 260 958 690 402 590 733 445 515 259 337 185 983 835 706 517 122 852 714 53 933 641 595 550 721 471 874 146 84 166 11
6 621 879 64 851 987 493 151 994 16 607 588 936 919 378 257 453 764 922 303 771 242 813 856 474 75 810 458 315 270 169 2
69 650 43 701 54 704 81 984 112 828 439 859 658 106 171 634 741 781 44 785 24 783 491 497 732 528 872 728 248 643 921 18
0 853 659 404 420 326 403 931 598 356 451 100 486 753 737 34 432 655 533 802 876 413 990 279 678 601 652 288 345 495 181
707 499 262 902 845 144 617 141 271 649 850 526 521 229 350 948 423 731 888 978 581 212 489 476 671 162 744 460 374 383
227 868 985 926 905 233 656 551 596 159 518 516 390 30 76 798 278 687 700 565 309 126 152 624 104 446 197 523 710 535 9
17 786 873 130 19 604 812 161 745 966 794 709 944 910 685 273 4 436 683 225 150 629 93 454 8 972 416 545 120 88 18 708 8
99 220 496 670 808 817 924 854 222 274 758 735 772 215 602 286 791 97 628 858 328 503 693 244 654 63 897 615 619 56 699
660 961 930 139 292 183 743 597 988 578 92 524 480 457 840 583 616 718 418 957 651 951 866 857 313 78 522 539 354 190 51
3 968 893 284 592 339 686 508 77 362 663 135 963 906 688 412 459 23 976 334 567 369 3 385 880 927 630 165 749 720 178 99
9 71 815 807 485 712 300 717 778 45 846 682 490 579 932 414 847 355 32 455 855 756 761 574 154 705 153 267 85 349 15 224
367 301 158 393 109 105 280 133 37 417 211 995 890 734 949 25 848 424 960 767 148 800 943 468 541 916 653 204 839 419 8
60 625 946 575 323 236 218 347 13 600 83 822 237 484 544 381 415 28 372 438 119 305 618 871 929 892 606 861 626 103 511
123 60 713 639 530 118 608 594 849 538 27 329 864 325 912 482 456 268 397 907 168 561 312 555 982 478 955 346 5 163 525
757 230 676 548 36 887 903 62 9 738 799 811 110 299 673 760 901 388 537 672 531 442 829 746 131 909 942 264 182 975 765
638 394 322 821 160 70 609 399 65 914 937 38 832 421 782 891 967 505 692 167 691 74 564 536 42 900 395 962 889 80 10 779
Страница 16 из 23
Число слов: 1464
украинский
Введите здесь текст для поиска
ENG
21:09
15.02.2022

«Гребінець»


Консоль отладки Microsoft Visual Studio
Please, enter the length of the sequnce: 1000
Enter the type of the primary sequence you want to be formed (desc, asc) or use "rand" by default:
The primary sequence is:
207 614 426 817 907 656 190 332 248 421 739 46 26 136 219 672 162 579 832 905 884 297 536 396 31 121 337 808 585 729 758
909 619 188 779 755 276 407 899 838 92 898 113 563 150 22 752 233 799 556 482 776 570 642 89 978 124 50 296 528 313 191
748 578 763 289 719 849 263 327 358 550 35 994 814 508 973 120 75 967 69 371 772 436 573 220 993 118 405 565 975 128 93
6 693 749 901 268 440 940 156 245 951 107 166 689 535 584 316 794 836 153 5 612 71 241 926 856 90 330 386 442 602 41 561
527 3 521 434 468 913 807 51 629 382 32 736 1 257 515 958 48 374 216 952 998 486 516 415 88 305 787 957 831 900 439 4 4
78 828 334 58 502 572 569 184 883 114 456 267 710 390 177 264 707 824 251 262 143 760 784 419 66 480 34 182 867 274 181
391 858 604 373 580 924 694 861 811 266 12 592 943 737 986 999 590 112 869 167 495 770 84 848 452 809 916 399 224 388 94
5 575 247 644 726 370 38 14 963 675 559 530 441 342 813 545 447 547 600 346 562 315 984 115 785 282 603 546 915 686 307
122 966 423 9 309 639 292 747 173 683 437 655 774 56 265 291 369 503 428 260 81 980 723 141 381 596 487 555 484 403 771
438 416 454 646 462 906 94 404 805 988 453 385 638 974 681 283 295 77 235 663 531 628 701 716 523 741 766 322 2 961 601
340 420 790 720 669 806 713 979 819 250 209 223 782 53 552 566 618 852 87 61 534 606 460 490 242 851 344 215 417 422 424
7 706 278 461 303 507 406 981 539 649 175 393 142 451 328 365 273 651 743 306 255 70 348 431 252 722 446 33 389 91 227
514 733 664 179 137 643 795 568 939 151 201 411 599 79 280 674 261 218 144 835 863 891 60 731 540 751 873 982 139 368 14
6 593 85 567 317 660 818 878 933 914 765 6 709 62 955 15 971 549 845 54 414 350 210 791 165 147 448 987 39 571 715 492 8
01 119 698 470 383 910 896 140 633 161 647 696 659 700 339 786 783 172 800 666 860 541 684 918 746 517 281 887 893 605 9
02 8 821 778 117 409 28 458 82 802 395 872 413 387 154 243 174 868 989 199 645 769 972 110 157 159 754 145 780 944 959 3
31 920 237 111 378 501 525 500 99 853 511 690 232 866 279 59 777 367 997 476 598 314 668 803 475 526 96 524 678 712 624
338 194 789 272 847 667 985 630 430 359 577 433 687 86 609 950 504 677 158 728 820 505 206 829 830 740 40 284 921 773 40
8 636 865 259 764 529 703 637 553 650 792 457 576 996 318 671 36 258 491 653 57 620 78 321 938 418 634 168 932 658 615 6
5 587 397 95 870 822 815 228 323 594 288 833 485 964 641 126 394 586 797 253 992 301 361 962 380 116 617 102 844 23 702
876 135 699 753 610 960 744 25 850 467 205 234 183 343 19 481 640 362 871 64 269 705 20 607 196 768 897 682 444 862 193
881 240 843 956 429 968 759 256 208 464 727 948 788 341 360 254 67 895 520 192 995 837 197 652 911 324 691 132 522 937 4
99 171 472 912 874 581 302 880 246 922 708 299 494 846 98 202 662 756 304 329 680 93 721 890 435 730 105 1000 333 353 32
6 21 198 410 826 750 857 970 732 45 229 103 735 225 83 466 277 42 796 894 965 44 859 622 917 841 632 238 888 942 101 68
597 695 804 934 941 270 211 465 625 351 704 879 877 249 401 49 823 392 903 798 463 717 493 711 186 164 670 589 138 513 3
36 16 929 767 76 106 551 349 991 718 432 497 825 312 509 445 864 697 130 613 129 311 882 591 80 812 185 108 398 688 542
400 886 131 163 231 134 377 133 892 189 352 11 236 976 160 320 10 510 97 13 271 616 518 450 816 363 946 308 923 55 498 5
Вывод
Показать выходные
Обозреватель тестов
Список ошибок
Вывод
Свойства
Обозреватель...
Изменения Gi...
Готово
algorithms2term
dev
Введите здесь текст для поиска
ENG
21:09
15.02.2022


Labwork 1.docx [Режим ограниченной функциональности] - Word (Сбой активации продукта)
Консоль отладки Microsoft Visual Studio
881 240 843 956 429 968 759 256 208 464 727 948 788 341 360 254 67 895 520 192 995 837 197 652 911 324 691 132 522 937 4
99 171 472 912 874 581 302 880 246 922 708 299 494 846 98 202 662 756 304 329 680 93 721 890 435 730 105 1000 333 353 32
6 21 198 410 826 750 857 970 732 45 229 103 735 225 83 466 277 42 796 894 965 44 859 622 917 841 632 238 888 942 101 68
597 695 804 934 941 270 211 465 625 351 704 879 877 249 401 49 823 392 903 798 463 717 493 711 186 164 670 589 138 513 3
36 16 929 767 76 106 551 349 991 718 432 497 825 312 509 445 864 697 130 613 129 311 882 591 80 812 185 108 398 688 542
400 886 131 163 231 134 377 133 892 189 352 11 236 976 160 320 10 510 97 13 271 616 518 450 816 363 946 308 923 55 498 5
74 919 230 477 449 104 927 611 855 928 347 854 372 947 17 204 427 969 775 37 935 425 74 635 195 885 187 631 357 676 109
657 930 412 543 654 557 889 953 742 757 533 725 355 325 152 300 335 673 375 298 125 149 685 345 745 27 286 30 275 538 14
8 379 366 990 290 455 925 155 949 217 665 692 648 489 519 473 781 623 560 73 738 18 354 474 239 294 554 47 213 558 24 22
6 977 214 931 840 483 762 488 532 127 169 724 364 595 178 793 839 548 621 842 459 293 72 583 43 356 496 319 827 582 875
287 714 384 221 512 200 734 176 661 588 244 52 537 469 834 63 471 100 983 627 203 310 608 679 402 222 123 180 626 285 95
4 29 479 376 544 564 443 908 506 904 212 170 810 761
Enter the algorithm to sort with (bubble, comb) or close program in another case:
comb
The new comb-sorted sequence:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117
118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177
178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237
238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267
268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297
298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327
328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357
358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387
388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417
418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447
448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477
1.5 Тестування алгоритму
Страница 17 из 24
Число слов: 1464
украинский
Введите здесь текст для поиска
ENG
21:10
15.02.2022



1.5 Тестування алгоритму

1.5.1 Часові характеристики оцінювання

В таблиці 3.2 наведені характеристики оцінювання числа порівнянь і числа перестановок алгоритму сортування бульбашки для масивів різної розмірності, коли масив містить упорядковану послідовність елементів.

Таблиця 3.2 – Характеристики оцінювання алгоритму сортування бульбашкою для упорядкованої послідовності елементів у масиві

Розмірність масиву	Число порівнянь	Число перестановок
10	81	0
100	9801	0
1000	998001	0
5000	24990001	0
10000	99980001	0
20000	399960001	0
50000	2499900001	0

Таблиця 3.2.2 – Характеристики оцінювання алгоритму сортування гребінцем для упорядкованої послідовності елементів у масиві

Розмірність масиву	Число порівнянь	Число перестановок
10	39	0
100	1233	0
1000	22034	0
5000	144865	0
10000	329653	0
20000	719246	0
50000	1997961	0

В таблиці 3.3 наведені характеристики оцінювання числа порівнянь і числа перестановок алгоритму сортування бульбашки для масивів різної розмірності, коли масиви містять зворотно упорядковану послідовність елементів.

Таблиця 3.3 – Характеристики оцінювання алгоритму сортування бульбашки для зворотно упорядкованої послідовності елементів у масиві.

Розмірність масиву	Число порівнянь	Число перестановок
10	81	45
100	9801	4950
1000	998001	499500
5000	24990001	12497500
10000	99980001	49995000
20000	399960001	199990000
50000	2499900001	1249975000

Таблиця 3.3.2 – Характеристики оцінювання алгоритму сортування гребінця для зворотно упорядкованої послідовності елементів у масиві.

Розмірність масиву	Число порівнянь	Число перестановок
10	39	5
100	1233	106
1000	22034	1528
5000	144865	9110
10000	329653	19164
20000	719246	40636
50000	1997961	109794

У таблиці 3.4 наведені характеристики оцінювання числа порівнянь і числа перестановок алгоритму сортування бульбашки для масивів різної розмірності, масиви містять випадкову послідовність елементів.

Таблиця 3.4 – Характеристика оцінювання алгоритму сортування бульбашкою за спаданням для випадкової послідовності елементів у масиві.

Розмірність масиву	Число порівнянь	Число перестановок
10	81	24
100	9801	2762
1000	998001	262360
5000	24990001	6393886
10000	99980001	26013701
20000	399960001	103898756
50000	2499900001	646797020

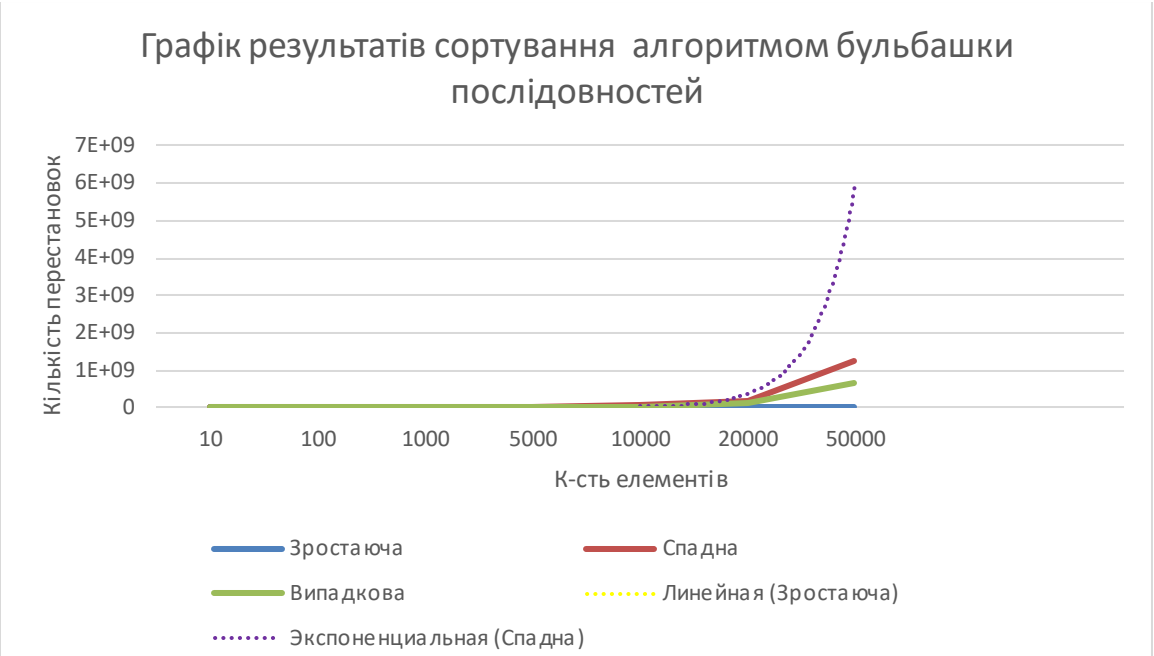
Таблиця 3.4.2 – Характеристика оцінювання алгоритму сортування гребінцем для випадкової послідовності елементів у масиві.

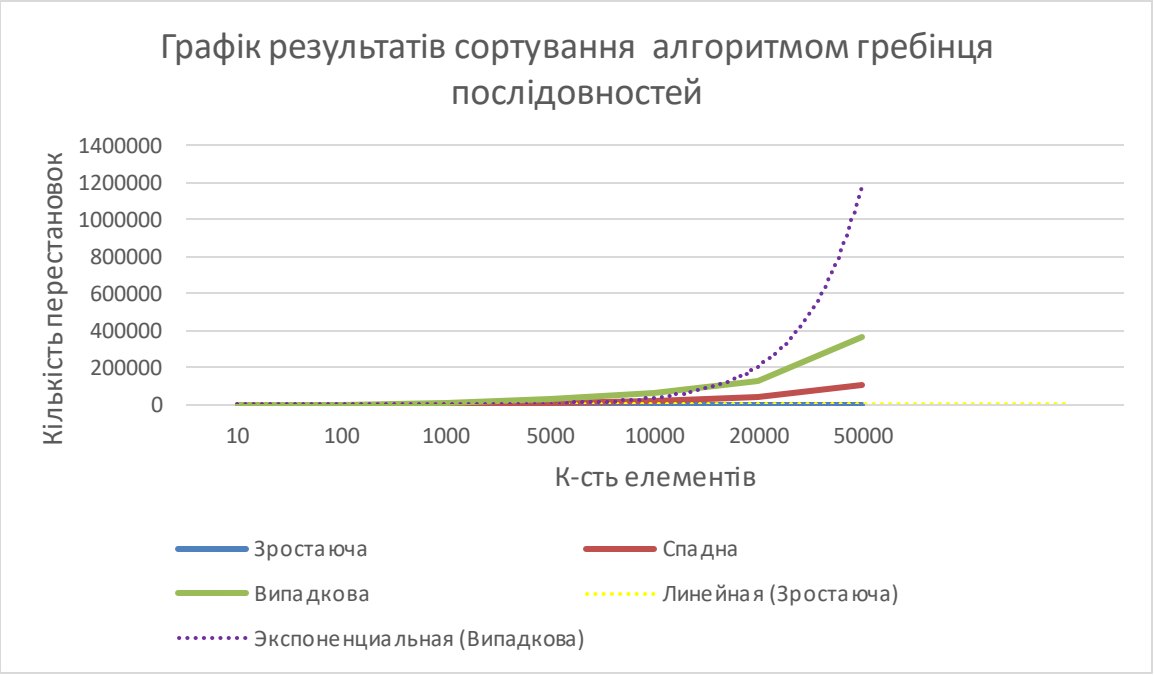
Розмірність масиву	Число порівнянь	Число перестановок
10	39	5
100	1233	219
1000	22034	4206

5000	144865	27094
10000	329653	59774
20000	719246	130433
50000	1997961	367660

1.5.2 Графіки залежності часових характеристик оцінювання від розмірності масиву

На рисунку 3.3 показані графіки залежності часових характеристик оцінювання від розмірності масиву для випадків, коли масиви містять упорядковану послідовність елементів (синій графік), коли масиви містять зворотно упорядковану послідовність елементів (червоний графік), коли масиви містять випадкову послідовність елементів (зелений графік), також показані асимптотичні оцінки гіршого (фіолетовий графік) і кращого (жовтий графік) випадків для порівняння. Рисунок 3.3 – Графіки залежності часових характеристик оцінювання





Висновок

При виконанні даної лабораторної роботи я скористався таймером, вимірюючи час виконання алгоритму, закріпив свої знання з правил використання алгоритмів та їх доречності в залежності від ситуації. Мною було проведене тестування алгоритму «бульбашки» та «гребінець» шляхом підстановки різних значень, які представляють кількість елементів послідовності.

КРИТЕРІЇ ОЦІНЮВАННЯ

У випадку здачі лабораторної роботи до 21.02.2022 включно максимальний бал дорівнює – 5. Після 21.02.2022 – 28.02.2022 максимальний бал дорівнює – 2,5. Після 28.02.2022 робота не приймається

Критерії оцінювання у відсотках від максимального балу:

- аналіз алгоритму на відповідність властивостям – 10%;
- псевдокод алгоритму – 15%;
- аналіз часової складності – 25%;
- програмна реалізація алгоритму – 25%;
- тестування алгоритму – 20%;
- висновок – 5%.