

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

**Кафедра ІІІ**

**Звіт**

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

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

**Виконав(ла)**

ІІІ-13 Шиманська Ганна Артурівна

(шифр, прізвище, ім'я, по батькові)

**Перевірив**

Сопов Олексій Олександрович

(прізвище, ім'я, по батькові)

Київ 2022

## ЗМІСТ

<b>1</b>	<b>МЕТА ЛАБОРАТОРНОЇ РОБОТИ .....</b>	<b>3</b>
<b>2</b>	<b>ЗАВДАННЯ .....</b>	<b>4</b>
<b>3</b>	<b>ВИКОНАННЯ.....</b>	<b>5</b>
3.1	АНАЛІЗ АЛГОРИТМУ НА ВІДПОВІДНІСТЬ ВЛАСТИВОСТЯМ .....	5
3.2	ПСЕВДОКОД АЛГОРИТМУ .....	5
3.3	АНАЛІЗ ЧАСОВОЇ СКЛАДНОСТІ.....	6
3.4	ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ .....	6
3.4.1	<i>Вихідний код.....</i>	<i>7</i>
3.4.2	<i>Приклад роботи .....</i>	<i>8</i>
3.5	ТЕСТУВАННЯ АЛГОРИТМУ .....	12
3.5.1	<i>Часові характеристики оцінювання.....</i>	<i>12</i>
3.5.2	<i>Графіки залежності часових характеристик оцінювання від розмірності масиву .....</i>	<i>15</i>
	<b>ВИСНОВОК .....</b>	<b>17</b>

## 1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

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

## 2 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»)

### 3 ВИКОНАННЯ

#### 3.1 Аналіз алгоритму на відповідність властивостям

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

Таблиця 3.1 – Аналіз алгоритму на відповідність властивостям

Властивість	Сортування бульбашкою	Сортування гребінцем
Стійкість	Стійкий	Стійкий
«Природність» поведінки (Adaptability)	Відсутня	Відсутня
Базуються на порівняннях	Так	Так
Необхідність в додатковій пам'яті (об'єм)	-	-
Необхідність в знаннях про структури даних	Наявна	Наявна

#### 3.2 Псевдокод алгоритму

**Функція Swap(int[] array, int step, int upperBound)**

для j від 0 до array.Length – i повторити

якщо array[j + step] < array[j]

то

(array[j], array[j + step]) = (array[j + step], array[j]);

все якщо

все повторити

все функція

##### 1) Сортування бульбашкою

**Функція DoSort(int[] array)**

для i від 1 до array.Length повторити

Swap(array, 1, i);

**все повторити**  
**все функція**

## 2) Сортування гребінцем

**Функція GetUpdatedGap(int gap)**

gap = (int)(gap / 1.2473)

**якщо** gap < 1

**то**

gap = 1

**все якщо**

**повернути** gap

**Функція DoSort(int[] array)**

gap = array.Length

**поки** gap != 1

gap = GetUpdatedGap(gap)

Swap(array, gap, gap);

**все поки**

**все функція**

## 3.3 Аналіз часової складності

	<b>Сортування бульбашкою</b>	<b>Сортування гребінцем</b>
найгірший випадок	$O(n^2)$	$O(n^2)$
середній випадок	$\Theta(n^2)$	$\Theta(n^2)$
найкращий випадок	$\Omega(n^2)$	$\Omega(n \log(n))$

## 3.4 Програмна реалізація алгоритму

### 3.4.1 Вихідний код

```
using System;
using System.Diagnostics;

namespace ArrayOperations
{
    public class ArrayGenerator
    {
        public int[] GetArray(int length, bool? sorted = null)
        {
            int[] array = new int[length];
            Random rand = new Random();
            for (int i = 0; i < length; i++)
            {
                if (sorted is null)
                    array[i] = rand.Next(-5000, 10001);
                else if (sorted is true)
                {
                    array[i] = i;
                }
                else
                {
                    array[i] = length - i;
                }
            }

            return array;
        }
    }

    public class BubbleSort
    {
        private int _comparisonAmount;
        private int _swapAmount;

        protected void Swap(int[] array, int step, int upperBound)
        {
            for (int j = 0; j < array.Length - upperBound; j++)
            {
                _comparisonAmount++;
                if (array[j + step] < array[j])
                {
                    (array[j], array[j + step]) = (array[j + step], array[j]);
                    _swapAmount++;
                }
            }
        }

        public void DisplayCharacteristics()
        {
            Console.WriteLine("The amount of comparisons: " +
                _comparisonAmount);
            Console.WriteLine("The amount of swaps: " + _swapAmount);
        }

        public void DisplayArray(int[] array)
        {
            Console.WriteLine($"\\n{this.GetType().Name}");
            for (int i = 0; i < array.Length; i++)
            {
                Console.Write("{0,7}", array[i]);
            }
            Console.WriteLine();
        }
    }
}
```

```

    }

    public double GetSortingTime(int[] array)
    {
        Stopwatch sw = Stopwatch.StartNew();
        DoSort(array);
        sw.Stop();
        return sw.Elapsed.TotalMilliseconds;
    }

    protected virtual void DoSort(int[] array)
    {
        for (int i = 1; i <= array.Length - 1; i++)
        {
            Swap(array, 1, i);
        }
    }
}

public class CombSort : BubbleSort
{
    private int GetUpdatedGap(int gap)
    {
        gap = (int) (gap / 1.2473);
        if (gap < 1) gap = 1;
        return gap;
    }

    protected override void DoSort(int[] array)
    {
        int gap = array.Length;
        while (gap != 1)
        {
            gap = GetUpdatedGap(gap);
            Swap(array, gap, gap);
        }
    }
}

```

### 3.4.2 Приклад роботи

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



```
Enter the length of array to generate:
100

Initial array is:

BubbleSort
694 9532 -1175 6156 -1136 8346 8694 6496 6307 4849 3597 4238 8486 847 3002 -3246 -3615 581 -910 6751 9098 -286 7922 4469 3633 -4299
8921 -4536 -2376 -148 5965 -2301 1385 -4097 -3589 4487 101 -2667 4797 -263 -554 -1892 523 1106 1248 -1141 -2298 5711 7783 9599 2301 9977 4
338 5248 9269 -4021 -3391 4357 -2478 -1838 9426 8230 3872 1454 8884 898 -2147 7583 -565 8516 6654 9052 5784 9354 2304 5989 9191 1816 -177
9 -226 8000 -1791 -2319 -1974 7552 5989 3361 6137 -3365 -4568 2877 5745 9811 1635 312 1217 4236 9070 2222 -4651

Execution time in ms 0,7084

Sorted array is

BubbleSort
-4651 -4568 -4536 -4299 -4097 -4021 -3615 -3589 -3391 -3365 -3246 -2667 -2478 -2376 -2319 -2301 -2298 -2147 -1974 -1892 -1838 -1791 -1779 -1175 -1141 -1136
-910 -565 -554 -286 -263 -226 -148 101 312 523 581 694 847 898 1106 1217 1248 1385 1454 1635 1816 2222 2301 2304 2877 3002 3
361 3597 3633 3872 4236 4238 4338 4357 4469 4487 4797 4849 5248 5711 5745 5784 5965 5989 5989 6137 6156 6307 6496 6654 6751 7552 758
3 7783 7922 8000 8230 8346 8486 8516 8694 8884 8921 9052 9070 9098 9191 9269 9354 9426 9532 9599 9811 9977

The amount of comparisons: 4950
The amount of swaps: 2446
```

Рисунок 3.1.1 – Сорткування масиву на 100 елементів (Bubble Sort)

```
Initial array is:

CombSort
-4969 1798 4159 -2476 4309 6983 4028 8634 -919 -1267 7578 -1376 3012 4186 -297 -4521 -4887 9598 2828 -1295 6133 9902 4684 7114 1818 3195
-1059 8158 6732 1125 7171 7337 2023 687 8722 9347 8650 4543 1713 -3463 9855 -107 2664 2124 -4062 1314 -2774 1185 1307 -3105 -2371 8365 -3
396 5698 5550 -4871 7115 7046 5706 -3886 3041 5870 -3099 7956 1563 -4760 6602 -2675 5465 357 -3171 570 2885 1291 -1149 1240 2238 -2441 617
5 -4496 968 7757 1851 4837 -1504 8806 5732 6325 5137 8592 -2769 -3397 8837 598 3497 -1015 -2009 4708 -4439 -4302

Execution time in ms 0,5026

Sorted array is

CombSort
-4969 -4887 -4871 -4760 -4521 -4496 -4439 -4302 -4062 -3886 -3463 -3397 -3396 -3171 -3105 -3099 -2774 -2769 -2675 -2476 -2441 -2371 -2009 -1504 -1376 -1295
-1267 -1149 -1059 -1015 -919 -297 -107 357 570 598 687 968 1125 1185 1240 1291 1307 1314 1563 1713 1798 1818 1851 2023 2124 2238 2
664 2828 2885 3012 3041 3195 3497 4028 4159 4186 4309 4543 4684 4708 4837 5137 5465 5550 5698 5706 5732 5870 6133 6175 6325 6602 673
2 6983 7046 7114 7115 7171 7337 7578 7757 7956 8158 8365 8592 8634 8650 8722 8806 8837 9347 9598 9855 9902

The amount of comparisons: 1229
The amount of swaps: 239
```

Рисунок 3.1.2 – Сорткування масиву на 100 елементів (Comb Sort)

```
Enter the length of array to generate:
1000

Initial array is:

BubbleSort
-2837 395 2153 2167 -2035 1971 -1252 973 5178 -3287 -1972 1132 -3605 7602 -3436 2612 -2734 5334 6354 -704 8350 5624 -1220 9561 59 732
-2741 -4915 -808 9511 6379 -2354 7990 -994 8836 5507 -2297 6259 4350 1460 -2435 1281 1182 6617 496 9080 -1369 5257 6621 6809 827 7728
497 7520 429 1540 -3386 2592 -2892 7233 -289 -1338 -3220 5668 335 -4618 3142 7389 8766 1058 -92 1007 -4016 -106 -3270 2068 -558 2164 406
4 979 -2900 2002 -1535 2384 3683 8651 7150 -4531 8578 2296 3893 5111 4151 7117 -3252 8904 -500 5514 1282 114 938 1243 6491 563 1900
9820 6743 -4398 5790 8361 -2902 4450 8529 6129 5132 7710 -4804 4397 -3016 1684 -1768 2672 8812 1470 7165 4798 6857 -1132 -1854 -2174 -2432
3928 -4118 -1050 -4959 5858 5512 7902 -4517 3863 -3093 6547 -320 -4783 197 9443 6583 8023 -3015 -962 8707 5104 3529 9598 -3119 8266 2431
21 3398 7103 -2037 2875 7457 2963 5793 3171 3568 4579 6087 9274 -2803 -2706 3913 3122 -223 1686 -2008 8595 -3728 7723 8215 8834 -3117 9108
-881 2464 -4602 -3715 -2932 -3224 -1574 491 -496 -1619 900 4032 9091 6717 -576 7027 875 -2996 -3065 -2289 -3160 6412 -3810 -4593 4820 5196
5273 3837 -1252 5675 8888 -871 992 8349 -1157 -1672 2772 2282 2510 4311 5847 6707 7791 532 -2779 5745 2595 1276 4170 -754 1848 6210 6
899 9172 7267 2707 1274 9991 1465 1872 1503 4589 -3258 3829 -507 -3230 -1960 -4259 2873 -3904 -746 8593 4494 2625 1864 134 8622 5659 818
6 9076 -2399 -1003 9668 9502 -1173 -2323 2231 1820 -3918 6136 7055 2782 -4183 -4362 -2193 -3743 4964 -1038 -3962 -4549 2705 1854 8403 3073 9992
3256 -3284 -727 2308 2133 4086 5615 -3196 2390 -729 -2494 4922 -2760 2 -3185 4540 6220 9660 6737 -958 6473 7776 3856 -182 606 5170
4660 4621 7735 1371 5747 594 1595 9511 5572 -4039 -2043 9513 3949 -2950 -3561 977 3427 6367 -4700 1336 2723 8961 9223 912 6485 2194 -33
35 -970 -1481 5480 2860 9456 -3299 6964 9427 -4007 -930 -3982 3524 6912 -1674 -2270 4430 -4146 1582 6708 710 4687 -2397 495 -652 7489 -483
9270 -2554 5699 5399 1823 4887 -1448 -1072 -2436 5991 -4909 3102 -1498 7811 6986 -2377 -4555 -3093 2409 -2157 -1612 -1990 -7 -470 8369 9331
9777 -3516 1979 -3573 8024 1132 5371 4939 -2569 4517 -1273 595 7247 1132 -29 2984 -4239 1739 4339 2753 -1921 8898 3724 2063 -2843 -1413 6
940 4646 -3872 6260 7454 1169 7031 2493 5111 -2932 6950 -3863 7967 3959 7731 3564 5379 -2532 8896 4851 -340 -3724 7872 5405 -2752 8254 519
1 615 6251 -4122 2760 9916 9271 4636 4193 -4276 3810 1172 7064 7559 3565 -246 6022 -2097 6035 9625 3772 -4620 5022 4516 -1481 1256 3167
2306 8928 5663 6775 723 4012 9201 5978 1415 6242 4232 4308 7034 -3133 -1469 9767 7086 9754 -793 405 -3663 -3715 9907 255 4969 9371 -
2376 -519 4811 5594 -3265 7360 -3496 1749 1964 5509 8531 -1050 -1914 7161 7864 -3420 -2850 3512 -3210 3595 9001 7758 241 -4386 8017 4762 93
13 -4150 1829 1020 9022 408 -3194 1352 1387 4719 -3833 5821 4507 6872 2804 284 4903 3022 -3765 3056 6667 2046 2541 9758 9136 1394 -1535
-3627 -4631 -133 9240 4197 7578 1973 8048 -4347 9920 -4057 6408 3122 -2856 6735 773 -1856 -4588 1329 9006 -82 -4210 3717 -2146 7220 9379
-40 8095 2266 3310 2072 1885 2627 -3058 -4568 982 -3480 5400 1970 8244 -2467 8825 2151 365 3959 6535 3113 911 9933 -3727 -603 9345 2
```

```
-3627 -4631 -153 9240 4197 7578 1973 8048 -4347 9920 -4057 6488 3122 -2056 6735 773 -1056 -4580 1329 9006 -82 -4210 3717 -2146 7220 9379
-40 8095 2266 3310 2072 1885 2627 -3058 -4568 982 -3400 5400 1970 8244 -2467 8825 2151 365 3959 6535 3113 911 9933 -3727 -603 9345 2
807 2388 8547 -4418 1653 -2988 -2981 4819 2617 3879 782 2344 2848 4059 -1140 1180 -3289 752 -748 2664 3442 3086 6472 1094 3640 -1649 389
6 -4315 1265 9050 8162 992 -1088 2727 9079 -4761 -1604 9851 9330 9509 3460 7152 627 3655 3042 9807 7517 9900 2674 -3067 -800 4181 7021
6733 992 9721 5917 2453 1942 -1698 3116 -3666 -1446 3568 5717 9790 -1569 -4879 4981 -1379 -2217 -3139 1422 -256 9204 6291 -2541 -2183 8052
3833 834 3996 3011 -669 4849 -1020 3971 2180 -3176 9358 -678 5943 2988 6028 1392 2126 3598 7059 7938 3253 6376 7396 -81 -947 9150 -26
61 -2242 2276 7529 4612 7262 3099 4401 8110 7468 9121 3323 9223 8719 1736 818 6252 8297 5624 9299 1841 3954 5395 8353 2078 -2307 400
-120 -201 7458 3438 -923 -1042 181 2510 -3778 -3306 4651 2213 -2905 6224 -3459 4799 8560 2272 8803 4375 8341 1201 -2566 -4876 9100 4295
-246 2309 3385 2363 3924 451 1918 1862 -1781 -398 -4349 -971 5043 -635 4042 1208 514 -877 -3400 1084 3528 -1925 299 9156 6836 1081 3
275 3319 2059 8679 -2635 7333 4339 4782 9204 7872 9125 3859 2771 -801 -4706 5233 3004 6940 4147 4531 7229 4438 -4668 2132 -4314 8311 852
4 3015 -4233 -3321 -2990 4230 5528 -2158 7176 8600 9803 -459 -2764 -1682 4691 -4740 161 6171 2083 6656 6353 7402 790 8295 5071 3359 9
-2696 3952 8837 2986 9928 7992 -946 8810 -3677 4015 7524 5861 -105 8331 -72 2023 -3306 5431 -1537 4703 828 9841 6969 -723 -838 -4951
-969 -3042 -3878 -2775 -2528 1609 3652 -779 3224 3991 8340 614 9876 1549 8183 -491 3367 1245 2637 -4691 -2812 8326 1424 378 -904 -2136 16
08 8656 9181 -2536 6883 4560 -1176 9879 3961 -2904 7687 9099 1543 915 -2791 -4327 107 5938 3684 6080 5555 6914 6521 -3682 1474 724 6640
-1148 7861 706 5534 5745 8129 9608 -5000 4996 40 759 7108 -1221 -3210 9958 7589 6086 1822 4260 9703 -3279 8101 -3486 3219 5487 6694
3016 2310 4694 -2865 7863 5991 1159 7920 9972 -2983 -1746 1562 -3058 -4030 1935 914 2213 5634 9348 8830 -95 4290 5557 9325 -4566 4497 7
206 1021 4918 752 6347 7408 -4953 9052 112 2298 306 2861 2178 4768 948 859 -3182 7981 4334 -477 -121 3298 2760 5008 887 881 307
1 6657
Execution time in ms 8,7222

Sorted array is

BubbleSort
-5000 -4959 -4953 -4951 -4915 -4909 -4879 -4876 -4804 -4783 -4761 -4740 -4706 -4700 -4691 -4668 -4631 -4620 -4618 -4602 -4593 -4588 -4568 -4566 -4555 -4549
-4531 -4517 -4418 -4398 -4386 -4362 -4349 -4347 -4327 -4315 -4314 -4276 -4259 -4239 -4233 -4210 -4183 -4150 -4146 -4122 -4118 -4057 -4039 -4030 -4016 -4007 -3
982 -3962 -3918 -3904 -3878 -3872 -3863 -3833 -3810 -3778 -3765 -3743 -3728 -3727 -3724 -3715 -3715 -3682 -3677 -3666 -3663 -3627 -3605 -3573 -3561 -3516 -349
6 -3486 -3480 -3459 -3436 -3420 -3400 -3386 -3335 -3321 -3306 -3306 -3299 -3289 -3287 -3284 -3279 -3270 -3265 -3258 -3252 -3230 -3224 -3220 -3210 -3196
-3194 -3185 -3182 -3176 -3160 -3139 -3133 -3119 -3117 -3093 -3093 -3067 -3065 -3058 -3058 -3042 -3016 -3015 -2996 -2990 -2988 -2983 -2981 -2950 -2932 -2932 -2905
-2904 -2902 -2900 -2892 -2865 -2856 -2850 -2843 -2837 -2812 -2803 -2791 -2779 -2775 -2764 -2760 -2752 -2741 -2734 -2706 -2696 -2661 -2635 -2569 -2566 -25
54 -2541 -2536 -2532 -2528 -2494 -2467 -2436 -2435 -2432 -2399 -2397 -2377 -2376 -2354 -2323 -2307 -2297 -2289 -2270 -2242 -2217 -2193 -2183 -2174 -2158 -2157
```

```
32 563 594 595 606 614 615 627 706 710 723 724 732 752 752 759 773 782 790 818 827 828 834 859 875 881 887
900 902 911 912 914 915 938 948 973 977 979 982 992 992 1007 1020 1021 1058 1081 1084 1094 1132 1132 1132 1169 1169
1172 1180 1182 1201 1208 1243 1245 1256 1265 1274 1276 1281 1282 1329 1336 1352 1371 1387 1392 1394 1415 1422 1424 1460 1465 1478 1
474 1503 1540 1543 1549 1562 1582 1595 1608 1609 1653 1684 1686 1736 1739 1749 1820 1822 1823 1829 1841 1848 1854 1862 1864 1872 188
5 1908 1918 1915 1924 1964 1970 1971 1973 1979 2002 2023 2046 2059 2063 2068 2072 2078 2083 2126 2132 2133 2151 2153 2164 2167 2178
2180 2194 2213 2213 2231 2266 2272 2276 2282 2296 2298 2306 2308 2309 2310 2344 2363 2384 2388 2390 2409 2431 2453 2464 2493 2510
2510 2541 2592 2595 2612 2617 2625 2627 2637 2664 2672 2674 2705 2707 2723 2727 2753 2760 2768 2771 2772 2782 2804 2807 2848 2860 28
61 2873 2875 2908 2963 2984 2986 3004 3011 3015 3016 3022 3042 3056 3073 3086 3099 3102 3113 3116 3122 3122 3142 3167 3171 3219 3224
3253 3256 3275 3298 3310 3319 3323 3359 3367 3385 3398 3427 3438 3442 3460 3512 3524 3528 3529 3564 3565 3568 3568 3595 3598 3640
3652 3655 3687 3684 3717 3724 3772 3810 3829 3833 3837 3852 3856 3859 3863 3871 3879 3893 3896 3913 3924 3928 3949 3954 3959 3959 3
961 3971 3991 3996 4012 4015 4032 4042 4059 4064 4086 4147 4151 4170 4181 4193 4197 4230 4232 4268 4290 4295 4308 4311 4334 4339 433
9 4350 4375 4397 4401 4430 4436 4438 4450 4494 4497 4507 4516 4517 4531 4540 4560 4579 4589 4612 4621 4646 4651 4660 4687 4691 4694
4703 4719 4742 4768 4782 4798 4799 4811 4819 4820 4849 4851 4887 4903 4918 4922 4939 4964 4969 4981 4996 5022 5043 5071 5104 5111
5111 5132 5170 5178 5191 5196 5233 5257 5273 5334 5371 5379 5395 5399 5455 5431 5480 5480 5487 5507 5509 5512 5514 5528 5534 5555 55
57 5572 5594 5615 5624 5624 5634 5659 5663 5668 5675 5699 5717 5745 5745 5747 5790 5793 5808 5821 5847 5858 5861 5917 5938 5943 5978
5991 5991 6022 6028 6035 6080 6086 6087 6129 6136 6171 6210 6220 6224 6242 6251 6252 6259 6260 6291 6347 6353 6354 6367 6376 6379
6412 6472 6473 6485 6488 6491 6521 6535 6547 6583 6617 6621 6640 6656 6657 6667 6694 6707 6708 6717 6733 6735 6737 6743 6775 6809 6
836 6857 6872 6883 6899 6912 6914 6940 6940 6950 6964 6969 6986 7006 7021 7027 7031 7034 7055 7059 7064 7103 7108 7117 7150 7152 716
1 7165 7176 7206 7220 7229 7233 7247 7262 7267 7333 7360 7389 7396 7402 7454 7457 7458 7468 7488 7489 7517 7520 7524 7529 7559 7578
7589 7602 7607 7710 7732 7728 7731 7735 7758 7776 7779 7811 7861 7863 7864 7872 7877 7902 7920 7938 7967 7981 7990 7992 8017 8023
8024 8048 8052 8095 8101 8110 8129 8162 8183 8186 8215 8244 8254 8266 8295 8297 8311 8326 8331 8340 8341 8349 8350 8353 8361 8369 84
03 8524 8529 8531 8547 8560 8578 8593 8595 8600 8622 8651 8656 8679 8707 8719 8766 8803 8810 8812 8825 8830 8834 8836 8837 8888 8896
8898 8904 8928 8961 9001 9006 9022 9050 9052 9076 9079 9080 9091 9099 9108 9108 9121 9125 9136 9150 9156 9172 9181 9201 9204 9204
9223 9223 9240 9270 9271 9274 9299 9313 9325 9330 9331 9345 9348 9358 9371 9379 9427 9443 9456 9502 9509 9511 9511 9513 9561 9598 9
608 9625 9660 9668 9703 9721 9754 9758 9767 9777 9790 9803 9807 9820 9841 9851 9876 9879 9900 9907 9916 9920 9928 9933 9972 9991 999
2 9998
The amount of comparisons: 499500
The amount of swaps: 239784
```

## Рисунок 3.2.1 – Сортунання масиву на 1000 елементів(Bubble Sort)

```
Initial array is:

CombSort
-2737 2534 9966 7848 7845 1936 7695 -738 231 5299 3511 3509 1987 7596 -3894 5884 1263 9012 -473 3462 2626 -2130 3290 7425 3992 4155
471 824 -23 -2608 9684 -4036 -537 347 5272 5099 -396 2542 9571 -4285 201 7705 2235 4686 5630 2875 5808 -2652 1726 449 -3831 -2857 3
154 3657 5610 9393 9245 -2460 -1144 -1310 -3536 1871 9286 -2161 5616 2546 -954 -3361 -2677 1807 1280 8722 9442 -1189 -1740 4922 5635 8604 -320
5 -3884 8347 -1877 9098 9528 -3778 7540 2810 5806 4738 5880 5855 -2937 -1314 5881 9252 8331 8419 -605 9071 -1917 -1172 4168 -4976 9919 9170
-2552 -2299 -657 397 -4312 -1242 -4360 -4255 -2261 343 8341 2773 9759 -3383 8076 7737 3240 1902 1444 5953 -784 5036 -1440 -440 -70 6503
1239 9533 8712 7289 -821 -1901 9180 -4643 8775 4838 -1533 409 4049 2122 5214 -3683 -4053 538 -4090 557 8660 -2222 -4005 -1653 5588 -2735 35
81 -1035 4954 2413 9142 9784 -4534 -815 7519 -3893 -2967 450 -3837 5242 3594 9402 -2157 -1763 6269 -2169 7854 9322 -4261 -2102 4089 8023 -1351
9374 7844 -1559 8538 5364 -3300 -3086 4518 5214 404 1362 5697 -1316 -58 -136 4603 4106 9284 5497 -626 668 6964 9259 9936 7758 2078
2757 -588 4259 -1774 2055 8325 -3882 6134 -3907 1341 4057 -2430 1670 -1250 9249 -4276 8380 3998 6482 2540 2585 7889 2989 -281 1634 -1386 8
898 9804 8928 8961 9001 9006 9022 9050 9052 9076 9079 9080 9091 9099 9108 9108 9121 9125 9136 9150 9156 9172 9181 9201 9204 9204
593 3648 7982 -2591 8586 -1474 -4220 -1714 3943 7503 -4742 1987 8349 3037 4578 2551 -1151 8523 -4454 1536 2614 1746 125 -56 8584 260 -154
6 217 9493 4868 6423 -460 6593 -1560 9733 2471 8153 8684 2756 531 -3211 -1617 4807 -3254 -4535 1392 5649 -1556 7963 -4966 4040 4666 -028
-4903 6000 1846 -1478 3037 -1174 3325 -4929 5563 -1207 9075 -3921 5718 5395 4908 3305 -2893 4398 165 767 5015 -254 -769 6939 -1621 -522
2191 4611 -4991 2255 9459 -4172 -3044 -4633 6495 2441 2886 -1052 115 4858 9431 459 1226 9591 5732 -4334 -252 5998 -4259 5139 5855 636 -11
26 -1301 4158 5352 1769 4907 -4857 -1441 6636 8715 -4938 -1692 -665 8247 9123 5350 8901 -2533 -4581 8160 -717 5308 1336 -445 8520 -360 -3793
-2288 9731 6193 3870 -129 6401 3822 6954 8258 1210 -3857 -4329 7979 -1195 -3444 8222 -4284 397 -2082 5256 7486 -3457 4398 7898 8274 4720
7696 -3647 3567 7364 -398 6832 -2872 3700 7431 -1172 -4558 -156 5191 1908 5514 -3708 -4135 -860 7758 -3204 -2560 4355 -2273 7086 619 9158 5
602 4385 -1279 -3832 332 9096 5596 5151 8705 2468 -1613 -4107 -3393 -689 8543 -721 1374 7728 7780 5873 5207 6010 9742 6194 -4322 258 934
0 6478 2279 -1660 8627 6481 1744 445 -3771 2744 -21 6263 8497 -3654 4248 -4961 3203 -1955 7905 9973 -2467 3446 -1747 8162 -2020 -1 9307
4747 -1049 9593 4643 2528 -4510 -4926 9756 9119 -2129 5365 3842 1907 4150 6163 1919 799 9301 5111 9232 6433 -3376 5167 -2194 6697 8289 -
4294 -2276 894 3224 4118 -4641 3627 7174 6997 -3507 -3363 8101 7451 5713 8422 -3410 5130 7168 7839 4203 998 822 1371 365 7455 8862 28
70 768 -2126 3123 3785 6786 7947 4239 9785 7350 3862 894 3754 5005 -4785 4733 7153 4669 -3557 2698 -3151 9339 -4191 4842 1495 7999 -32
-2506 2291 4884 6354 -4471 5770 5257 2490 -2142 4300 -1126 2709 -148 -4046 -1789 5988 -3929 2729 -765 -1586 -2166 3907 6265 3669 6702 -1079
-244 7012 3531 -41 3032 -3373 5787 1817 5452 -3052 -100 -4005 3333 5125 8497 -2486 7357 -4567 3279 -3040 6592 2544 8638 -1650 9738 -2887 -2
740 4584 -2867 3507 6027 -3785 9653 6608 1013 8245 8959 7900 -3732 -2754 6923 8035 -298 -3737 1088 -4925 9396 4110 -83 -4680 6630 5831 -461
1 4742 2330 7213 8374 181 221 918 9368 -3798 -317 -3055 922 -1315 -3657 6725 -889 -4749 -1445 9458 9166 1032 38 8557 -2159 7375 7263
9642 -2283 3150 -333 -3187 6877 -4805 -4175 -89 -722 3801 9872 3778 2679 5350 -2121 5720 3352 -2243 5341 -2598 8733 -2329 -1 -335 6765
```

```
332 -4331 2841 1390 1313 2823 9615 2798 5755 4546 -3485 4336 -4742 8372 -1968 422 1580 2999 7198 1730 5507 -3361 7379 2282 4605 5357 -3
421 -4086 5357 1917 -2377 7033 2504 142 7807 9202 -1145 -353 8703 -2588 -918 -2747 6142 1445 2315 -2204 1860 6097 2807 -743 7110 5423 -281
2 -2669
Execution time in ms 1,847

Sorted array is

CombSort
-4999 -4993 -4991 -4976 -4966 -4961 -4954 -4939 -4938 -4929 -4926 -4925 -4916 -4903 -4857 -4855 -4846 -4805 -4793 -4785 -4749 -4748 -4742 -4742 -4736 -4680
-4643 -4641 -4633 -4620 -4611 -4581 -4567 -4558 -4535 -4534 -4510 -4471 -4454 -4447 -4410 -4360 -4334 -4331 -4329 -4323 -4322 -4312 -4294 -4285 -4284 -4276 -4
265 -4261 -4259 -4255 -4220 -4191 -4182 -4175 -4172 -4141 -4135 -4107 -4090 -4071 -4053 -4036 -4010 -4006 -4005 -4005 -4001 -3929 -3921 -3987 -3893 -3888 -388
4 -3882 -3857 -3837 -3832 -3831 -3826 -3798 -3793 -3785 -3778 -3771 -3737 -3732 -3708 -3683 -3657 -3654 -3647 -3564 -3557 -3536 -3507 -3492 -3491 -3485 -3457
-3455 -3444 -3435 -3421 -3419 -3410 -3399 -3393 -3383 -3376 -3373 -3363 -3361 -3361 -3300 -3254 -3211 -3205 -3204 -3188 -3187 -3168 -3165 -3151 -3094 -3086 -
3085 -3055 -3052 -3045 -3044 -3040 -3012 -2994 -2967 -2966 -2937 -2893 -2887 -2881 -2881 -2872 -2867 -2857 -2812 -2803 -2754 -2747 -2740 -2737 -2735 -2701 -26
77 -2669 -2652 -2608 -2604 -2598 -2591 -2588 -2584 -2560 -2552 -2533 -2526 -2519 -2512 -2506 -2486 -2467 -2460 -2430 -2383 -2377 -2350 -2329 -2299 -2288 -2283
-2276 -2273 -2261 -2260 -2243 -2224 -2222 -2204 -2199 -2194 -2169 -2166 -2161 -2159 -2157 -2142 -2140 -2130 -2129 -2126 -2121 -2102 -2098 -2082 -2020
-1968 -1955 -1940 -1933 -1919 -1917 -1910 -1901 -1877 -1862 -1850 -1849 -1847 -1846 -1800 -1789 -1774 -1763 -1760 -1747 -1740 -1714 -1692 -1660 -1653 -1650 -1
621 -1617 -1613 -1586 -1579 -1560 -1556 -1546 -1539 -1533 -1485 -1478 -1474 -1471 -1445 -1441 -1440 -1431 -1386 -1351 -1316 -1315 -1314 -1310 -1310 -1303 -130
1 -1279 -1250 -1242 -1207 -1195 -1189 -1185 -1174 -1172 -1172 -1155 -1151 -1145 -1144 -1129 -1126 -1126 -1079 -1052 -1049 -1035 -1004 -954 -918 -889 -860
-828 -821 -815 -784 -769 -765 -743 -738 -722 -721 -717 -715 -689 -665 -657 -626 -605 -588 -537 -522 -512 -473 -460 -445 -440 -431
-398 -396 -386 -371 -360 -353 -341 -335 -333 -317 -298 -297 -281 -254 -252 -244 -179 -179 -156 -148 -136 -129 -100 -89 -83 -70 -
58 -56 -41 -32 -23 -21 -1 -1 -1 5 16 36 38 48 68 115 125 142 144 144 165 181 201 205 217 221 231 258 260
332 332 343 347 365 365 394 397 397 404 409 422 445 449 450 459 471 481 531 538 557 558 559 619 636 651
663 668 727 736 767 768 799 822 824 894 894 918 922 998 1000 1013 1032 1045 1054 1074 1088 1115 1203 1210 1218 1226 1
239 1263 1280 1308 1313 1336 1341 1362 1363 1371 1374 1390 1392 1404 1444 1445 1468 1491 1493 1495 1519 1536 1537 1541 1580 1634 167
0 1726 1730 1737 1744 1746 1765 1769 1779 1795 1807 1817 1821 1846 1860 1871 1890 1896 1902 1907 1908 1917 1936 1936 1987 1987 2022
2026 2055 2063 2078 2118 2122 2150 2168 2168 2191 2235 2255 2279 2282 2291 2315 2324 2330 2413 2434 2441 2446 2468 2471 2482 2483
2490 2504 2528 2534 2540 2542 2544 2546 2551 2556 2585 2614 2626 2631 2660 2669 2679 2698 2709 2729 2744 2756 2757 2770 2773 2798 28
07 2810 2823 2841 2851 2870 2875 2878 2886 2886 2889 2918 2940 2989 2999 3032 3034 3037 3037 3047 3071 3073 3123 3150 3154 3203 3224
3240 3278 3279 3290 3305 3325 3333 3352 3446 3462 3462 3507 3509 3511 3531 3536 3567 3567 3574 3581 3594 3594 3627 3648 3657 3669
3700 3754 3774 3778 3785 3788 3801 3822 3831 3842 3862 3870 3917 3943 3954 3987 3992 3998 4040 4049 4057 4062 4089 4106 4110 4118 4
150 4155 4158 4165 4168 4203 4239 4248 4259 4286 4300 4310 4318 4336 4355 4357 4371 4385 4398 4398 4466 4468 4518 4546 4578 4584 460
3 4605 4605 4611 4643 4654 4669 4680 4686 4689 4720 4733 4738 4742 4747 4761 4777 4807 4838 4842 4858 4860 4868 4884 4907 4908 4922
4954 4990 5004 5005 5015 5036 5051 5072 5099 5111 5125 5130 5133 5139 5151 5167 5191 5207 5214 5214 5225 5236 5242 5256 5257 5272
5299 5308 5311 5318 5341 5350 5350 5352 5357 5357 5364 5365 5386 5395 5423 5427 5452 5497 5505 5507 5514 5563 5573 5575 5588 5596 56
02 5608 5610 5616 5630 5635 5649 5679 5697 5716 5718 5720 5732 5755 5770 5773 5787 5806 5808 5831 5839 5895 5895 5895 5895 5880 5880 5881
5884 5897 5902 5953 5978 5988 5998 6000 6010 6027 6037 6048 6055 6097 6128 6134 6142 6163 6185 6193 6194 6226 6263 6265 6269 6269
6354 6392 6401 6420 6423 6433 6433 6448 6478 6481 6482 6495 6503 6535 6536 6592 6593 6608 6630 6636 6661 6666 6666 6697 6697 6702 6
725 6765 6782 6786 6799 6818 6832 6833 6877 6923 6939 6942 6954 6964 6997 7012 7033 7086 7098 7110 7133 7153 7168 7173 7174 7198 721
3 7263 7289 7350 7357 7364 7375 7379 7420 7425 7431 7451 7455 7486 7491 7503 7519 7540 7596 7689 7695 7696 7705 7728 7758 7758 7807
7807 7822 7835 7839 7844 7845 7848 7854 7889 7899 7899 7900 7905 7935 7947 7963 7979 7982 7999 8015 8023 8035 8059 8076 8101 8128
8153 8160 8162 8222 8245 8247 8255 8258 8260 8274 8284 8289 8325 8331 8333 8341 8347 8349 8362 8369 8372 8374 8380 8419 8422 8444 84
97 8497 8520 8523 8538 8543 8557 8584 8586 8593 8604 8627 8630 8638 8660 8684 8703 8705 8712 8715 8722 8733 8742 8775 8804 8809 8862
8868 8873 8901 8910 8959 8966 8990 9012 9039 9071 9075 9096 9098 9108 9119 9123 9142 9158 9166 9170 9180 9185 9202 9232 9245 9249
9252 9259 9284 9286 9301 9307 9322 9339 9340 9368 9374 9387 9393 9396 9402 9431 9431 9442 9458 9459 9493 9494 9528 9533 9543 9571 9
591 9593 9615 9653 9662 9670 9684 9704 9711 9724 9731 9733 9737 9738 9742 9756 9759 9772 9784 9785 9871 9872 9907 9919 9936 9951 996
6 9973
The amount of comparisons: 22022
The amount of swaps: 4342
```

Рисунок 3.2.1 – Сортування масиву на 1000 елементів(Comb Sort)



### 3.5 Тестування алгоритму

#### 3.5.1 Часові характеристики оцінювання

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

Таблиця 3.2.1 – Характеристики оцінювання алгоритму сортування бульбашки для упорядкованої послідовності елементів у масиві

Розмірність масиву	Число порівнянь (Bubble Sort)	Число перестановок (Bubble Sort)
10	45	0
100	4950	0
1000	499500	0
5000	12497500	0
10000	49995000	0
20000	199990000	0
50000	1249975000	0

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

Розмірність масиву	Число порівнянь (Comb Sort)	Число перестановок (Comb Sort)
10	36	0
100	1229	0
1000	22022	0
5000	144862	0
10000	329644	0
20000	719230	0
50000	1997920	0

В таблицях 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 3.3.1 – Характеристики оцінювання алгоритму сортування бульбашки для зворотно упорядкованої послідовності елементів у масиві.

Розмірність масиву	Число порівнянь	Число перестановок
10	45	45
100	4950	4950
1000	499500	499500
5000	12497500	12497500
10000	49995000	49995000
20000	199990000	199990000
50000	1249975000	1249975000

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

Розмірність масиву	Число порівнянь	Число перестановок
10	36	9
100	1229	110
1000	22022	1512
5000	144862	9016
10000	329644	19132
20000	719230	40720
50000	1997920	109522

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

Таблиця 3.4.1 – Характеристика оцінювання алгоритму сортування бульбашки для випадкової послідовності елементів у масиві.

Розмірність масиву	Число порівнянь	Число перестановок
10	45	14
100	4950	2458

1000	499500	249247
5000	12497500	6272819
10000	49995000	25125273
20000	199990000	100220872
50000	1249975000	625486366

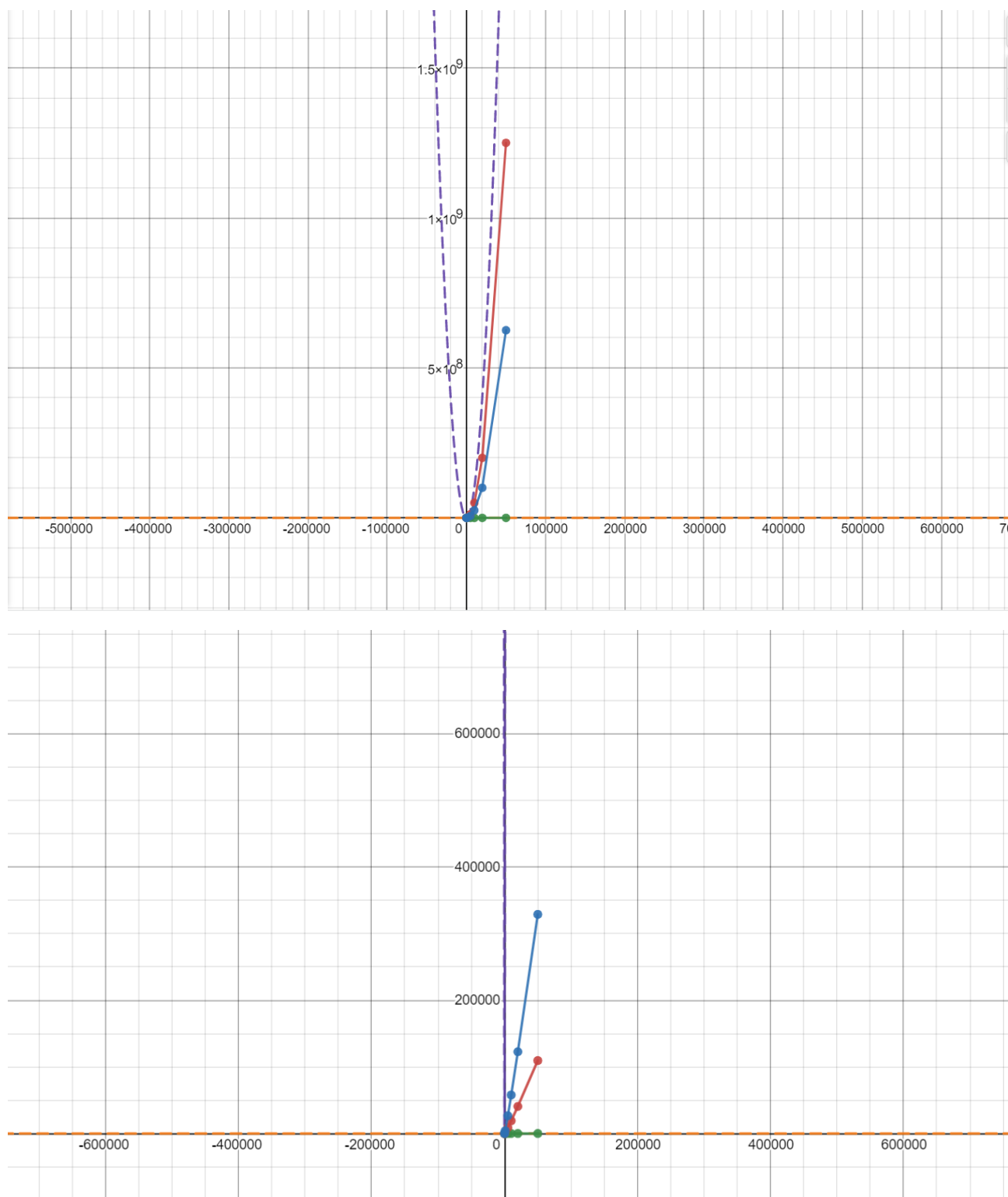
Таблиця 3.4.1 – Характеристика оцінювання алгоритму сортування гребінцем для випадкової послідовності елементів у масиві.

Розмірність масиву	Число порівнянь	Число перестановок
10	36	11
100	1229	237
1000	22022	4189
5000	144862	26799
10000	329644	57804
20000	719230	122814
50000	1997920	329078

### 3.5.2 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 (зелений графік), коли масиви містять зворотно упорядковану послідовність елементів (червоний графік), коли масиви містять випадкову послідовність елементів (синій 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 дослідила алгоритми сортування бульбашкою та гребінцем. Я провела аналіз алгоритмів на відповідність властивостям, проаналізувала часову складність алгоритмів та порівняла часові характеристики.