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, Мєшков Андрій Ігорович
(шифр, прізвище, ім'я, по батькові)

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

Київ 2022

ЗМІСТ

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 на відповідність властивостям наведено в таблиці 3.1.

Таблиця 3.1 – Аналіз алгоритму на відповідність властивостям

Властивість	**Сортування бульбашкою**
Стійкість	+
«Природність» поведінки (Adaptability)	+
Базуються на порівняннях	+
Необхідність в додатковій пам'яті (об'єм)	–
Необхідність в знаннях про структури даних	+

3.2 Псевдокод алгоритму

```
for i=0 to size-1 step 1
        is:=false
        for j=0 to size-i-1 step 1
                if arr[j] > arr[j+1]
                        then
                        temp:= arr[j]
                        arr[j]:= arr[j + 1]
                        arr[j + 1]:= temp
                        is:=true
        if not is
                then
                break
```

3.3 Аналіз часової складності

Найгірша швидкодія - $O(n^2)$

Найкраща швидкодія - $O(n)$

Середня швидкодія - $O(n^2)$

3.4 Програмна реалізація алгоритму

3.4.1 Вихідний код

```
58  int* bubble_sort(int arr[], int size){
59      int comparison=0, permutation=0;
60      int temp;
61      bool is;
62      for (int i = 0; i < size - 1; i++) {
63          is = false;
64          for (int j = 0; j < size - i - 1; j++) {
65              if (arr[j] > arr[j + 1]) {
66                  temp = arr[j];
67                  arr[j] = arr[j + 1];
68                  arr[j + 1] = temp;
69                  is = true;
70                  permutation++;
71              }
72              comparison++;
73          }
74          if(!is) break;
75      }
76      cout<<"comparison: "<<comparison<<endl;
77      cout<<"permutation: "<<permutation<<endl;
78      return arr;
79  }
```

3.4.2 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

```
n = 100
8 50 74 59 31 73 45 79 24 10 41 66 93 43 88 4 28 30 13 70 58 61 34 100 17 36 98 27 68 11 80 22 94 37 86 46 29 92 95 2 54 9 69 91 25 97 23 67 78 99 82 14 15
    64 26 16 18 96 6 5 52 89 83 53 38 39 72 32 76 7 63 20 55 90 71 21 35 51 60 48 40 12 19 62 77 75 57 85 56 81 3 1 65 49 84 47 44 33 42 87

comparison: 4950
permutation: 2461
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
```

Рисунок 3.2 – Сортування масиву на 1000 елементів

```
n = 1000
808 250 74 659 931 273 545 879 924 710 441 166 493 43 988 504 328 730 841 613 304 170 158 561 934 100 279 817 336 98 827 513 268 811 634 980 150 580 822 968
    673 394 337 486 746 229 92 195 358 2 154 709 945 669 491 125 197 531 904 723 667 550 25 802 854 978 409 299 982 636 14 866 815 64 537 426 670 116 95 630
    502 518 196 106 405 452 189 124 506 883 753 567 717 338 439 145 898 872 829 138 359 178 398 295 905 610 232 176 143 400 969 413 261 558 595 9 396 114 7
    963 943 366 83 853 768 696 713 672 902 591 832 739 58 617 791 641 680 973 99 96 320 455 224 290 761 906 127 507 814 771 239 221 845 367 535 227 395 364
    551 160 624 948 386 218 540 248 497 886 421 736 916 626 12 153 245 296 819 397 693 816 992 34 554 548 826 211 663 212 809 378 762 869 996 777 440 875 332
    557 302 873 985 756 790 408 16 194 770 681 456 856 964 503 677 109 639 998 652 850 204 732 532 15 420 776 10 181 930 55 738 546 318 526 201 257 565 598
    649 705 151 977 555 798 505 382 749 66 379 700 210 130 484 448 608 774 323 306 177 54 225 631 401 445 371 286 17 899 156 134 577 179 267 712 615 820 592
    721 763 198 32 589 590 874 878 305 255 961 481 731 956 547 108 1 927 36 858 115 594 361 355 419 586 564 103 918 644 748 270 474 104 460 407 825 133 974
    604 921 951 901 521 534 259 4 935 781 880 833 575 543 772 311 984 971 41 724 651 972 230 319 747 300 231 622 959 697 65 599 752 941 552 206 81 275 650
    414 321 26 13 784 789 118 325 258 512 691 411 566 743 404 353 308 142 911 800 128 172 788 415 642 349 843 316 975 446 374 912 240 254 190 167 357 861 593
    117 581 868 940 699 812 892 729 662 46 121 241 159 454 628 351 728 828 796 443 517 516 925 602 937 282 711 217 782 792 126 385 168 950 477 47 468 745 519
    719 690 919 29 30 35 755 541 936 84 49 253 877 238 676 246 416 193 38 716 694 165 801 90 671 847 39 544 339 322 720 360 236 913 685 990 785 735 668 520
    616 632 799 640 5 766 392 122 107 451 704 33 655 466 623 40 658 467 997 495 556 222 313 294 428 233 199 857 571 915 767 836 922 44 87 606 944 312 432 993
    837 834 329 684 490 435 27 523 365 891 885 442 180 538 871 562 582 803 733 333 855 760 848 463 11 942 113 957 391 45 102 962 511 929 806 174 97 422 301
    263 264 890 251 759 653 93 317 715 61 369 742 235 129 572 449 876 330 472 821 765 80 343 139 383 576 981 597 88 734 262 381 779 424 499 846 510 611 219
    291 56 744 276 152 252 438 101 285 68 141 708 563 970 314 979 508 657 787 740 585 515 20 664 553 289 164 380 462 488 354 900 881 284 237 910 568 835 111
    436 71 588 298 647 91 112 464 952 214 601 960 331 633 560 778 780 384 82 522 600 417 619 453 584 660 870 596 470 173 220 851 244 393 525 362 621 726 94
    123 469 24 794 342 356 903 348 947 852 524 737 867 926 140 695 161 293 277 465 485 120 860 146 399 144 487 514 967 192 175 751 528 1000 430 813 135 260
    754 226 859 492 769 688 527 473 205 475 345 849 618 839 509 807 200 256 830 489 897 865 19 479 309 324 678 954 272 476 271 266 280 76 185 157 665 887 48
    666 683 315 242 431 86 989 687 496 429 184 137 579 692 722 775 863 707 627 645 376 132 412 646 287 494 958 637 370 907 939 187 686 923 182 203 209 831
    938 920 909 578 882 31 326 437 995 946 603 234 457 459 202 444 406 119 344 213 933 783 917 265 155 28 953 387 423 818 215 888 682 559 434 21 805 908 698
    966 458 625 741 894 50 797 388 949 425 895 105 368 750 274 864 706 450 986 703 307 471 638 281 725 614 148 402 188 110 78 223 840 418 607 896 228 461 629
    727 297 62 53 8 377 498 823 278 501 60 85 569 914 583 171 136 89 346 757 310 23 149 375 186 57 163 363 862 433 773 609 37 675 928 810 482 976 987 884 42
    718 69 288 573 77 410 549 786 390 999 147 758 656 191 842 804 793 216 570 51 889 373 764 59 838 372 955 283 994 500 529 648 169 350 389 689 6 991 22 249
    3 208 303 674 327 73 536 965 542 844 483 605 70 824 679 243 427 612 269 292 539 661 67 79 480 347 334 893 247 587 52 72 654 932 643 620 530 702 447 335
    183 18 340 478 63 714 983 162 701 341 403 795 131 574 352 635 207 533 75
----------
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
    447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484
    485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522
    523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560
    561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598
    599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636
    637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674
    675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712
    713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750
    751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788
    789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826
    827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864
    865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902
    903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940
    941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978
    979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000
```

3.5 Тестування алгоритму

3.5.1 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 бульбашки для упорядкованої послідовності елементів у масиві

Розмірність масиву	Число порівнянь	Число перестановок
10	9	0
100	99	0
1000	999	0
5000	4999	0
10000	9999	0
20000	19999	0
50000	49999	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	45	45
100	4950	4950
1000	499500	499500
5000	12497500	12497500
10000	49995000	49995000
20000	199990000	199990000
50000	1249975000	1249975000

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

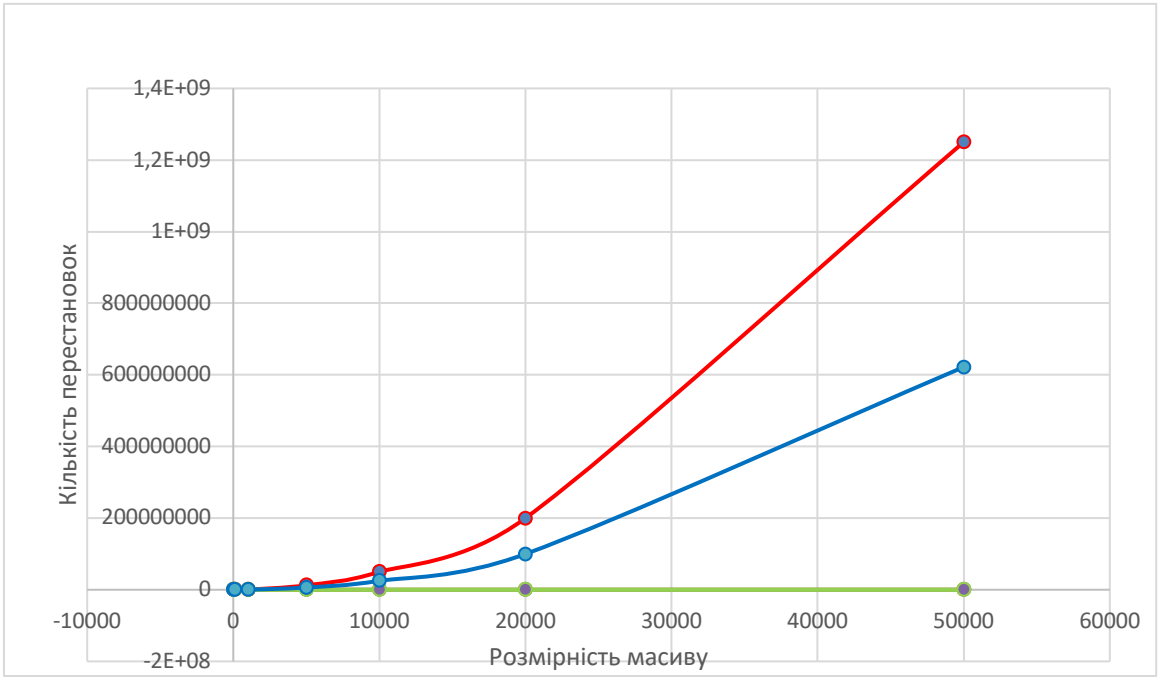
Таблиця 3.4 – Характеристика оцінювання алгоритму сортування бульбашки для випадкової послідовності елементів у масиві.

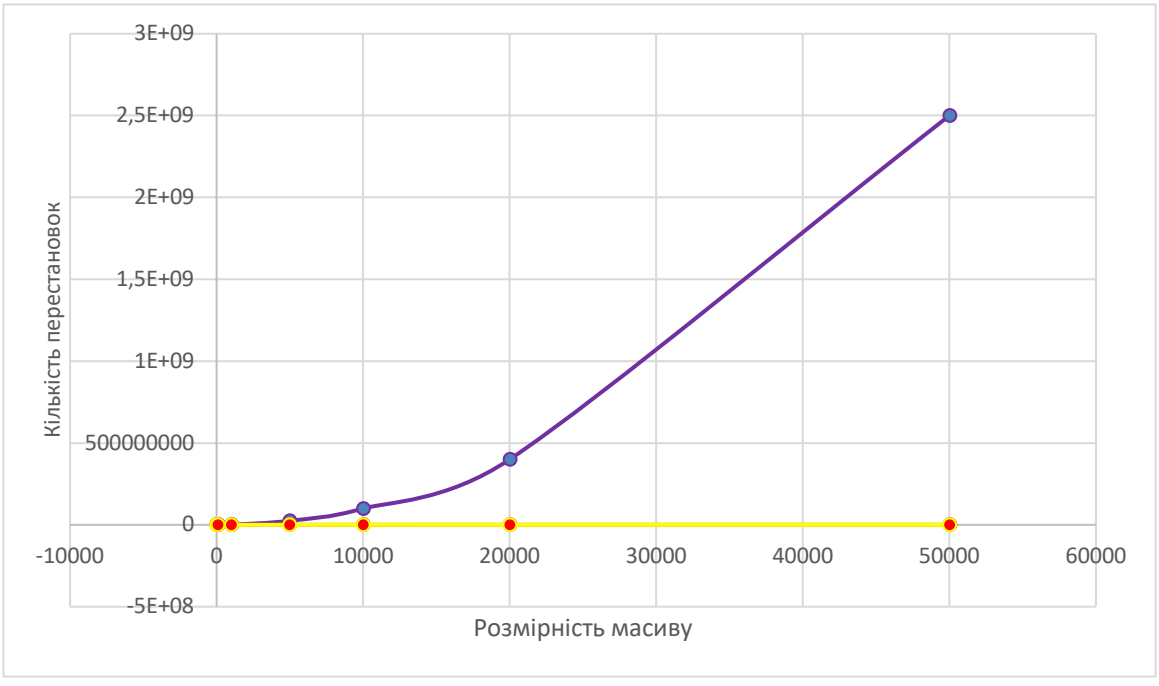
Розмірність масиву	Число порівнянь	Число перестановок
10	45	28
100	4922	2461
1000	498939	255736
5000	12476997	6265813
10000	49994259	24986962
20000	199989097	99905886
50000	1249914274	621884328

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





3.1 Аналіз алгоритму на відповідність властивостям

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

Таблиця 3.1 – Аналіз алгоритму на відповідність властивостям

Властивість	**Сортування гребінцем**
Стійкість	_
«Природність» поведінки (Adaptability)	_
Базуються на порівняннях	+
Необхідність в додатковій пам'яті (об'єм)	_
Необхідність в знаннях про структури даних	+

3.2 Псевдокод алгоритму

```
factor:= 1.2473309;
step:= size - 1
while  step>=1
        for i=0 to size-step step 1
                if arr[i] > arr[i+step]
                        then
                        temp:= arr[i]
                        arr[i]:= arr[j + step]
                        arr[i + step]:= temp
        step:=step/factor
```

3.3 Аналіз часової складності

Найгірша швидкодія - $\Omega(n^2)$

Найкраща швидкодія - Ω (nlogn)

Середня швидкодія - Θ (n^2/ 2p)

3.4 Програмна реалізація алгоритму

3.4.1 Вихідний код

```
69  int* comb_sort(int arr[], int size){
70      int temp;
71      int comparison=0, permutation=0;
72      double factor = 1.2473309;
73      int step = size - 1;
74      while (step >= 1){
75          for (int i = 0; i + step < size; i++){
76              if (arr[i] > arr[i + step]){
77                  temp = arr[i];
78                  arr[i] = arr[i + step];
79                  arr[i + step] = temp;
80                  permutation++;
81              }
82              comparison++;
83          }
84          step /= factor;
85      }
86      cout<<"comparison: "<<comparison<<endl;
87      cout<<"permutation: "<<permutation<<endl;
88      return arr;
89  }
```

3.4.2 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

```
n = 100
8 50 74 59 31 73 45 79 24 10 41 66 93 43 88 4 28 30 13 70 58 61 34 100 17 36 98 27 68 11 80 22 94 37 86 46 29 92 95 2 54 9 69 91 25 97 23 67 78 99 82 14 15
    64 26 16 18 96 6 5 52 89 83 53 38 39 72 32 76 7 63 20 55 90 71 21 35 51 60 48 40 12 19 62 77 75 57 85 56 81 3 1 65 49 84 47 44 33 42 87

comparison: 1233
permutation: 247
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
```

Рисунок 3.2 – Сортування масиву на 1000 елементів

```
n = 1000
808 250 74 659 931 273 545 879 924 710 441 166 493 43 988 504 328 730 841 613 304 170 158 561 934 100 279 817 336 98 827 513 268 811 634 980 150 580 822 968
    673 394 337 486 746 229 92 195 358 2 154 709 945 669 491 125 197 531 904 723 667 550 25 802 854 978 409 299 982 636 14 866 815 64 537 426 670 116 95 630
    502 518 196 106 405 452 189 124 506 883 753 567 717 338 439 145 898 872 829 138 359 178 398 295 905 610 232 176 143 400 969 413 261 558 595 9 396 114 7
    963 943 366 83 853 768 696 713 672 902 591 832 739 58 617 791 641 680 973 99 96 320 455 224 290 761 906 127 507 814 771 239 221 845 367 535 227 395 364
    551 160 624 948 386 218 540 248 497 886 421 736 916 626 12 153 245 296 819 397 693 816 992 34 554 548 826 211 663 212 809 378 762 869 996 777 440 875 332
    557 302 873 985 756 790 408 16 194 770 681 456 856 964 503 677 109 639 998 652 850 204 732 532 15 420 776 10 181 930 55 738 546 318 526 201 257 565 598
    649 705 151 977 555 798 505 382 749 66 379 700 210 130 484 448 608 774 323 306 177 54 225 631 401 445 371 286 17 899 156 134 577 179 267 712 615 820 592
    721 763 198 32 589 590 874 878 305 255 961 481 731 956 547 108 1 927 36 858 115 594 361 355 419 586 564 103 918 644 748 270 474 104 460 407 825 133 974
    604 921 951 901 521 534 259 4 935 781 880 833 575 543 772 311 984 971 41 724 651 972 230 319 747 300 231 622 959 697 65 599 752 941 552 206 81 275 650
    414 321 26 13 784 789 118 325 258 512 691 411 566 743 404 353 308 142 911 800 128 172 788 415 642 349 843 316 975 446 374 912 240 254 190 167 357 861 593
    117 581 868 940 699 812 892 729 662 46 121 241 159 454 628 351 728 828 796 443 517 516 925 602 937 282 711 217 782 792 126 385 168 950 477 47 468 745 519
    719 690 919 29 30 35 755 541 936 84 49 253 877 238 676 246 416 193 38 716 694 165 801 90 671 847 39 544 339 322 720 360 236 913 685 990 785 735 668 520
    616 632 799 640 5 766 392 122 107 451 704 33 655 466 623 40 658 467 997 495 556 222 313 294 428 233 199 857 571 915 767 836 922 44 87 606 944 312 432 993
    837 834 329 684 490 435 27 523 365 891 885 442 180 538 871 562 582 803 733 333 855 760 848 463 11 942 113 957 391 45 102 962 511 929 806 174 97 422 301
    263 264 890 251 759 653 93 317 715 61 369 742 235 129 572 449 876 330 472 821 765 80 343 139 383 576 981 597 88 734 262 381 779 424 499 846 510 611 219
    291 56 744 276 152 252 438 101 285 68 141 708 563 970 314 979 508 657 787 740 585 515 20 664 553 289 164 380 462 488 354 900 881 284 237 910 568 835 111
    436 71 588 298 647 91 112 464 952 214 601 960 331 633 560 778 780 384 82 522 600 417 619 453 584 660 870 596 470 173 220 851 244 393 525 362 621 726 94
    123 469 24 794 342 356 903 348 947 852 524 737 867 926 140 695 161 293 277 465 485 120 860 146 399 144 487 514 967 192 175 751 528 1000 430 813 135 260
    754 226 859 492 769 688 527 473 205 475 345 849 618 839 509 807 200 256 830 489 897 865 19 479 309 324 678 954 272 476 271 266 280 76 185 157 665 887 48
    666 683 315 242 431 86 989 687 496 429 184 137 579 692 722 775 863 707 627 645 376 132 412 646 287 494 958 637 370 907 939 187 686 923 182 203 209 831
    938 920 909 578 882 31 326 437 995 946 603 234 457 459 202 444 406 119 344 213 933 783 917 265 155 28 953 387 423 818 215 888 682 559 434 21 805 908 698
    966 458 625 741 894 50 797 388 949 425 895 105 368 750 274 864 706 450 986 703 307 471 638 281 725 614 148 402 188 110 78 223 840 418 607 896 228 461 629
    727 297 62 53 8 377 498 823 278 501 60 85 569 914 583 171 136 89 346 757 310 23 149 375 186 57 163 363 862 433 773 609 37 675 928 810 482 976 987 884 42
    718 69 288 573 77 410 549 786 390 999 147 758 656 191 842 804 793 216 570 51 889 373 764 59 838 372 955 283 994 500 529 648 169 350 389 689 6 991 22 249
    3 208 303 674 327 73 536 965 542 844 483 605 70 824 679 243 427 612 269 292 539 661 67 79 480 347 334 893 247 587 52 72 654 932 643 620 530 702 447 335
    183 18 340 478 63 714 983 162 701 341 403 795 131 574 352 635 207 533 75

comparison: 22034
permutation: 4244
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
    447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484
    485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522
    523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560
    561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598
    599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636
    637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674
    675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712
    713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750
    751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788
    789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826
    827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864
    865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902
    903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940
    941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978
    979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000
```

3.5 Тестування алгоритму

3.5.1 Часові характеристики оцінювання

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

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

Розмірність масиву	Число порівнянь	Число перестановок
10	39	0
100	1223	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 гребінцем для зворотно упорядкованої послідовності елементів у масиві.

Розмірність масиву	Число порівнянь	Число перестановок
10	39	5
100	1223	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 гребінцем для масивів різної розмірності, масиви містять випадкову послідовність елементів.

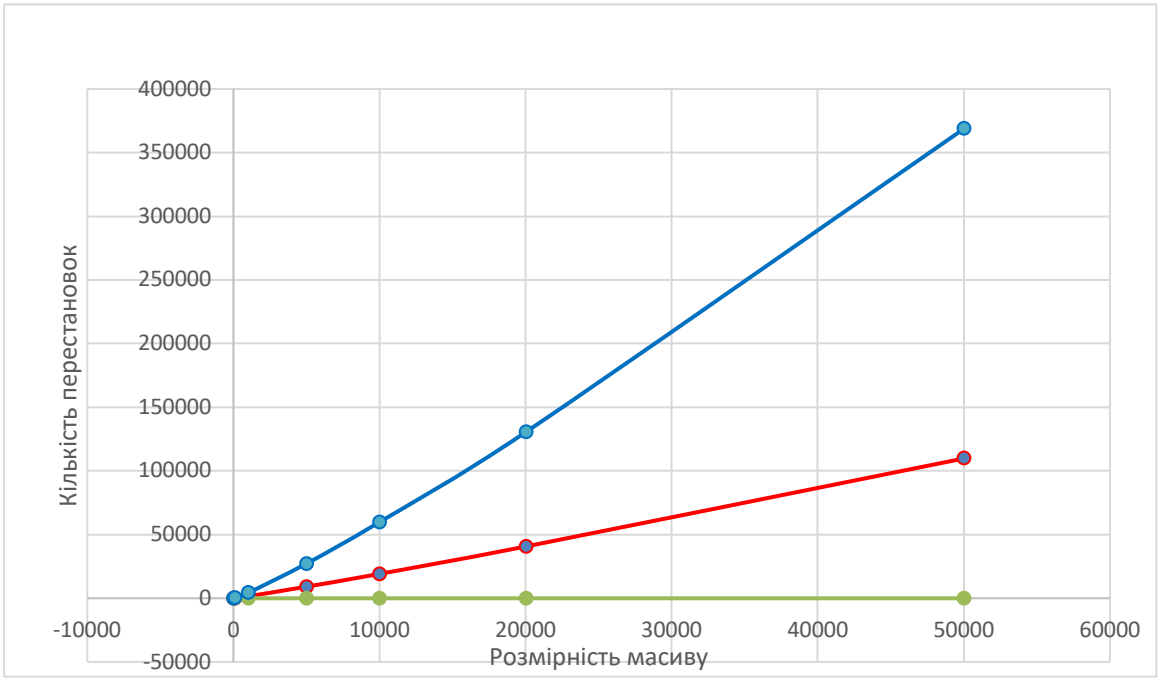
Таблиця 3.4 – Характеристика оцінювання алгоритму сортування гребінцем для випадкової послідовності елементів у масиві.

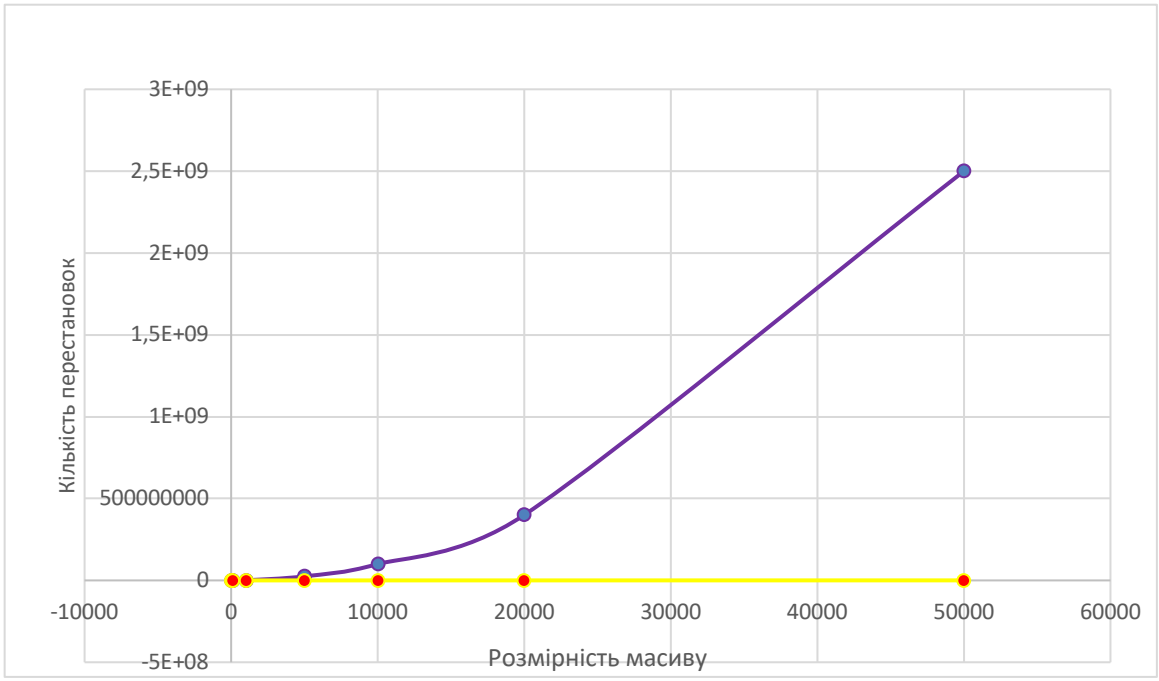
Розмірність масиву	Число порівнянь	Число перестановок
10	39	12
100	1223	247
1000	22034	4244
5000	144865	27235
10000	329653	59866
20000	719246	130411
50000	1997961	368711

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 було проаналізовано два алгоритму внутрішнього сортування: сортування бульбашкою і гребінцем.

Було оцінено їх часову складність для 3-х типів вхідних даних: упорядковані, зворотно упорядковані та випадкові дані. Для упорядкованої послідовності – найкращій результат. Для бульбашки найгірший випадок - зворотно упорядкована, для гребінця – з випадковими даними.

Отже, алгоритми є повністю стабільними для будь-якої кількості та упорядкованості даних.

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%.