



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра Інформаційної Безпеки

Практикум з Алгоритмів та структур даних
Лабораторна робота №2
МЕТОДИ СОРТУВАННЯ МАСИВІВ

Мета роботи:

познайомитися з роботою поширених методів сортування, з критеріями та методикою їх порівняння.

Перевірив:

Виконав:

студент II курсу

групи ФБ-01

Сахній Н.Р.

Виконання лабораторної роботи

а) ПРОСТІ МЕТОДИ СОРТУВАННЯ

Сортування вставками (Selection sort)

Програмний код:

```
# АСД -- Лабораторна робота № 2 -- (ФБ-01 Сахній Назар)
# SELECTION SORT
import time
from numpy.random import randint

comparisons = 0
moves = 0

def selection_sort(arr):
    global comparisons
    global moves

    for index in range(len(arr) - 1):
        min_value = index

        # Пошук індекса min елемента у невідсортованому масиві
        for search in range(index + 1, len(arr)):
            comparisons += 1
            if arr[search] < arr[min_value]:

                # Оновлюємо індекс min елемента
                min_value = search

        # Міняємо місцями з першим невідсортованим елементом
        arr[index], arr[min_value] = arr[min_value], arr[index]
        moves += 1
    return arr

ls100 = randint(-100, 100, 100)
ls1000 = randint(-1000, 1000, 1000)
ls10000 = randint(-10000, 10000, 10000)
print(f"\n\nДовільний числовий масив із 10000 елементів:\n {ls100}")
start = time.time()
print(f"\n\nВідсортований даний масив методом вибору:\n
{selection_sort(ls100)}"
      f"\n\nКількість ОБМІНІВ: {moves}, ПОРІВНЯНЬ: {comparisons}"
      f"\nЧас сортування {time.time() - start}")

print(f"\n\nДовільний числовий масив із 10000 елементів:\n {ls1000}")
start = time.time()
print(f"\n\nВідсортований даний масив методом вибору:\n
{selection_sort(ls1000)}"
      f"\n\nКількість ОБМІНІВ: {moves}, ПОРІВНЯНЬ: {comparisons}")
```

```

f"\nЧас сортування {time.time() - start}")

print(f"\n\nДовільний числовий масив із 10000 елементів:\n {ls10000}")
start = time.time()
print(f"\n\nВідсортований даний масив методом вибору:\n
{selection_sort(ls10000)}")
f"\n\nКількість ОБМІНІВ: {moves}, ПОРІВНЯНЬ: {comparisons}"
f"\nЧас сортування {time.time() - start}")

```

Результат виконання програмного коду:

```

Run: Selection_sort X
C:\Users\t-1000\PycharmProjects\ASD_Lab2\venv\Scripts\python.exe C:/Users/t-1000/PycharmProjects/ASD_Lab2/Selection_sort.py

Довільний числовий масив із 100 елементів:
[ 62 69 54 69 97 20 -88 43 -55 -39 -19 32 -61 90 72 90 -99 -83
-14 -21 -95 -64 43 -1 95 73 2 -88 51 -99 50 -29 -73 -23 10 -20
70 22 -95 79 77 -50 -90 31 -43 89 22 29 -7 -63 29 62 83 -58
-85 -49 78 -95 80 69 -17 63 -91 43 -30 -38 54 81 44 87 -59 -61
85 -15 -36 92 48 18 -95 -42 -40 -78 -81 79 -81 53 -34 -77 -81 -85
17 -97 -97 -99 -61 -26 4 78 8 -14]

Відсортований даний масив методом вибору:
[-99 -99 -99 -97 -97 -95 -95 -95 -95 -91 -90 -88 -88 -85 -85 -83 -81 -81
-81 -78 -77 -73 -64 -63 -61 -61 -61 -59 -58 -55 -50 -49 -43 -42 -40 -39
-38 -36 -34 -30 -29 -26 -23 -21 -20 -19 -17 -15 -14 -14 -7 -1 2 4
8 10 17 18 20 22 22 29 29 31 32 43 43 43 44 48 50 51
53 54 54 62 62 63 69 69 69 70 72 73 77 78 78 79 79 80
81 83 85 87 89 90 90 92 95 97]

Кількість ОБМІНІВ: 99, ПОРІВНЯНЬ: 4950
Час сортування 0.000982046127319336

Довільний числовий масив із 1000 елементів:
[-407 483 -543 605 -896 816 -694 613 -38 431 -683 568 -833 -891
-503 -922 -81 427 -265 -336 -141 -390 -523 -478 598 440 -298 491
614 361 671 957 276 601 -168 -700 -761 -278 257 -41 22 -424
603 146 789 302 820 542 -334 -585 177 -248 -874 361 778 -909
-684 -837 -898 -95 912 -478 839 -199 -478 688 817 787 580 539
518 218 554 997 -644 -885 796 546 -204 -370 -186 -437 -253 66
433 -549 -595 707 -274 -6 -67 122 858 -550 -63 278 558 -324
64 -581 -700 662 748 325 760 865 45 339 278 -92 -148 319
287 -855 -942 964 -747 -306 -523 -952 -559 937 752 562 377 -735
545 -176 -887 888 -775 -660 -522 802 2 -606 -277 580 885 771

```

```

Відсортований даний масив методом вибору:
[-995 -991 -987 -986 -983 -981 -974 -969 -968 -967 -967 -965 -965 -959
-959 -959 -956 -953 -953 -952 -950 -947 -945 -944 -942 -942 -940 -937
-936 -936 -934 -933 -927 -927 -925 -925 -922 -922 -916 -916 -912 -912
-911 -910 -909 -907 -906 -905 -903 -902 -898 -896 -896 -895 -893 -892
-891 -891 -888 -886 -886 -885 -885 -884 -879 -875 -874 -872 -870 -869
-867 -865 -859 -857 -855 -855 -854 -853 -852 -849 -848 -843 -840 -839
-837 -836 -834 -833 -832 -832 -831 -828 -825 -822 -822 -819 -816 -814
-814 -806 -804 -803 -802 -801 -801 -801 -796 -795 -794 -785 -785 -784
-784 -782 -777 -774 -773 -769 -764 -763 -761 -759 -759 -758 -758 -758
-754 -750 -750 -747 -743 -742 -738 -737 -736 -735 -733 -732 -732 -731
-729 -728 -728 -727 -716 -715 -708 -707 -703 -701 -700 -700 -697 -696
606 606 603 603 601 600 600 600 607 606 606 607 607 607

```

```
Кількість ОБМІНІВ: 1098, ПОРІВНЯНЬ: 504450
Час сортування 0.10799908638000488

Довільний числовий масив із 10000 елементів:
[ 3244 -5621 8281 ... -9658 2579 -6778]

Відсортований даний масив методом вибору:
[-10000 -10000 -9998 ... 9997 9998 9998]

Кількість ОБМІНІВ: 11097, ПОРІВНЯНЬ: 50499450
Час сортування 11.176067113876343

Process finished with exit code 0
```

6) МОДИФІКОВАНІ МЕТОДИ СОРТУВАННЯ

Метод Шелла (Shell sort)

Програмний код:

```
# АСД -- Лабораторна робота № 2 -- (ФБ-01 Сахній Назар)
# SHELL SORT
from math import log2
import time
from numpy.random import randint

comparisons = 0
moves = 0

def shell_sort(arr):
    global comparisons
    global moves

    distance = [1]
    for k in range(round(log2(len(arr)) - 1)):
        distance.append(2 * distance[-1] + 1)
    distance.reverse()

    for interval in distance:

        # Виконуємо сортування вставками відповідних підмасивів
        for i in range(interval, len(arr)):
            temp = arr[i]
            j = i
            comparisons += 1
            while j >= interval and arr[j - interval] > temp:
                comparisons += 1
                arr[j] = arr[j - interval]
                j -= interval
            arr[j] = temp
            moves += 1

    return arr
```

```

ls100 = randint(-100, 100, 100)
ls1000 = randint(-1000, 1000, 1000)
ls10000 = randint(-10000, 10000, 10000)
print(f"\n\nДовільний числовий масив із 100 елементів:\n {ls100}")
start = time.time()
print(f"\n\nВідсортований даний масив методом Шелла:\n
{shell_sort(ls100)}"
      f"\n\nКількість ОБМІНІВ: {moves}, ПОРІВНЯНЬ: {comparisons}"
      f"\nЧас сортування {time.time() - start}")

print(f"\n\nДовільний числовий масив із 1000 елементів:\n {ls1000}")
start = time.time()
print(f"\n\nВідсортований даний масив методом Шелла:\n
{shell_sort(ls1000)}"
      f"\n\nКількість ОБМІНІВ: {moves}, ПОРІВНЯНЬ: {comparisons}"
      f"\nЧас сортування {time.time() - start}")

print(f"\n\nДовільний числовий масив із 10000 елементів:\n {ls10000}")
start = time.time()
print(f"\n\nВідсортований даний масив методом вибору:\n
{shell_sort(ls10000)}"
      f"\n\nКількість ОБМІНІВ: {moves}, ПОРІВНЯНЬ: {comparisons}"
      f"\nЧас сортування {time.time() - start}")

```

Результат виконання програмного коду:

```

Run: Shell_sort X
C:\Users\t-1000\PycharmProjects\ASD_Lab2\venv\Scripts\python.exe C:/Users/t-1000/PycharmProjects/ASD_Lab2/Shell_sort.py

Довільний числовий масив із 100 елементів:
[ 59 -37 -27 -84 29 8 -24 8 -82 -21 -71 91 34 -26 -50 -62 56 -12
-53 56 30 60 65 -5 36 55 54 56 1 56 55 -78 -16 -48 -75 -48
-26 -78 -16 68 -31 -43 35 9 85 -68 5 25 -61 -50 84 61 -99 51
49 24 52 -81 80 11 -82 84 -12 -77 -42 -64 35 48 25 -24 -88 -32
18 -56 -29 56 78 -94 -45 -22 65 -84 58 71 7 14 -9 20 73 -21
-73 85 -70 98 79 -26 40 19 73 -27]

Відсортований даний масив методом Шелла:
[-99 -94 -88 -84 -84 -82 -82 -81 -78 -78 -77 -75 -73 -71 -70 -68 -64 -62
-61 -56 -53 -50 -50 -48 -48 -45 -43 -42 -37 -32 -31 -29 -27 -27 -26 -26
-26 -24 -24 -22 -21 -21 -16 -16 -12 -12 -9 -5 1 5 7 8 8 9
11 14 18 19 20 24 25 25 29 30 34 35 35 36 40 48 49 51
52 54 55 55 56 56 56 56 56 58 59 60 61 65 65 68 71 73
73 78 79 80 84 84 85 85 91 98]

Кількість ОБМІНІВ: 480, ПОРІВНЯНЬ: 838
Час сортування 0.0009989738464355469

```

```

Довільний числовий масив із 1000 елементів:
[ 704  511 -699 -395 -380 -394 -293  -19  853 -371  621 -885  -28 -142
 872  833  89 -358  103  122  544  724 -235  776  439 -175 -173 -449
-413 -323  760  591  14 -437 -515  422 -941  359  69 -260  780  709
 535  602 -324  479  152 -831  232  491  69 -994  707  709 -809  616
 625 -430 -636 -920  875  898  200  680 -845 -663  317 -346 -917 -686
 323  187  -50 -617  173  759  742  216  835  866  930 -277  -3  766

```

```

Відсортований даний масив методом Шелла:
[-999 -997 -994 -993 -991 -987 -986 -979 -978 -978 -977 -976 -975 -972
-971 -967 -967 -966 -955 -953 -951 -948 -947 -945 -941 -940 -939 -938
-935 -933 -933 -928 -920 -920 -920 -920 -917 -916 -915 -911 -910 -905 -904
-901 -901 -900 -898 -894 -892 -891 -885 -882 -879 -878 -874 -872 -870
-866 -859 -858 -848 -845 -844 -840 -839 -839 -835 -835 -831 -831 -830
-826 -826 -824 -824 -820 -819 -818 -817 -812 -809 -809 -805 -800 -799
 707  709  700  700  705  700  701  775  775  777  770  770  760  765

```

```

Кількість ОБМІНІВ: 8467, ПОРІВНЯНЬ: 15752
Час сортування 0.012955188751220703

```

```

Довільний числовий масив із 10000 елементів:
[ 6593  618 -7713 ... -5597 -3421 -9868]

```

```

Відсортований даний масив методом вибору:
[-10000 -9997 -9995 ... 9996 9998 9998]

```

```

Кількість ОБМІНІВ: 122098, ПОРІВНЯНЬ: 269596
Час сортування 0.09196710586547852

```

```

Process finished with exit code 0

```

в) АНАЛІЗ ВИКОНАНОЇ РОБОТИ

У сортуванні методом Шелла мною було використано послідовність **Кнута** для обрахунку відстані між елементами:

$d_{k-1} = 2d_k + 1$, де $d_k = 1$, а значення $k = \log_2 N - 1$ (N – к-ть елементів у списку)

Псевдокод для реалізації сортування методом Шелла:

```

Shell_Sort(A, N)
1. for k ← 1 to m
2. do for i ← d[k] + 1 to N
3.   do a ← A[i]
4.     j ← i
5.     while j - d[k] ≥ 1 i A[j - d[k]] > a
6.       do A[j] ← A[j - d[k]]
7.       j ← j - d[k]
8.     A[j] ← a

```

Основний принцип, на якому функціонує цей метод це сортування вставками, проте на відміну від того способу, де обирається послідовно кожен елемент і

порівнюється з попередніми, то в цьому способі виконується порівняння та перестановка тих елементів, що розташовані на різній відстані один від одного.

Ефективність досягається тим, що кожне наступне впорядкування вимагає меншої кількості перестановок, оскільки деякі елементи вже стали на свої місця.

Результати порівняння методів сортування										
	Сортування вибором					Метод Шелла				
N	К-ть копіювань (М)		К-ть порівнянь (С)		Час (Т)	К-ть копіювань (М)		К-ть порівнянь (С)		Час (Т)
	Теорет.	Експерим.	Теорет.	Експерим.		Теорет.	Експерим.	Теорет.	Експерим.	
100	50	99	4 950	4 950	0,001	251	480	251	838	0,00099
1000	500	1 098	499500	504 450	0,108	3 981	8 467	3 981	15 752	0,01296
10000	5000	11 097	49 995000	50 499450	11,176	63 096	122 098	63 096	269 596	0,09197

- Оцінювання якісних (теоретичних) характеристик програмного коду сортування вибором здійснювалося за наступними формулами:

$$M = \frac{n - 1}{2} \quad C = \frac{(n - 1)n}{2}$$

- Для методу Шелла відома лише порядкова оцінка:

$$M = C = O(n^{1,2}), \text{ де } n - \text{кількість елементів у списку}$$