

## ЛАБОРАТОРНА РОБОТА № 13-14

### Сортування одновимірного масиву

**Мета:** набуття практичних навиків роботи з одновимірними масивами, а саме сортування елементів масиву різними методами. Здійснення порівняння та аналізу ефективності використовуваних методів сортування.

#### Хід роботи:

Завдання 1. Згенеруйте одновимірний масив і виведіть його на екран.

Лістинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <windows.h>
#define N 100

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(NULL));
    int arr[N], n;
    printf("Введіть кількість елементів масиву з діапазону [1; 100]: ");
    scanf_s("%d", &n);
    printf("arr = {");
    for (int i = 0; i < n; i++) {
        arr[i] = rand() % 21 - 10;
        printf("%d", arr[i]);
        if (i != n - 1) printf(", ");
    }
    printf("}\n");
    return 0;
}
```

Результат виконання програми:

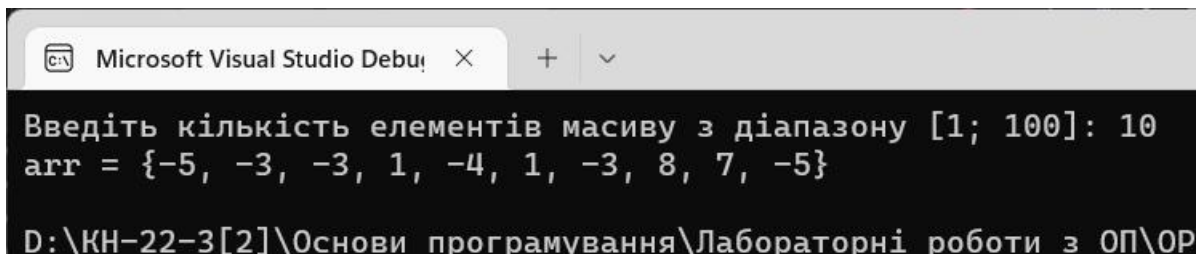


Рис. 1. Результат виконання першої програми

					ДУ «Житомирська політехніка».22.122.13.000 – Лр13-14			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Черниш М.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Терещук С.О.						Аркушів
Керівник								1
Н. контр.								11
Зав. каф.							ФІКТ Гр. КН-22-3[2]	

Завдання 2. Відсортуйте масив кожним описаним методом.

### 1. Сортування обміном.

Лістинг програми:

```
case 1:
    do {
        fl = 0;
        for (int i = 1; i < n; i++)
            if (arr[i - 1] > arr[i]) {
                tmp = arr[i];
                arr[i] = arr[i - 1];
                arr[i - 1] = tmp;
                fl = 1;
            }
    } while (fl); break;
```

Результат виконання програми:



Рис. 2. Результат виконання сортування методом обміну

### 2. Сортування методом вибору.

Лістинг програми:

```
case 2:
    for (int i = 0; i < n - 1; i++) {
        imin = i;
        for (int j = i + 1; j < n; j++)
            if (arr[j] < arr[imin]) imin = j;
        tmp = arr[i];
        arr[i] = arr[imin];
        arr[imin] = tmp;
    } break;
```

## Результат виконання програми:



Рис. 3. Результат виконання сортування методом вибору

## 3. Сортування вставками.

### Лістинг програми:

```
case 3:
    for (int i = 1; i < n; i++) {
        tmp = arr[i];
        for (int j = i - 1; j >= 0 && arr[j] > tmp; j--) {
            arr[j + 1] = arr[j];
            arr[j] = tmp;
        }
        break;
    }
```

### Результат виконання програми:



Рис. 4. Результат виконання сортування вставками

#### 4. Сортвання методом Шелла.

Лістинг програми:

```
case 4:
    step = size / 2;
    while (step > 0) {
        for (int i = 0; i < (size - step); i++) {
            int j = i;
            while (j >= 0 && arr[j] > arr[j + step]) {
                tmp = arr[j];
                arr[j] = arr[j + step];
                arr[j + step] = tmp;
                j--;
            }
        }
        step = step / 2;
    } break;
```

Результат виконання програми:



Рис. 5. Результат виконання сортування методом Шелла

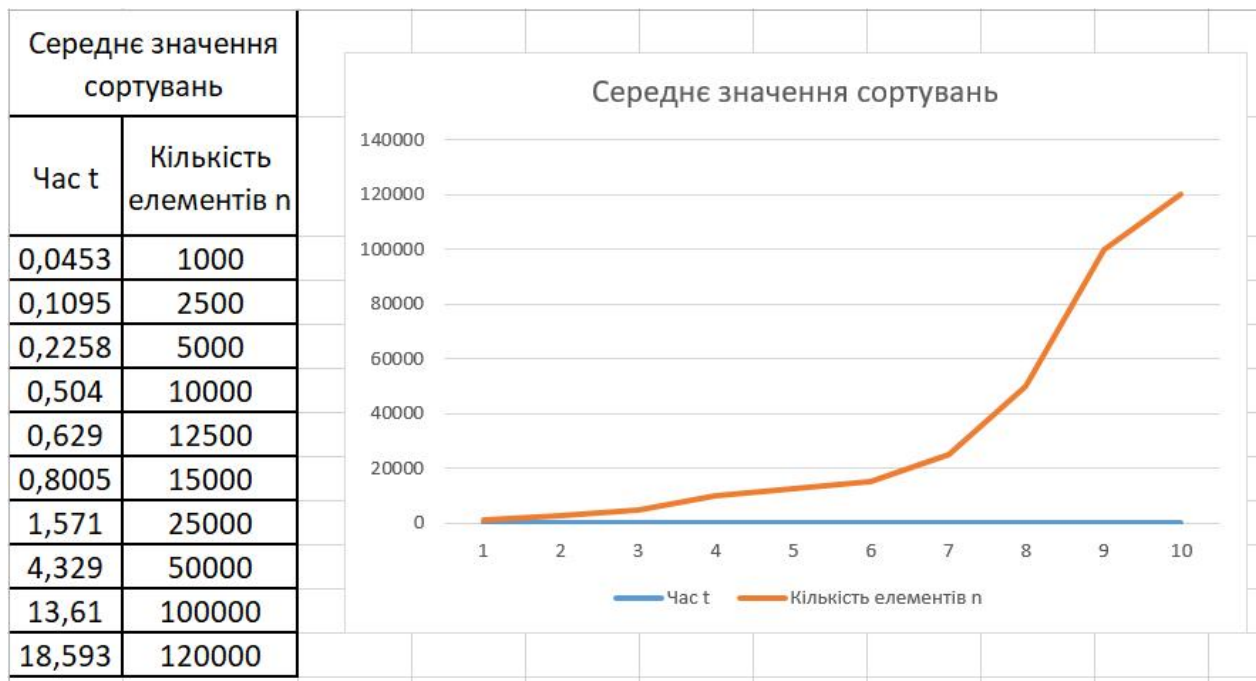


Рис. 6. Графік середнього значення сортувань

Отже, порівнявши час виконання обраних методів сортування, зроблено висновок, що сортування методом Шелла дає найкращий результат для великих масивів. А найгіршим виявився метод сортування обміном.

Завдання на самостійну роботу:

Завдання 1. Реалізувати методи сортування обміном, вибору, вставками та Шелла для масивів дійсних типів даних за спаданням, та порівняти їх швидкодію.

Лістинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <windows.h>
#define N 120000

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(NULL));
    float arr[N], tmp, dd;
    int n, m, fl, imax, step;
    printf("Введіть кількість елементів масиву з діапазону [1; 100]: ");
    scanf_s("%d", &n);
    int size = n;
    printf("arr = {");
    for (int i = 0; i < n; i++) {
        arr[i] = -10 + (float)rand() * (10 - (-10)) / RAND_MAX;
        printf("%.2f", arr[i]);
    }
}
```

```

        if (i != n - 1) printf(", ");
    }
    printf("}\n");
    printf("\nВиберіть метод сортування: 1 - Сортування обміном. 2 - Сортування методом
    вибору.\n3 - Сортування вставками. 4 - Сортування методом Шелла: ");
    scanf_s("%d", &m);
    clock_t start, finish;
    start = clock();        // час початку виконання програми
    switch (m) {
    case 1:
        do {
            fl = 0;
            for (int i = 1; i < n; i++)
                if (arr[i - 1] < arr[i]) {
                    tmp = arr[i];
                    arr[i] = arr[i - 1];
                    arr[i - 1] = tmp;
                    fl = 1;
                }
        } while (fl); break;
    case 2:
        for (int i = 0; i < n - 1; i++) {
            imax = i;
            for (int j = i + 1; j < n; j++)
                if (arr[j] > arr[imax]) imax = j;
            tmp = arr[i];
            arr[i] = arr[imax];
            arr[imax] = tmp;
        } break;
    case 3:
        for (int i = 1; i < n; i++) {
            tmp = arr[i];
            for (int j = i - 1; j >= 0 && arr[j] < tmp; j--) {
                arr[j + 1] = arr[j];
                arr[j] = tmp;
            }
        } break;
    case 4:
        step = size / 2;
        while (step > 0) {
            for (int i = 0; i < (size - step); i++) {
                int j = i;
                while (j >= 0 && arr[j] < arr[j + step]) {
                    tmp = arr[j];
                    arr[j] = arr[j + step];
                    arr[j + step] = tmp;
                    j--;
                }
            }
            step = step / 2;
        } break;
    default: printf("Введене число не лежить в діапазоні [1; 4]\n");
    }
    printf("\narr = {");
    for (int i = 0; i < n; i++) {
        printf("%.2f", arr[i]);
        if (i != n - 1) printf(", ");
    }
    printf("}\n");
    finish = clock();        // час завершення виконання програми
    dd = (float)(finish - start) / CLOCKS_PER_SEC;
    // час виконання програми
    printf("\nЧас виконання обраного методу сортування = %.3f\n", dd);
    return 0;

```

		Черниш М.			ДУ «Житомирська політехніка».22.122.13.000 – Лр13-14	Арк.
		Герещук С.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

}

Результат виконання програми:

```

Microsoft Visual Studio Debug Console
Введіть кількість елементів масиву з діапазону [1; 100]: 50
arr = {-1.75, 1.84, -3.20, -7.49, 0.17, -6.95, -3.79, 3.30, -3.74, -7.06, -8.38, -6.88,
-5.22, 4.97, -2.10, -8.40, -5.92, 2.54, -0.40, 0.81, -9.04, -3.71, -8.39, 5.63, -8.03,
-7.19, 5.53, 7.25, -4.38, 0.39, -5.18, 0.16, 3.78, 7.54, 1.89, 3.48, -9.85, 9.76, 0.14,
7.16, 4.28, -9.70, -5.71, 3.69, -1.10, -9.53, -8.10, 5.90, -1.07, -4.84}

Виберіть метод сортування: 1 - Сортування обміном. 2 - Сортування методом вибору.
3 - Сортування вставками. 4 - Сортування методом Шелла: 4

arr = {9.76, 7.54, 7.25, 7.16, 5.90, 5.63, 5.53, 4.97, 4.28, 3.78, 3.69, 3.48, 3.30, 2.
54, 1.89, 1.84, 0.81, 0.39, 0.17, 0.16, 0.14, -0.40, -1.07, -1.10, -1.75, -2.10, -3.20,
-3.71, -3.74, -3.79, -4.38, -4.84, -5.18, -5.22, -5.71, -5.92, -6.88, -6.95, -7.06, -7
.19, -7.49, -8.03, -8.10, -8.38, -8.39, -8.40, -9.04, -9.53, -9.70, -9.85}

Час виконання обраного методу сортування = 0.005

```

Рис. 10. Результат виконання першої програми самостійної роботи

Отже, сортування методом Шелла дає найкращий результат для великих масивів як дійсних так і цілих типів даних. А найгіршим знову виявився метод сортування обміном.

Завдання 2. \*Дано масив із 16 чисел.

- Розписати покроково результат пірамідального сортування і подати у вигляді таблиці.
- Побудувати дерево.

Таблиця 1. Покрокові результати пірамідального сортування

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Масив до сортування	10	-2	3	9	7	6	2	-10	9	10	4	-5	3	-4	-3	-2
1	10	-2	3	9	7	6	2	-2	9	10	4	-5	3	-4	-3	-10
2	10	-2	3	9	10	6	2	-2	9	7	4	-5	3	-4	-3	-10
3	10	-2	6	9	10	3	2	-2	9	7	4	-5	3	-4	-3	-10
4	10	10	6	9	7	3	2	-2	9	-2	4	-5	3	-4	-3	-10





34	-4	-2	2	-3	-2	-5	-10	3	3	4	6	7	9	9	10	10
35	2	-2	-4	-3	-2	-5	-10	3	3	4	6	7	9	9	10	10
36	-10	-2	-4	-3	-2	-5	2	3	3	4	6	7	9	9	10	10
37	-2	-10	-4	-3	-2	-5	2	3	3	4	6	7	9	9	10	10
38	-2	-2	-4	-3	-10	-5	2	3	3	4	6	7	9	9	10	10
39	-5	-2	-4	-3	-10	-2	2	3	3	4	6	7	9	9	10	10
40	-2	-5	-4	-3	-10	-2	2	3	3	4	6	7	9	9	10	10
41	-2	-3	-4	-5	-10	-2	2	3	3	4	6	7	9	9	10	10
42	-10	-3	-4	-5	-2	-2	2	3	3	4	6	7	9	9	10	10
43	-3	-10	-4	-5	-2	-2	2	3	3	4	6	7	9	9	10	10
44	-3	-5	-4	-10	-2	-2	2	3	3	4	6	7	9	9	10	10
45	-10	-5	-4	-3	-2	-2	2	3	3	4	6	7	9	9	10	10
46	-10	-5	-4	-3	-2	-2	2	3	3	4	6	7	9	9	10	10

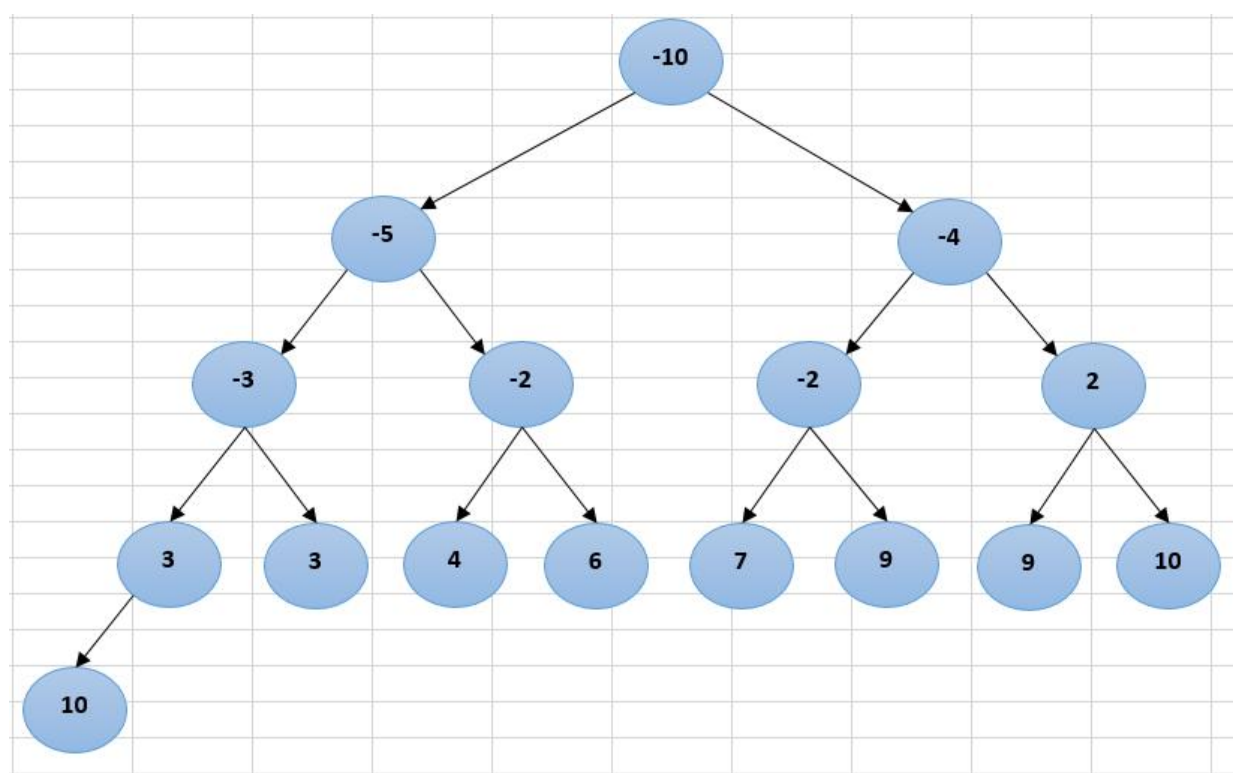


Рис. 11. Відсортоване дерево

### Завдання 3. \*Реалізувати метод пірамідального сортування на мові програмування Сі.

Лістинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <windows.h>
#include <utility>
using namespace std;
#define N 100

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(NULL));
    int arr[N], n, father, maxson, last_n;
    printf("Введіть кількість елементів масиву з діапазону [1; 100]: ");
    scanf_s("%d", &n);
    printf("\narr = {");
    for (int i = 1; i <= n; i++) {
        arr[i] = rand() % 21 - 10;
        printf("%d", arr[i]);
        if (i != (n + 1) - 1) printf(", ");
    }
    printf("}\n");
    for (int j = n / 2; j >= 1; j--) {
        father = j;
        while (father <= n / 2) {
            maxson = 2 * father;
            if (2 * father + 1 <= n && arr[2 * father] < arr[2 * father + 1])
                maxson = 2 * father + 1;
            if (arr[maxson] > arr[father]) {
                swap(arr[maxson], arr[father]);
                father = maxson;
            }
            else break;
        }
    }
    for (int i = n; i > 1; i--) {
        swap(arr[1], arr[i]);
        last_n = i - 1;
        father = 1;
        while (father <= last_n / 2) {
            maxson = 2 * father;
            if (2 * father + 1 <= last_n && arr[2 * father] < arr[2 * father + 1])
                maxson = 2 * father + 1;
            if (arr[maxson] > arr[father]) {
                swap(arr[maxson], arr[father]);
                father = maxson;
            }
            else break;
        }
    }
    printf("\narr = {");
    for (int i = 1; i <= n; i++) {
        printf("%d", arr[i]);
        if (i != (n + 1) - 1) printf(", ");
    }
```

		Черниш М.			ДУ «Житомирська політехніка». 22.122.13.000 – Лр13-14	Арк.
		Герещук С.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    printf("{}\n");
    return 0;
}

```

Результат виконання програми:

```

Введіть кількість елементів масиву з діапазону [1; 100]: 16
arr = {2, 4, -4, -3, 5, -1, -2, 4, -2, -5, 6, 4, -6, -4, -9, 2}
arr = {-9, -6, -5, -4, -4, -3, -2, -2, -1, 2, 2, 4, 4, 4, 5, 6}

```

Рис. 12. Результат виконання другої програми самостійної роботи

Завдання 4. Визначити яка схема належить якому методу сортування.

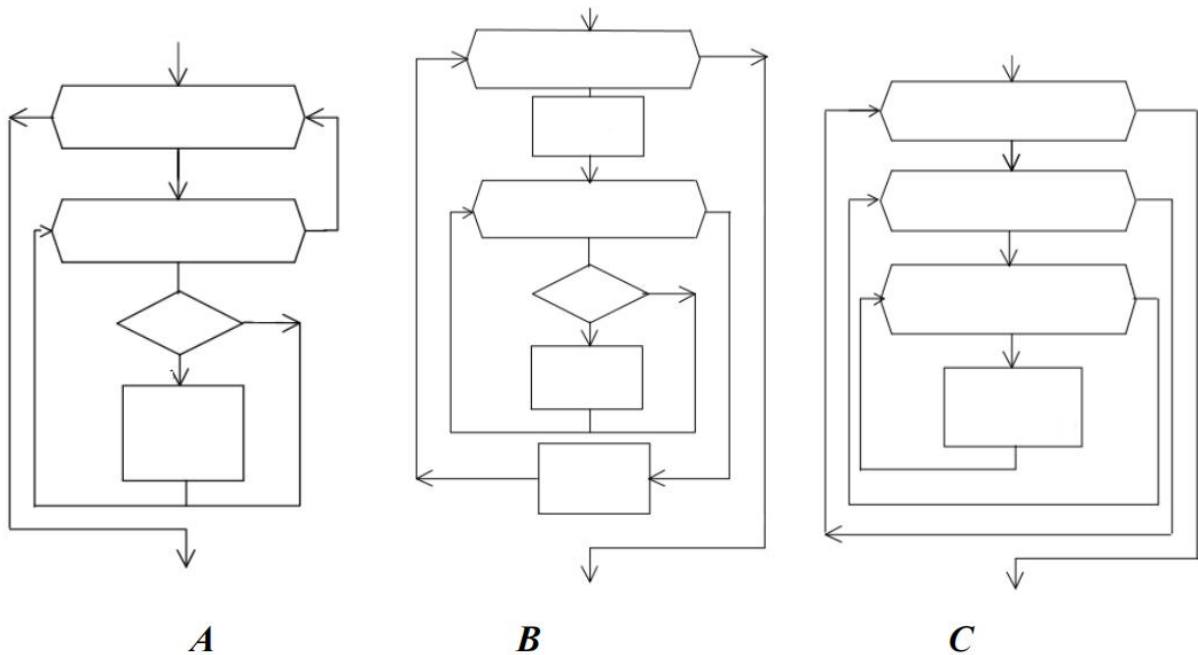


Рис. 13. Схеми різних методів сортування

- А) Метод сортування вставками.
- В) Сортування методом вибору.
- С) Сортування методом Шелла.