

## Лабораторна робота №5

Тема: Реалізація внутрішнього сортування масивів методами швидкого сортування, Шелла та злиття.

Мета роботи: вивчення методів сортування масиву з кращою швидкістю, придбання навичок їх програмування.

## Хід роботи

## Завдання для виконання

Варіант	Вигляд масиву	Характеристика масиву	Блок – схема методу
20	Масив з 30 дійсних чисел в інтервалі (-20..20)	Знайти кількість елементів у інтервалі від -13 до 13	Вибору

1. Згенерувати масив випадкових чисел за своїм варіантом лабораторної №4.

## Лістинг 5.1 – Код програми

```
// Чавурська Єлизавета, П-31, АіСД, Варіант 20, Лр 5
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    const int N = 30;
    setlocale(LC_CTYPE, "Ukr");
    double arr[N] = {
        -15.3, 12.7, 8.1, -4.6, 19.9, -20.0, 5.5, -11.2, 2.4, 17.3,
        -3.5, 14.6, -8.8, 9.2, -19.4, 6.7, 0.0, 11.1, -12.3, 13.5,
        -9.9, 4.3, -7.2, 10.4, -18.6, 3.6, 7.5, -1.9, 15.0, -14.8
    };
    cout << fixed << setprecision(2);
    cout << "Початковий масив:\n";
    for (int i = 0; i < N; i++) {
        cout << setw(7) << arr[i];
        if ((i + 1) % 10 == 0) cout << endl;
    }
    // 1 Сортування методом вибору
    for (int i = 0; i < N - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < N; j++) {
            if (arr[j] < arr[minIndex])
                minIndex = j;
        }
        // Обмін елементів
    }
}
```

```
        double temp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = temp;
    }
    cout << "\nВідсортований масив:\n";
    for (int i = 0; i < N; i++) {
        cout << setw(7) << arr[i];
        if ((i + 1) % 10 == 0) cout << endl;
    }
    // 2 Підрахунок елементів у діапазоні [-13; 13]
    int count = 0;
    for (int i = 0; i < N; i++) {
        if (arr[i] >= -13 && arr[i] <= 13)
            count++;
    }
    cout << "\nКількість елементів у інтервалі [-13; 13]: " << count <<
endl;
    cout << "Чавурська Єлизавета, П-31, AiCD, Варіант 20, Лр 4";
    return 0;
}
```

Початковий масив:

-15.30	12.70	8.10	-4.60	19.90	-20.00	5.50	-11.20	2.40	17.30
-3.50	14.60	-8.80	9.20	-19.40	6.70	0.00	11.10	-12.30	13.50
-9.90	4.30	-7.20	10.40	-18.60	3.60	7.50	-1.90	15.00	-14.80

Відсортований масив:

-20.00	-19.40	-18.60	-15.30	-14.80	-12.30	-11.20	-9.90	-8.80	-7.20
-4.60	-3.50	-1.90	0.00	2.40	3.60	4.30	5.50	6.70	7.50
8.10	9.20	10.40	11.10	12.70	13.50	14.60	15.00	17.30	19.90

Кількість елементів у інтервалі [-13; 13]: 20  
Чавурська Єлизавета, П-31, AiCD, Варіант 20, Лр 5

Рисунок 5.1 – Вікно результатів програми

2. Написати програму сортування масиву методом швидкого сортування.

Лістинг 5.2 – Код програми

```
// Чавурська Єлизавета, П-31, AiCD, Варіант 20, Лр 5
#include <iostream>
#include <iomanip>
using namespace std;
// Функція обміну елементів
void s(double &a, double &b) {
    double temp = a;
    a = b;
    b = temp;
}
int p(double arr[], int low, int high) {
```

```
double l = arr[high];
int i = low - 1;
for (int j = low; j < high; j++) {
    if (arr[j] < l) {
        i++;
        s(arr[i], arr[j]);
    }
}
s(arr[i + 1], arr[high]);
return (i + 1);
}
// Рекурсивна функція швидкого сортування
void k(double arr[], int low, int high) {
    if (low < high) {
        int pi = p(arr, low, high);
        k(arr, low, pi - 1);
        k(arr, pi + 1, high);
    }
}
int main() {
    setlocale(LC_CTYPE, "Ukr");
    const int N = 30;
    double arr[N] = {
        -15.3, 12.7, 8.1, -4.6, 19.9, -20.0, 5.5, -11.2, 2.4, 17.3,
        -3.5, 14.6, -8.8, 9.2, -19.4, 6.7, 0.0, 11.1, -12.3, 13.5,
        -9.9, 4.3, -7.2, 10.4, -18.6, 3.6, 7.5, -1.9, 15.0, -14.8
    };
    cout << fixed << setprecision(2);
    cout << "Початковий масив:\n";
    for (int i = 0; i < N; i++) {
        cout << setw(7) << arr[i];
        if ((i + 1) % 10 == 0) cout << endl;
    }
    // Виклик швидкого сортування
    k(arr, 0, N - 1);
    cout << "\nВідсортований масив:\n";
    for (int i = 0; i < N; i++) {
        cout << setw(7) << arr[i];
        if ((i + 1) % 10 == 0) cout << endl;
    }
    int count = 0;
    for (int i = 0; i < N; i++) {
        if (arr[i] >= -13 && arr[i] <= 13)
            count++;
    }
    cout << "\nКількість елементів у інтервалі [-13; 13]: " << count <<
endl;
    cout << "Чавурська Єлизавета, П-31, AiCD, Варіант 20, Лр 5";
    return 0;
}
```

```
Початковий масив:
-15.30  12.70   8.10  -4.60  19.90 -20.00   5.50 -11.20   2.40  17.30
 -3.50  14.60  -8.80   9.20 -19.40   6.70   0.00  11.10 -12.30  13.50
 -9.90   4.30  -7.20  10.40 -18.60   3.60   7.50  -1.90  15.00 -14.80

В?дсортований масив:
-20.00 -19.40 -18.60 -15.30 -14.80 -12.30 -11.20  -9.90  -8.80  -7.20
 -4.60  -3.50  -1.90   0.00   2.40   3.60   4.30   5.50   6.70   7.50
   8.10   9.20  10.40  11.10  12.70  13.50  14.60  15.00  17.30  19.90

К?льк?сть елемент?в у ?нтервал? [-13; 13]: 20
Чавурська Єлизавета, П-31, А?СД, Вар?ант 20, Лр 5
```

Рисунок 5.2 – Вікно результатів програми

3. Написати програму сортування масиву методом злиття.

Лістинг 5.4 – Код програми

```
// Чавурська Єлизавета, П-31, AiCD, Варіант 20, Лр 5
#include <iostream>
using namespace std;
// Функція злиття двох підмасивів
void merge(int arr[], int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;
    int L[50], R[50];
    for (int i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];
    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        }
        else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}
```

```
        k++;
    }
}
void func(int arr[], int left, int right) {
    if (left < right) {
        int mid = (left + right) / 2;
        func(arr, left, mid);
        func(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}
int main() {
    setlocale(LC_CTYPE, "Ukr");
    const int n = 30;
    int arr[n] = {
        -15, 10, 3, -7, 20, -19, 12, -5, 0, 8,
        17, -10, -3, 5, -17, 7, -1, 13, -9, 2,
        -11, 15, -8, 9, -6, 11, -12, 18, -4, 4
    };
    cout << "Початковий масив:\n";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
    // Сортування
    func(arr, 0, n - 1);

    cout << "\nВідсортований масив:\n";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
    // Підрахунок кількості елементів у межах [-13; 13]
    int count = 0;
    for (int i = 0; i < n; i++) {
        if (arr[i] >= -13 && arr[i] <= 13)
            count++;
    }
    cout << "\nКількість елементів у інтервалі [-13; 13]: " << count <<
endl;
    cout << "Чавурська Єлизавета, П-31, АіСД, Варіант 20, Лр 5";
    return 0;
}
```

```
Початковий масив:
-15 10 3 -7 20 -19 12 -5 0 8 17 -10 -3 5 -17 7 -1 13 -9 2 -11 15 -8
9 -6 11 -12 18 -4 4

Відсортований масив:
-19 -17 -15 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -1 0 2 3 4 5 7 8 9 10 1
1 12 13 15 17 18 20

Кількість елементів у інтервалі [-13; 13]: 23
Чавурська Єлизавета, П-31, AiCD, Варіант 20, Лр 5
```

Рисунок 5.3 – Вікно результатів програми

4. Написати програму сортування масиву методом Шелла.

Лістинг 5.4 – Код програми

```
// Чавурська Єлизавета, П-31, AiCD, Варіант 20, Лр 5
#include <iostream>
using namespace std;
void func(int arr[], int n) {
    for (int gap = n / 2; gap > 0; gap /= 2) {
        for (int i = gap; i < n; i++) {
            int temp = arr[i];
            int j = i;
            while (j >= gap && arr[j - gap] > temp) {
                arr[j] = arr[j - gap];
                j -= gap;
            }
            arr[j] = temp;
        }
    }
}
int main() {
    setlocale(LC_CTYPE, "Ukr");
    const int n = 30;
    int arr[n] = {
        -15, 10, 3, -7, 20, -19, 12, -5, 0, 8,
        17, -10, -3, 5, -17, 7, -1, 13, -9, 2,
        -11, 15, -8, 9, -6, 11, -12, 18, -4, 4
    };
    cout << "Початковий масив:\n";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
    func(arr, n);
    cout << "\nВідсортований масив (метод Шелла):\n";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
    cout << "Чавурська Єлизавета, П-31, AiCD, Варіант 20, Лр 5";
}
```

```

    return 0;
}

```

```

Початковий масив:
-15 10 3 -7 20 -19 12 -5 0 8 17 -10 -3 5 -17 7 -1 13 -9 2 -11 15 -8 9 -6
11 -12 18 -4 4

Відсортований масив (метод Шелла):
-19 -17 -15 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -1 0 2 3 4 5 7 8 9 10 11 12
13 15 17 18 20
Чавурська Єлизавета, П-31, А?СД, Вар?ант 20, Лр 5

```

Рисунок 5.4 – Вікно результатів програми

5. Записати нижню, верхню та асимптотичну оцінку складності кожного методу.

№	Метод сортування	Нижня оцінка $\Omega(n)$	Верхня оцінка $O(n)$	Асимптотична (середня) $\Theta(n)$
1	Метод бульбашки (Bubble Sort)	$\Omega(n)$	$O(n^2)$	$\Theta(n^2)$
2	Метод швидкого сортування (Quick Sort)	$\Omega(n \log n)$	$O(n^2)$	$\Theta(n \log n)$
3	Метод злиття (Merge Sort)	$\Omega(n \log n)$	$O(n \log n)$	$\Theta(n \log n)$
4	Метод Шелла (Shell Sort)	$\Omega(n \log n)$	$O(n^2)$	$\Theta(n^{1.5})$

6. Вивести вхідні дані і результати роботи на екран.
7. Зробити висновок про роботу та записати у звіт.

Висновок: на цій лабораторній роботі я вивчила методи сортування масиву з кращою швидкістю, придбала навичок їх програмування.