

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Львівський національний університет імені Івана Франка

Факультет електроніки та комп'ютерних технологій

ЗВІТ

про виконання ЛАБОРАТОРНОЇ РОБОТИ № 1

з дисципліни “Алгоритми і структури даних”

на тему “Сортування злиттям”

Виконав

студент 2 курсу

групи ФЕП-23

Чепара Станіслав Богданович

Перевірив

доцент кафедри РКТ, к.ф.-м.н.

Середницька Христина Ігорівна

Львів 2025

Мета роботи

Ознайомитися з методом сортування злиттям, реалізувати його в середовищі C++ у вигляді окремих функцій **Merge** та **MergeSort**, відпрацювати навички створення проекту з підключенням власної бібліотеки.

Теоретичні відомості

Сортування злиттям (Merge Sort) – це алгоритм «розділяй і володарюй».
Ідея полягає в тому, щоб:

1. Рекурсивно поділити масив на дві половини, доки не залишаться відрізки довжиною 1 елемент.
 2. Послідовно зливати відсортовані половини в більші масиви, отримуючи впорядкований результат.
-

3. Хід виконання роботи

3.1. Створено бібліотеку Sort із трьома файлами:

- **Sort.h** – оголошення функцій.
 - **Sort.cpp** – реалізація функцій **Merge** та **MergeSort**.
 - **main.cpp** – головна програма для зчитування даних та демонстрації роботи алгоритму.
-

3.2. Код програми ↓

Sort.cpp

```
1 #include "Sort.h"
2 #include <climits> // для INT_MAX
3
4 void Merge(std::vector<int>& a, int left, int mid, int right) {
5     // копіюємо ліву і праву частини в допоміжні вектори
6     std::vector<int> L;
7     std::vector<int> R;
8
9     for (int i = left; i <= mid; ++i) L.push_back(a[i]);
10    for (int j = mid + 1; j <= right; ++j) R.push_back(a[j]);
11
12    // додаємо «некінченність» в кінці – це сторожовий елемент
13    L.push_back(INT_MAX);
14    R.push_back(INT_MAX);
15
16    // індекси по допоміжних масивах
17    int i = 0;
18    int j = 0;
19
20    // зливаємо назад у a[left..right]
21    for (int k = left; k <= right; ++k) {
22        if (L[i] <= R[j]) {
23            a[k] = L[i];
24            ++i;
25        } else {
26            a[k] = R[j];
27            ++j;
28        }
29    }
30 }
31
32 void MergeSort(std::vector<int>& a, int left, int right) {
33     if (left >= right) return; // база: 0 або 1 елемент – вже відсортовано
34     int mid = left + (right - left) / 2;
35
36     // сортуємо ліву і праву половини
37     MergeSort(a, left, mid);
38     MergeSort(a, mid + 1, right);
39
40     // зливаємо дві відсортовані частини
41     Merge(a, left, mid, right);
42 }
```

Sort.h

```
1 #pragma once
2 #include <vector>
3
4 // Зливає дві відсортовані частини масиву a: [left..mid] і [mid+1..right]
5 void Merge(std::vector<int>& a, int left, int mid, int right);
6
7 // Рекурсивне сортування злиттям діапазону [left..right]
8 void MergeSort(std::vector<int>& a, int left, int right);
```

main.cpp

```

1 #include <iostream>
2 #include <vector>
3 #include "Sort.h"
4 using namespace std;
5
6 int main() {
7     //ios::sync_with_stdio(false);
8     //cin.tie(nullptr);
9
10    int n;
11    cout << "Введіть кількість елементів: ";
12    if (!(cin >> n) || n < 0) {
13        cout << "Некоректне n\n";
14        return 0;
15    }
16
17    cout << "Введіть " << n << " цілих чисел: ";
18    vector<int> a(n);
19    for (int i = 0; i < n; ++i) cin >> a[i];
20
21    // сортуємо
22    if (n > 0) MergeSort(a, 0, n - 1);
23
24    cout << "Відсортований масив: ";
25    for (int i = 0; i < n; ++i) {
26        cout << a[i];
27        if (i + 1 < n) cout << " ";
28    }
29    cout << "\n";
30
31    return 0;
32 }

```

3.3. Приклад роботи програми

```

• stanislav@fedora:~/Desktop/Algoritms&Data_structure/Lab1/build$ ./Lab1
Введіть кількість елементів: 5
Введіть 5 цілих чисел: 3
4
5
0
12
Відсортований масив: 0 3 4 5 12

```

4. Висновки

У даній лабораторній роботі я ознайомився з алгоритмом сортування злиттям та реалізував його на мові C++.

Було створено бібліотеку **Sort** з двома функціями: **Merge** та **MergeSort**.

Сортування злиттям є універсальним і надійним методом впорядкування, який використовується навіть у промислових сортувальниках.