**Львівський Національний Університет імені ІВАНА ФРАНКА**

Факультет прикладної математики та інформатики

**Алгоритми і Структури Даних**

**ЛАБОРАТОРНА РОБОТА №1(2025)**

**Алгоритми сортування(Counting Sort)**

Виконав:

**Заяц Ростислав Васильович**

Кафедра Програмування. Група ПМО-11с

Оцінка:

Перевірив:

**Тема:** Алгоритми сортування(Counting Sort).

**Мета:** Дослідити і навчитись використовувати алгоритм сортування Counting Sort.

**Counting Sort**

Сортування підрахунком – це алгоритм сортування, який підходить для сортування виключно цілих чисел у певному діапазоні. Працює він так:

1. Виділяємо додаткову пам'ять для створення контейнеру, який буде містити всі можливі значення в діапазоні.
2. Підраховуємо як часто в початковому списку трапляються певні значення і зберігаємо їх в допоміжний контейнер.
3. За допомогою допоміжного контейнера заповнюємо список вже відсортованими значеннями.

**Реалізація.**

1. Оголошуємо функцію-реалізацію нашого алгоритму.

Функція приймає два значення:

Int \* arr початковий невідсортований масив

Size\_t n – розмір масиву.

1. В тілі функції створюємо дві змінні, які будуть зберігати максимальне і мінімальне значення, тобто початок і кінець діапазону значень. Зображення, що містить текст, знімок екрана, Шрифт

   Автоматично згенерований опис

За початкові значення максимального і мінімального значення, використовуємо перший елемент масиву. Вираховуємо діапазон значень для допоміжного контейнера (кін. – поч. + 1)

1. Створюємо допоміжний динамічний масив, розміром нашого діапазону, заповнюємо його нулями.

Зображення, що містить текст, Шрифт, знімок екрана, число

Автоматично згенерований опис

Створюємо цикл, який буде проходити від початку до кінця діапазону.

Беремо значення першого елементу початково масиву, віднімаємо від нього мінімальне число, отримуємо індекс в допоміжному масиві, який позначає певне значення в цьому діапазоні, змінюємо лічильник цього значення на один. Приклад: help[142-42] = help[100]++. Це означає, що значення яке знаходиться під індексом 100, зустрічається в початковому масиві один раз.

1. Оголошуємо змінну яка буде зберігати значення поточного індексу. Початкове значення 0.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Створюємо цикл, який буде проходити від 0 до нашого діапазону, тобто буде переміщатись по нашому допоміжному контейнеру. Оголошуємо в середині циклу умову «Допоки значення допоміжного масиву більше за 0 🡺 виконувати”. Викону наступні дії: пробігаємо по масиву help (він містить кількість кожного числа).

Якщо help[i] > 0, то додаємо число i + min у arr (тобто повертаємо йому правильне значення).

Зменшуємо help[i] на 1, бо вставили один екземпляр числа.

1. Чистимо пам'ять.



**Тестування.**

Щоб перевірити чи дійсно працює наш алгоритм оголосимо динамічний масив довільного розміру, ініціалізуємо числами від 0 до 9999, за допомогою rand().

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

В якості зерна для створення набору випадкових чисел, оберемо довільне число.



**Перевірка №1.**

Seed = 71.

Size = 5

Вхідний масив: 

Відсортований масив: 

**Перевірка №2.**

Seed = 71.

Size = 15

Вхідний масив: 

Відсортований масив: 

**Перевірка №3.**

Seed = 993.

Size = 8

Вхідний масив: 

Відсортований масив: 

Алгоритм досконало виконує свою роботу.

**Висновок.**

Counting Sort хороший алгоритм, який чудово виконує свою роботу в малому діапазоні чисел, проте зі збільшенням діапазону втрачає швидкість і ефективність.