**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

Кафедра інформаційних систем

Алгоритми та складність

Лабораторний проект № 4

Сортування масиву нулів і одиниц

**Звіт**

**Виконала:**

студентка групи К-28

Гожда Марія Олександрівна

**Київ-2018**

**Сортування масиву нулів і одиниц**

1. **Умова завдання.**

Відсортувати масив, що скаладається тільки з нулів та одиниць з наступними критеріями:

А) Алгоритм стійкий і час роботи O(n)

B) Час роботи O(n) і сортування проводиться без використання додаткової пам`яті

C) Алгоритм має бути стійким і сортування проводиться на місці

1. **Основні поняття.**

*Стійке сортування – сортування, що не змінює порядок елементів з однаковим ключем*

1. **Опис, аналіз та основні модулі програми.**

• sort1(vector<int> &A)

Сортування, що відповідає критерію А)

Спочатку підраховується кількість нулів і одиниць в масиві і ці значення записуються відповідно у змінні х та у.

Потім створюється допоміжний масив В на таку ж саму кількість елементів як і масив А. Знову проходимось циклом по масиву А, починаючи з кінця і записуємо нулі та одиниці на ті місця на які вказують індекси х та у. Після кожного запису зменшуємо відповідний індекс на одиницю. При цьому взаємне розташування одиниць(нулів) по відношенню одне до одного не змінюється.

В кінці копіюємо значення відсортованого масиву В у масив А.

Час роботи O(n)

• sort2 (vector<int> &A)

Сортування, що відповідає критерію В)

Створюємо нову змінну с, покладемо в неї 0.

Проходимо по масиву А, кожного разу коли зустрічаємо 0 міняємо його місцями з елементом в масиві, який має індекс с. Після цього збільшуємо с на одиницю.

Таким чином всі елементи у масиві з індексами менше ніж с гарантовано будуть нулями. Дійшовши до кінця циклу, отримаємо відсортовану послідовність.

Час роботи O(n)

Додаткова пам`ять не використовується

• sort3(vector<int> &A)

Сортування, що відповідає критерію С)

Алгоритм схожий на попередній тільки замість того щоб одразу обмінювати місцями два елементи ми послідовно зсуваємо елемент до потрібної позиції. Таким чином відносний порядок елементів з однаковим ключем зберігається.

1. **Реалізація алгоритму.**

Реалізовано на С++

1. **Інтерфейс програми.**

Дані (тести) прописані в файлі срр, а результат виводиться в консоль.

1. **Використані структури даних.**

Вектор std::vector<>;

1. **Тестові приклади.**

• vector<int> A = {1,0,1,0,1,0,1}; Вихід: 0 0 0 1 1 1 1

• vector<int> B = {1,1,1,1,1,1,0,0,0,0,0,0,1,1,}; Вихід: 0 0 0 0 0 0 1 1 1 1 1 1 1 1

• vector<int> C = {0,0,0,0,1,1,0,0,1,0,1}; Вихід: 0 0 0 0 0 0 0 1 1 1 1

1. **Використані джерела.**
   * + 1. Лекція №6
       2. https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%B1%D1%96%D0%BB%D1%8C%D0%BD%D0%B5\_%D1%81%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F2.