Київський національний університет імені Тараса Шевченка

факультет комп’ютерних наук та кібернетики

кафедра інформаційних систем

Лабораторна робота №4

0-1 сортування

Виконав студент 2 курсу

Групи К-28

Мосьпан Олег Олександрович

2018

Постановка задачі

Нехай маємо масив, що містить n записів з даними для сортування, і що ключ кожного запису приймає значення 0 або 1. Алгоритм для сортування такого набору записів повинен мати деякі з трьох наступних характеристик: 1) час роботи алгоритму О(n); 2) алгоритм має бути стійким; 3) сортування проводиться на місці, тобто крім вихідного масиву використовується додаткова пам’ять, що не перевищує деякої постійної величини.

Розробити і реалізувати алгоритм, що задовольняє

1) критеріям 1 і 2.

2) критеріям 1 і 3.

3) критеріям 2 і 3 (бажано з найкращим часом).

Опис алгоритму

Створимо структуру Pair з двома полями key та id - ключ запису та його індекс відповідно. a[0..n-1] – масив записів Pair.

1) Заведемо дві черги q0, q1. Послідовно зліва направо пройдемось по масиву a, якщо a[i].key = 0, то додаємо iндекс i в чергу q0, інакше – в q1. На другому кроці помістимо на початок масиву a записи в, яких індексами є послідовні елементи черги q0, а ключем - число 0; на вільні місця масиву - аналогічно для q1. Алгоритм потребує O(n) додаткової пам’яті.

2) Заводимо два показники p0, p1. Якщо p0 = k, то це означає, що записи з індексами 0..k-1 мають ключ 0. Якщо p1 = k – записи k + 1..n - 1 мають ключ 1. Нехай спочатку i = 0. Виконуємо наступні кроки, допоки p0 != p1 (якщо p0 = p1, то масив впорядкований). Якщо a[i].key = 0, то нічого не треба змінювати і ми інкрементуємо i та p0. Інакше треба перемістити даний запис в позицію p1. Для цього ми міняємо місцями записи a[i] та a[p1], та декрементуємо p1. Оскільки нам нічого не відомо про елемент в позиції p1, ми не змінюємо i та повторюємо даний крок знову. На кожній ітерації відстань між p0 і p1 зменшується на одиницю, звідси і випливає складність O(n).

3) Використаємо модифікацію сортування злиттям. Опишемо сортування для звичайного масиву цілих чисел a[0..n-1]. Як і при звичайному сортуванні злиттям, відсортуємо окремо ліву та праву половини масиву, а далі викликаємо функцію злиття двох частин. У даному випадку злиття представлене рекурсивною функцією. Спочатку ми знаходимо елементи, що мають лежати в першій половині масиву (найменші), потім переміщаємо найменші елементи в цю половину. Тоді можна окремо впорядкувати першу і другу половини масиву (кожна з половин також буде складатися з двох впорядкованних частин, що треба об'єднати). Знайти індекси найменших елементів обох масивів, що не ввійшли до 1-ї половини можна знайти за один лінійний прохід. Далі відбувається циклічний зсув частини масиву, що можна зробити за допомогою трьох перегортань. Складність T(n) = 2T(n/2) + O(nlogn) = O(nlog^2n).

Інтерфейс

Назва файлу, де знаходяться вхідні дані вводиться з клавіатури.

**Вхідні дані:** Натуральне число n (n <= 10^5) – розмір масиву записів. Далі n рядків, в кожному з яких два числа key[i] та id[i] – ключ i-го запису та його індекс відповідно (key[i] = {0, 1}, i = 0..n-1).

**Вихідні дані:** Початковий масив записів. Далі цей же масив відсортований за трьома різними алгоритмами.

Тести

Вхідні дані 1:

7

0 0

1 1

0 2

0 3

1 4

0 5

1 6

Вихідні дані 1:

Start array:

(0, 0) (1, 1) (0, 2) (0, 3) (1, 4) (0, 5) (1, 6)

O(n) and stable:

(0, 0) (0, 2) (0, 3) (0, 5) (1, 1) (1, 4) (1, 6)

O(n) and without adding memory:

(0, 0) (0, 5) (0, 2) (0, 3) (1, 4) (1, 6) (1, 1)

stable and without adding memory, O(nlog^2n):

(0, 0) (0, 2) (0, 3) (0, 5) (1, 1) (1, 4) (1, 6)

Вхідні дані 2:

11

1 0

1 1

0 2

1 3

0 4

0 5

1 6

1 7

0 8

1 9

1 10

Вихідні дані 2:

Start array:

(1, 0) (1, 1) (0, 2) (1, 3) (0, 4) (0, 5) (1, 6) (1, 7) (0, 8) (1, 9) (1, 10)

O(n) and stable:

(0, 2) (0, 4) (0, 5) (0, 8) (1, 0) (1, 1) (1, 3) (1, 6) (1, 7) (1, 9) (1, 10)

O(n) and without adding memory:

(0, 8) (0, 5) (0, 2) (0, 4) (1, 3) (1, 6) (1, 7) (1, 1) (1, 9) (1, 10) (1, 0)

stable and without adding memory, O(nlog^2n):

(0, 2) (0, 4) (0, 5) (0, 8) (1, 0) (1, 1) (1, 3) (1, 6) (1, 7) (1, 9) (1, 10)