**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Кафедра інформаційних систем**

**Алгоритми та складність**

**Лабораторна робота № 2**

**Звіт**

**Виконав:**

студент групи К-29

Карсім Олег Сергійович

**Київ-2018**

**Зміст завдання :**

У нас є n болтів певного розміру та n відповідних гайок. Можна порівнювати чи підходять гайка і болт одне до одного, чи ні. Порівнювати між собою гайки та болти заборонено. Уточнення : кількість гайок певного розміру має бути рівна кількості болтів певного розміру, для кожного болта існує одна і тільки одна гайка. Потрібно розбити всі гайки і болти на пари за час, рівний Ɵ(n\*log(n)).

В задачі сказано, що є n болтів та гайок певного розміру, а також гарантовано, що можливо утворити з ними пари так, щоб гайці відповідав бовт того ж розміру. Не можна порівнювати між собою бовти та гайки, але можна сказати, чи певний бовт більше певної гайки, чи ні, чи вони підходять. В програмі реалізовано алгоритм, який базується на процедурі partition, яка використовуться в алгоритмі швидкого сортування, для того щоб ітеративно розбивати на дві підмножини бовти або гайки, які більше або менше даного бовта або гайки.

**Функції:**

void match(char\*bolt, char\*gaika, int left, int right);

Ця функція приймає як аргументи : масив гайок або болтів, даний лівий та правий елементи, а також опорний елемент.

Повертає вона інший опорний елемент від якого ми будемо пізніше відштовхуватись в іншій процедурі **match;**

Суть цієї функції в перестановці елементів .

int pod(char \*array, int left, int right, char pivot)

Функція для використання поділу масиву та перестановки функцією swap

**Складність :**

Складність алгоритму рівна **Ɵ(n \*log(n)).** Функція виконується за лінійний час **Ɵ(n)** , оскільки вона перебирає елементи масиву від даного лівого до правого по-порядку.

Метод **match** виконується за логарифмічний час **Ɵ(log(n)) ,** оскільки розмір масивів на кожному кроці зменшується(ми їх розділяємо на менші). Щось схоже відбувається при рекурсивному розбитті масиву на два підмасиви при сортуванні злиттям.

Отже, в загальному складність алгоритму дорівнює **Ɵ(n \*log(n)).**

**Література :**

Лекція №3 (Шкільняк Оксана Степанівна)