**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

Кафедра інформаційних систем

Алгоритми та складність

Лабораторний проект № 2

Задача про болти і гайки

**Звіт**

**Виконала:**

студентка групи К-28

Гожда Марія Олександрівна

**Київ-2018**

**Задача про болти і гайки**

1. **Умова 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)).

1. **Основні модулі програми і їх опис.**

int separation(vector<int> &v, int left, int right, int pivot)

Ця функція приймає як аргументи : вектор 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 опорних елементів на початок і поверненні опорного елемента.

void match(vector<int> &bolts, vector<int> &nuts, int left, int right)

Ця процедура викликає функцію separation для масиву гайок та болтів.

int pivot = partition(nuts, left, right, bolts[right]);

separation (bolts, left, right, nuts[pivot]);

Потім вона рекурсивно розділяє масиви болтів та гайок на 2 підмасиви (перший до опорного елемента, другий після).

1. **Аналіз алгоритму.**

Складність алгоритму рівна Ɵ(n \*log(n)). Тому що функція separation 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)).

1. **Реалізація алгоритму.**

Реалізовано на С++

1. **Інтерфейс програми.**

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

1. **Використані структури даних.**

Вектори типу

vector<int> v

1. **Тестовий приклад.**

int n = 10;

vector<int> bolts = { 8, 3, 9, 2, 7, 1, 11, 4, 6, 5 };

vector<int> nuts = { 9, 1, 3, 11, 5, 6, 2, 7, 8, 4 };

*Вивід:*

Bolts and nuts after matching:

1 2 3 4 5 6 7 8 9 11

1 2 3 4 5 6 7 8 9 11

1. **Використані джерела.**

1. Лекція №3

2. *https://www.geeksforgeeks.org*