**Київський національний університет імена Тараса Шевченка  
Факультет комп’ютерних наук та кібернетики**

**Алгоритми та складність  
Лабораторна робота №2**

**Алгоритм розбивки всіх гайок і болтів на відповідні пари за час Θ(n log n)**

**Звіт**

**Підготував:**студент групи К-29  
Дацюк Віталій Олегович

**Київ-2019**

**1.Постановка завдання**

Нехай є *n* болтів різного розміру та *n* відповідних гайок. Припустимо, можна порівнювати, чи підходять гайка і болт одне до одного, або гайка більша (чи менша). Порівняти між собою дві гайки чи два болти неможливо. Розробіть і реалізуйте алгоритм розбивки всіх гайок і болтів на відповідні пари за час Θ(n log n).

**2.Опис алгоритму**

Наш алгоритм буде брати за основу алгоритм швидкого сортування. Спочатку ми беремо перший болт і порівнюємо його з першою гайкою. Якщо гайка більше, то “кидаємо” її в кінець підмасиву, якщо гайка менше – то в початок. За рахунок того, що всі болти різні і кожен з низ має відповідну гайку, ми обов’язково знайдемо відповідну гайку для будь-якого болта. Відсортувавши таким чином гайки на більші та менші, ми однозначно визначили позицію одного болта та гайки у масивах. Далі беремо другий болт, порівнюємо його з усіма гайками, місце яких вже однозначно визначені (в нашому випадку це одна гайка – перша) і повторюємо сортування для відповідної групи(групи, де є гайка таких же розмірів,що і болт).

Алгоритм бистрого сортування працює за O(nLogn) , що задовольняє вимогам задачі.

**3.Реалізація алгоритму швидкого сортування**

Алгоритм складається з трьох частин:

* Вибрати елемент з масиву і назвемо його опорним
* Розбивка: перерозподіл елементів в масиві таким чином, що елементи менше опорного розміщуються перед елементом, а більші або рівні після
* Рекурсивно застосувати перших два кроки до двох підмасивів зліва та справа опорного елементу. Рекурсія не

**Література:**

* <https://www.geeksforgeeks.org/quick-sort/>
* <https://ru.wikipedia.org/wiki/%D0%91%D1%8B%D1%81%D1%82%D1%80%D0%B0%D1%8F_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0>
* **“The Art of Computer Programming, vol.3. Sorting and Searching” 2-ed Donald E. Knuth**