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

**Алгоритми та складність  
Лабораторна робота №4**

**Три алгоритми, що задовольняють попарно трьом правилам**

**Звіт**

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

**Київ-2019**

**1.Постановка завдання**

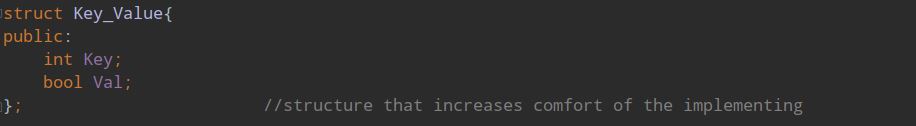
Нехай маємо масив, що містить n записів з даними для сортування, і що ключ кожного запису приймає значення 0 або 1. Алгоритм для сортування такого набору записів повинен мати деякі з трьох наступних характеристик: 1) час роботи алгоритму О(n); 2) алгоритм має бути стійким; 3) сортування проводиться на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

Розробіть і реалізуйте алгоритм, що задовольняє

1. критеріям 1 і 2.

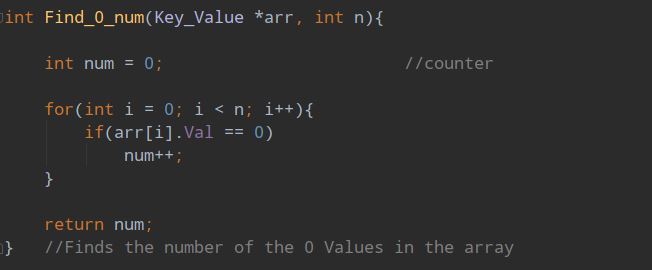
2. критеріям 1 і 3.

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

**2.Основні модулі програми** Для зручності створено структуру Key\_Value, яка містить у собі дві змінних: int Key та bool Val. Key призначено для відмінностей між кожним об’єктом, а Val зберігає значення запису. Всі алгоритми працюють з поданою структурою.

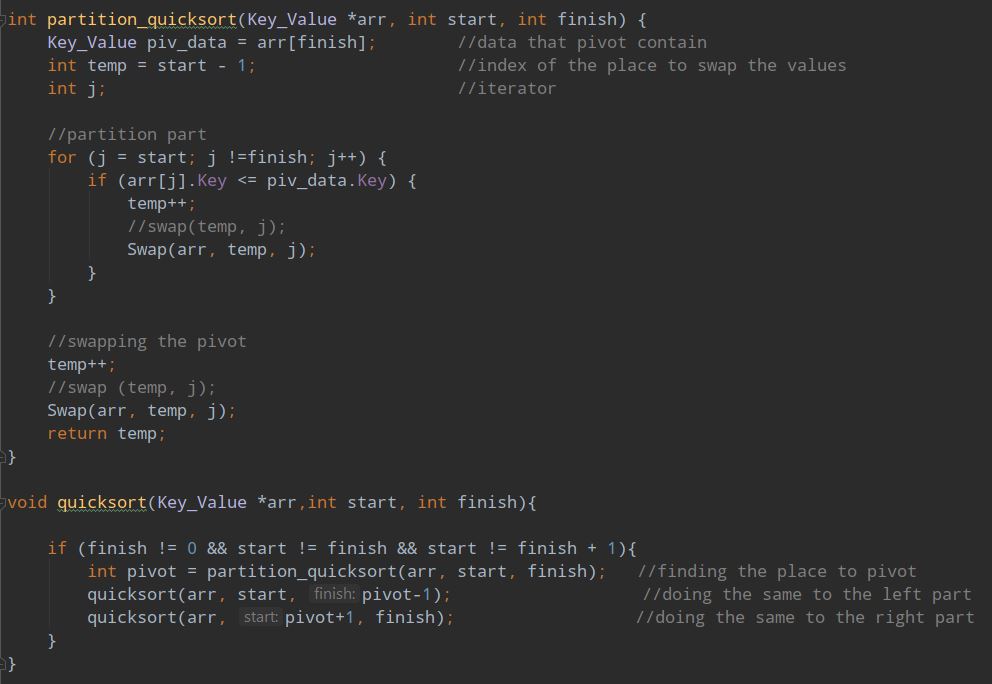
1. **Знаходження кількості нульових елементів**

Функція знаходить кількість елементів в нашому масиві, значення яких дорівнює 0. Використовується в алгоритмах сортування.



1. **Quicksort**

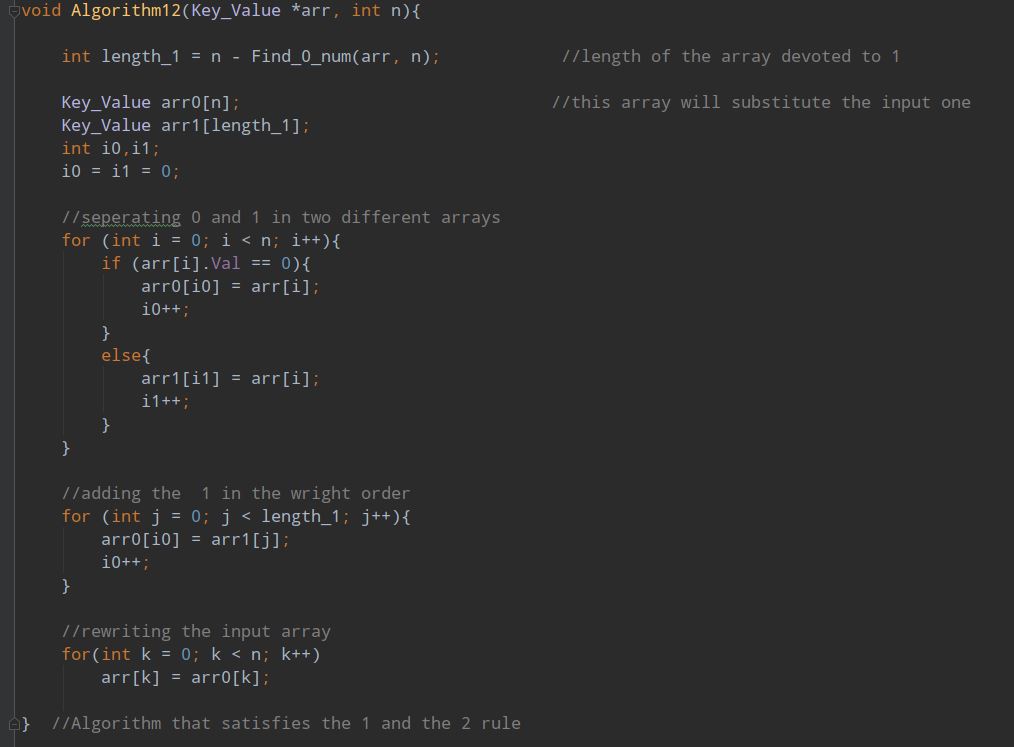
Усім відомий алгоритм quicksort. Використовується в алгоритмі сортування ,який задовольняє правилам 1 та 3.



1. **Алгоритм 12**

Цей алгоритм задовольняє правилам 1 та 2. Він створює два підмасиви, довжини яких відповідають кількості нулів та одиниць відповідно. Далі ми проходимо по основному масиву і записуємо нуль чи одиницю в відповідний масив, збільшивши індекс запису на один для відповідного масиву. Таким чином після проходження всього масиву в нас буде на виході два підмасиви, де зберігаються нулі і одиниці в їх відносному порядку. В кінці записуємо ці два масиви в один, спочатку нулі, потім одиниці.

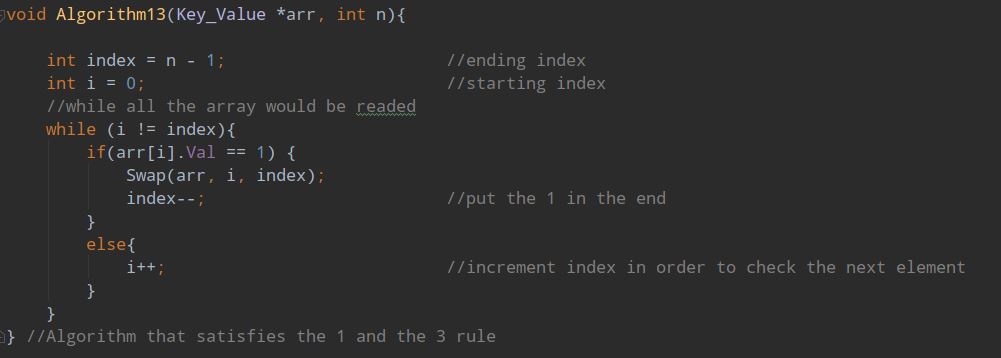
Складність: O(n)



1. **Алгоритм 13**

Цей алгоритм задовольняє правилам 1 та 3. Алгоритм проходить по масиву. Зустрівши нуль, він просто інкрементує індекс елементу. Зустрівши ж одиницю, алгоритм міняє місцями останній елемент з поточним, зменшує індекс кінця на один, при цьому не збільшуючи індекс проходження масиву. Таким чином нулі опиняться на початку, коли одиниці в кінці. Алгоритм не створює додаткову пам’ять.

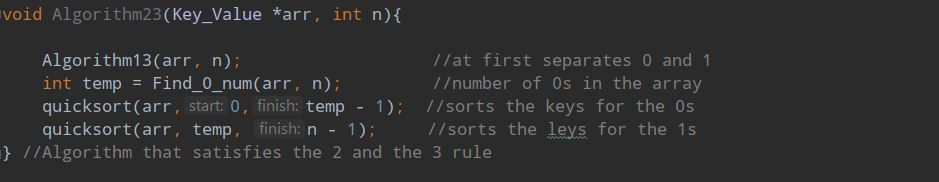
Складність: O(n)



1. **Алгоритм 23**

Цей алгоритм задовольняє правилам 1 та 3.  
Алгоритм використовує алгоритм 13, для відмежування нулів і одиниць за лінійний час. Потім для підмасиву з нулів і підмасиву з одиниць використовується алгоритм quicksort. Таким чином не створюється додаткова пам’ять, для стабільного сортування.

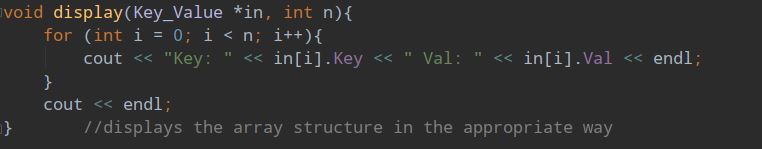
Складність: O(n) + O(n\*logn) = O(n\*logn)



**3.Інтерфейс користувача**

До інтерфейсу користувача належать такі модулі:

* Усі інші описані вже вище функції
* Функцію виводу масиву на екран

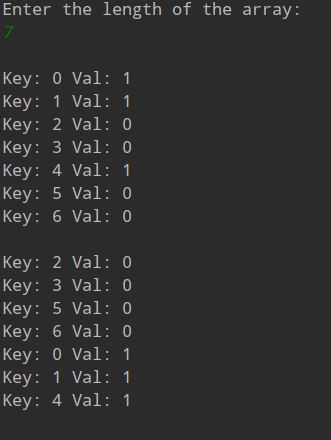


**4.Тестові приклади**

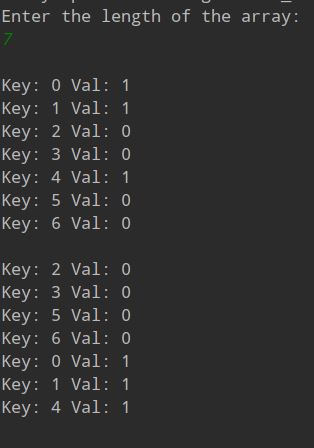
* Алгоритм13:



* Алгоритм23:



* Алгоритм12:



**Література:**

1. <https://www.geeksforgeeks.org/quick-sort/>
2. <https://www.geeksforgeeks.org/stability-in-sorting-algorithms/>