

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Алгоритми та методи обчислення

Лабораторна робота №2

**«Обчислювальна складність
алгоритмів сортування»**

Виконав:

студент групи ІВ-93

Підгайний Д. Р.

Перевірив:

Порєв В. М.

Київ

2021 р.

Тема: «Обчислювальна складність алгоритмів сортування».

Мета: Закріплення навичок практичної оцінки алгоритмічної складності логічних алгоритмів на прикладі алгоритмів сортування.

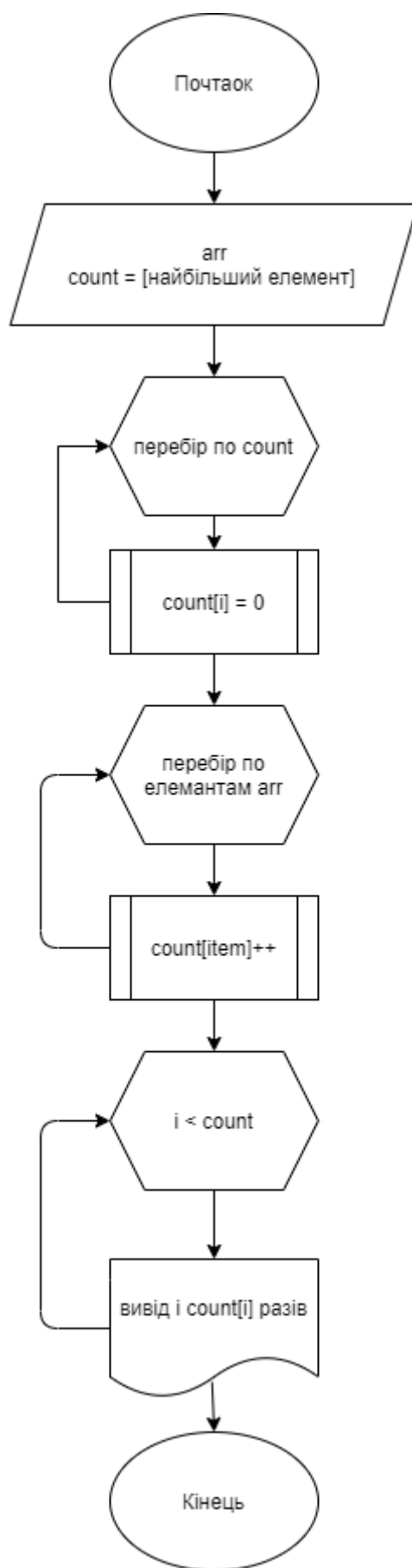
Завдання: Використовуючи відповідний до варіанту алгоритм сортування написати програму сортування масиву даних. Застосовуючи дану програму, дослідити часову складність алгоритму сортування та порівняти її з теоретичною алгоритмічною складністю.

Визначення варіанту:

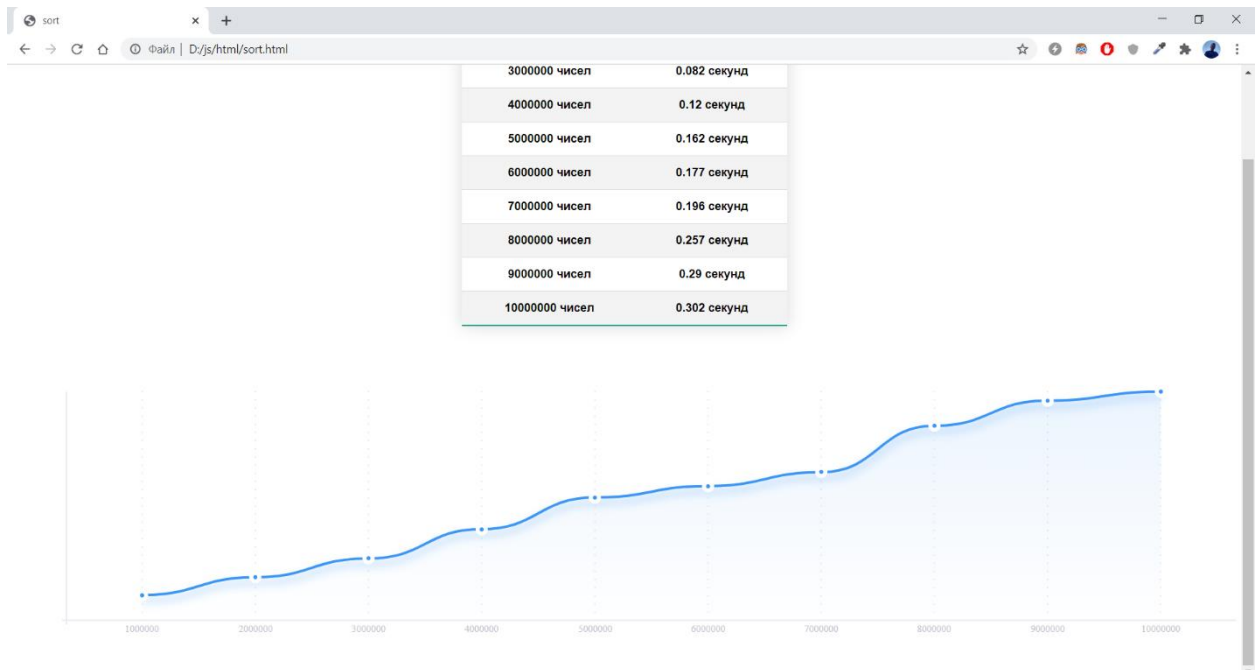
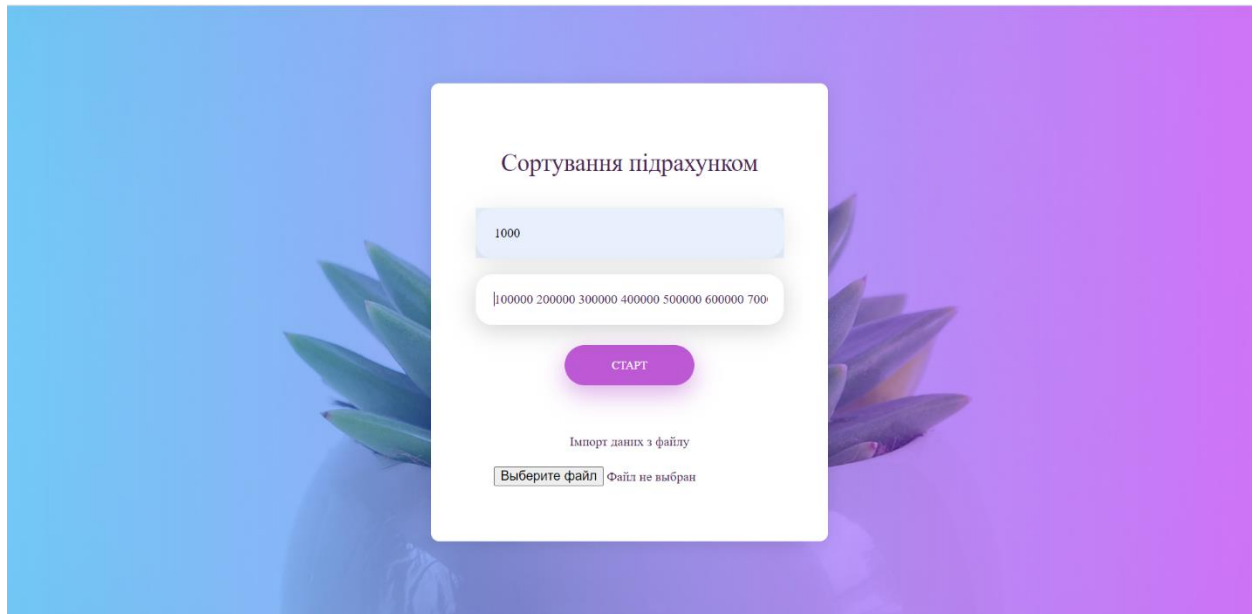
18	Алгоритм 11. Сортування підрахунком	$O(n)$
----	-------------------------------------	--------

Код з файлу script/main.js:

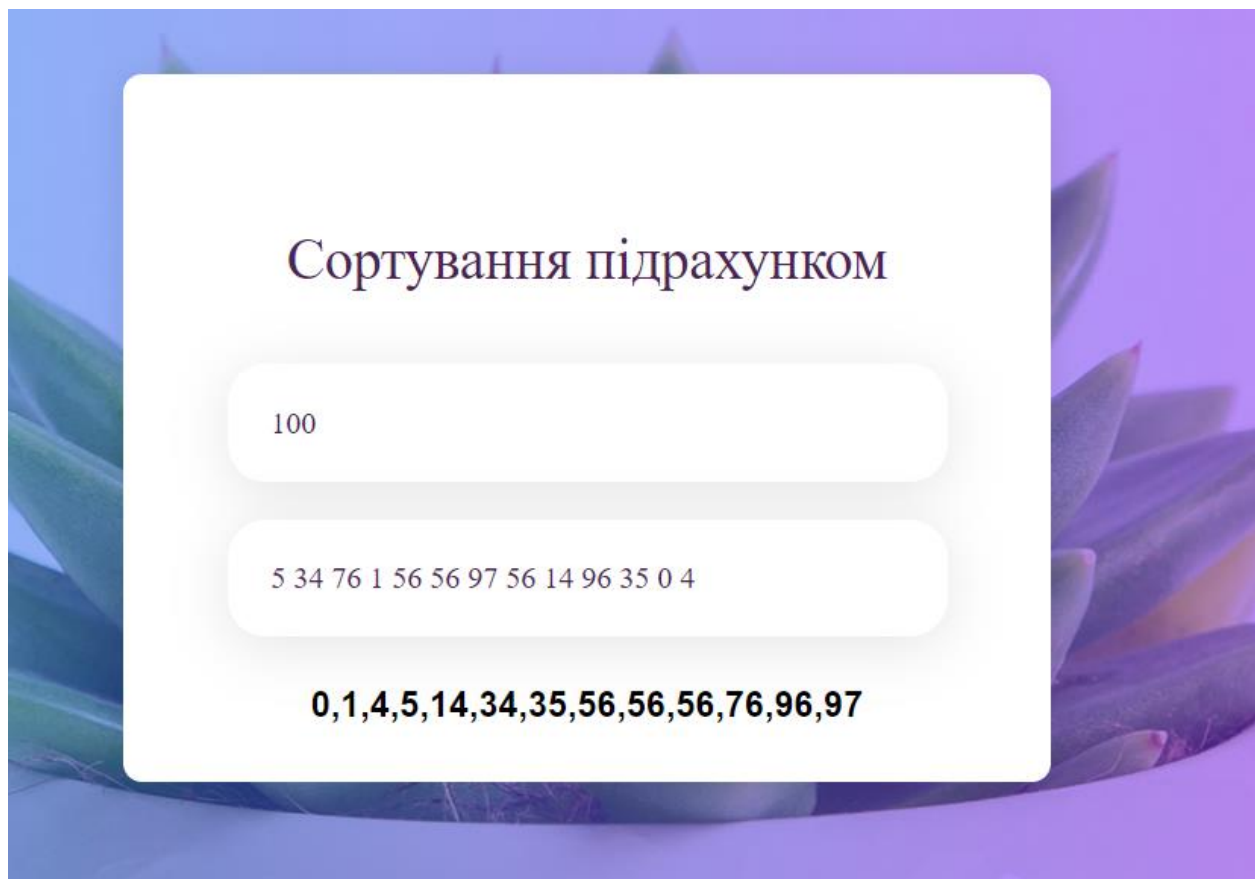
```
1 // Алгоритм 11. Сортування підрахунком o(n)
2 // найбільше число для генерування випадкового масиву
3 let THE_LARGEST_NUMBER = 1000;
4 const ONE_NUMBER = /\d+;/;
5 const AN_ARRAY_OF_NUMBERS = /[ ]?\d+;/;
6 //----->
7
8 get_an_array = (len_arr) => [...Array(len_arr)].map(_ => Math.floor(Math.random() * THE_LARGEST_NUMBER));
9
10 // повертає час виконання в секундах
11 measure_time = (fun, parameter) => {
12   let s_time = Date.now();
13   fun(parameter);
14   return (Date.now() - s_time) / 1000;
15 }
16
17 // сам алгоритм сортування
18 my_sort = (arr) => {
19   let count = Array(THE_LARGEST_NUMBER).fill(0);
20   arr.map(item => count[item]++);
21   let b = [];
22   count.map((item, i) => b.push(...Array(item).fill(i)));
23   return b;
24 }
25
26 //----->
```



Перший метод введення даних:



Другий метод:



Висновки: під час виконання лабораторної роботи були закріплені знання з базових понять алгоритмів, навички практичної оцінки алгоритмічної складності логічних алгоритмів на прикладі алгоритмів сортування.

У ході тестування алгоритму сортування підрахунком графік швидкості наглядово ілюструє лінійну складність обчислень. Результати виконання наведені вище підтверджують працездатність програми, а отже головна мета роботи досягнута.