МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Системний аналіз та інформаційно-аналітичні технології»

Лабораторна робота № 4  
з дисципліни «Алгоритми та структури даних»

(Варіант III)

Виконав:

Олексій ДРАБЧАК

студент групи КН-320Г

Перевірила:

Євгенія МОШКО

Харків 2022

**Тема лабораторної роботи:** Алгоритми сортування та асимптотичний аналіз алгоритмів.

**Мета:** вивчити основні алгоритми сортування масивів і освоїти їх на практиці. Перевірити роботу алгоритмів на різних наборах даних, провести асимптотичний аналіз алгоритмів сортування.

Порядок виконання роботи:

1. Створити функцію сортування масиву алгоритмами сортування, які треба обрати у таблиці 3 відповідно до свого варіанту. Варіант завдання вибирається за номером студента в списку групи.
2. Будь-яким способом заповнити елементи масиву значеннями.
3. Виконати сортування масиву першим алгоритмом і проаналізувати отримані результати.
4. Перевірити всі варіанти вихідного заповнення масиву: випадковим чином, відсортованого у порядку зростання, відсортованого у порядку спадання. Переконатися в правильності сортування у всіх випадках. Зробити висновки.
5. Повторити пункти 2-4 для другого та третьего алгоритму сортування.
6. Дослідити складність алгоритмів.
7. Провести асимптотичний аналіз алгоритмів сортування та зробити висновки.
8. Здійснити порівняння алгоритмів сортування для n = 10, 100, 1000, 10000 і в наступному порядку вхідних елементів:

* елементи вже впорядковані;
* елементи в зворотному порядку;
* розстановка елементів випадкова.

1. Виконання порівняння алгоритмів сортування на основі наступних кроків:

* створити таблицю асимптотичних оцінок трудомісткості алгоритмів в кращому, середньому, гіршому випадках;
* розставити лічильники операцій у функціях угруповань;
* провести експеримент, визначити середню кількість операцій для різних угруповань, побудувати графіки;
* для кожного виклику сортування генерувати новий масив;
* створити таблиці і представити графіки експериментальних оцінок

алгоритмів.

Необхідно враховувати, що порівняння алгоритмів проводиться на однакових вхідних даних.

Зберегти файл з тестом програми для подальших робіт. Заборонено використовувати готові реалізації структур даних (наприклад, STL).

Індивідуальний варіант завдання представлено на рис. 4.1.

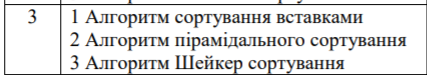
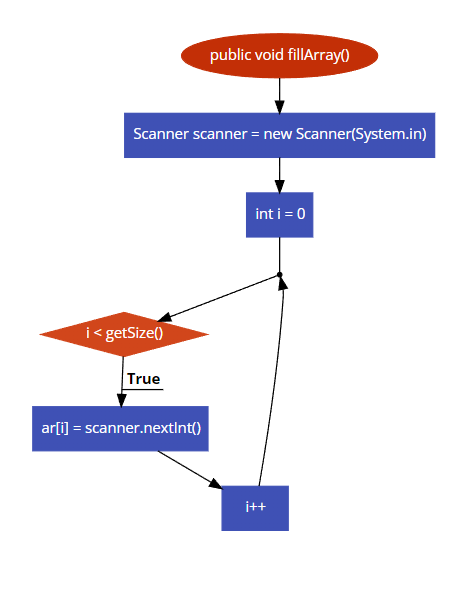
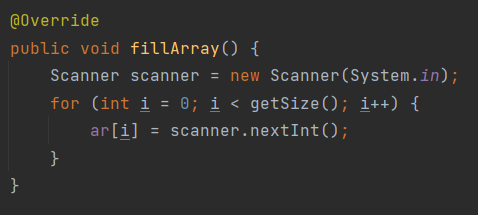
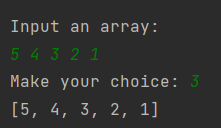


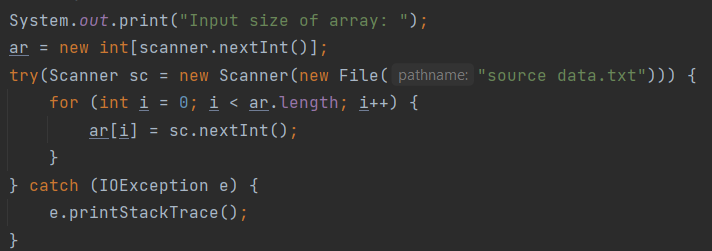
Рисунок 4.1 – Варіант завдання.

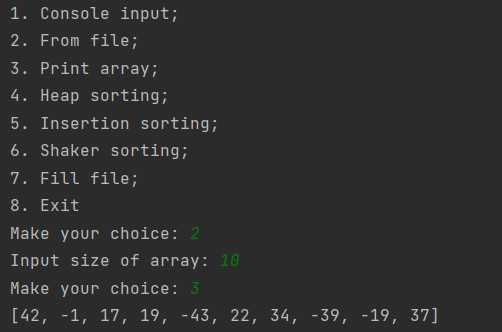
1. Метод заповнення масиву з консолі та з файла.



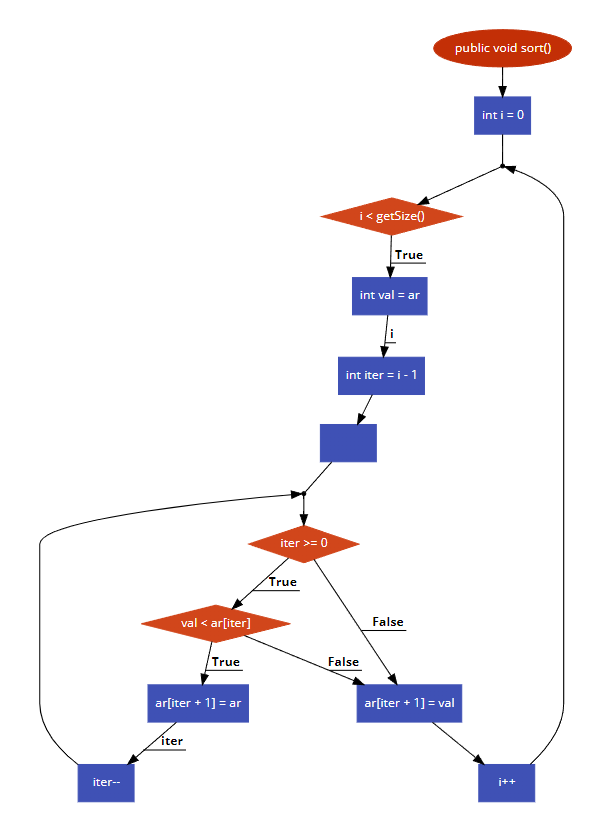


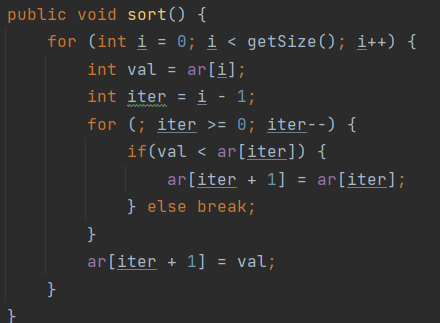


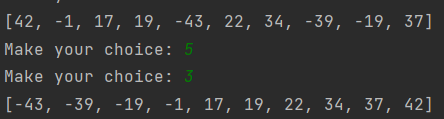




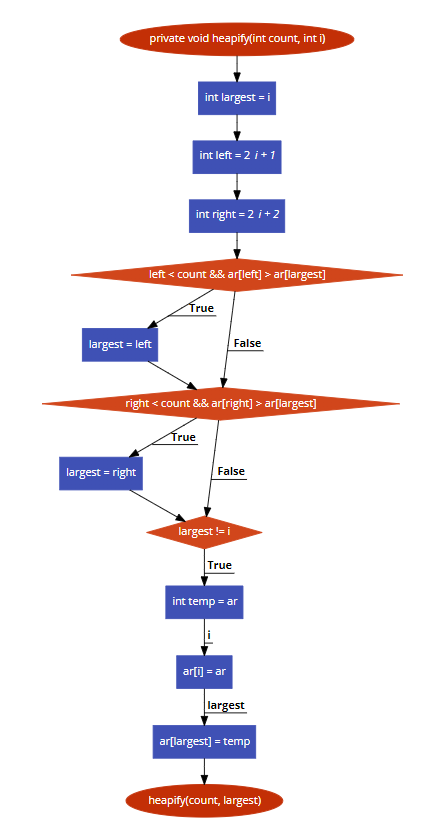
1. Сортування вставками.

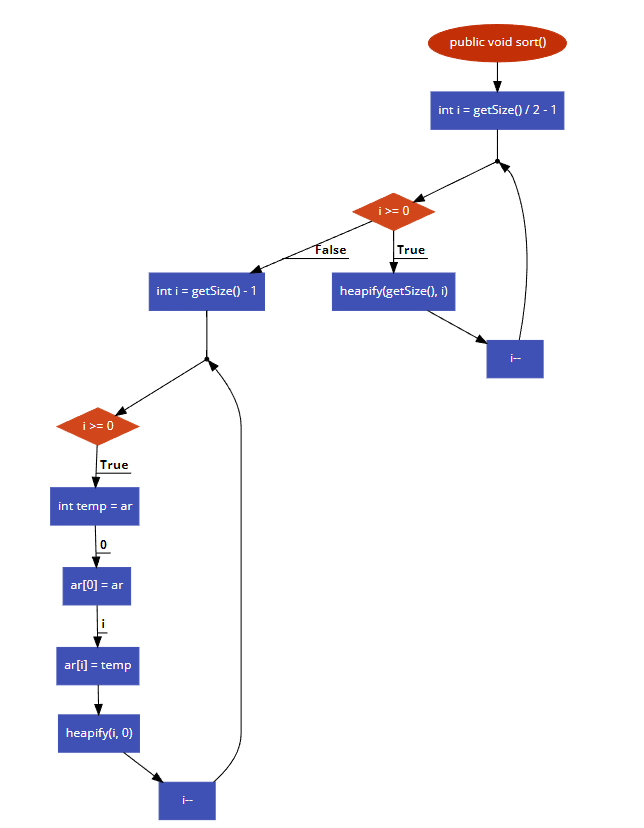


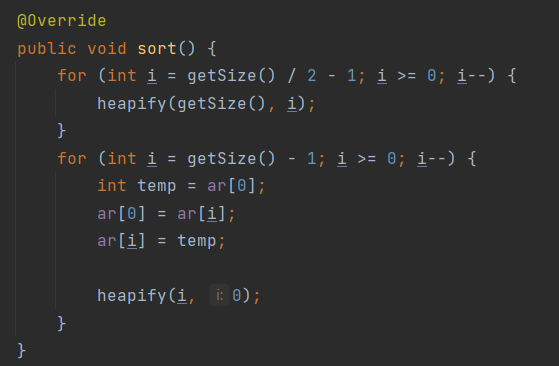


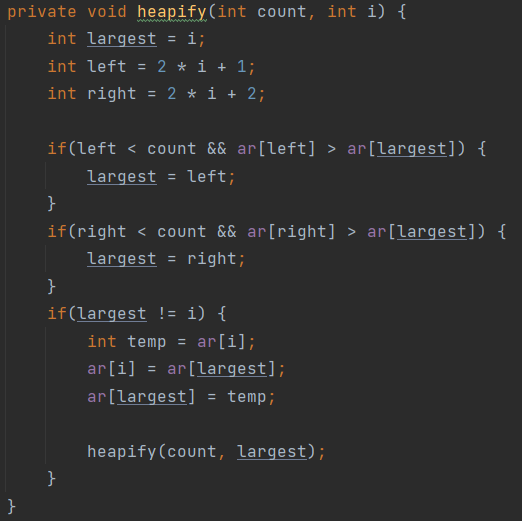


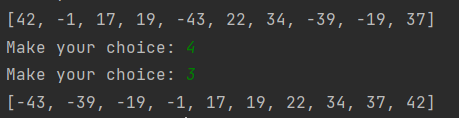
1. Сортування кучею.



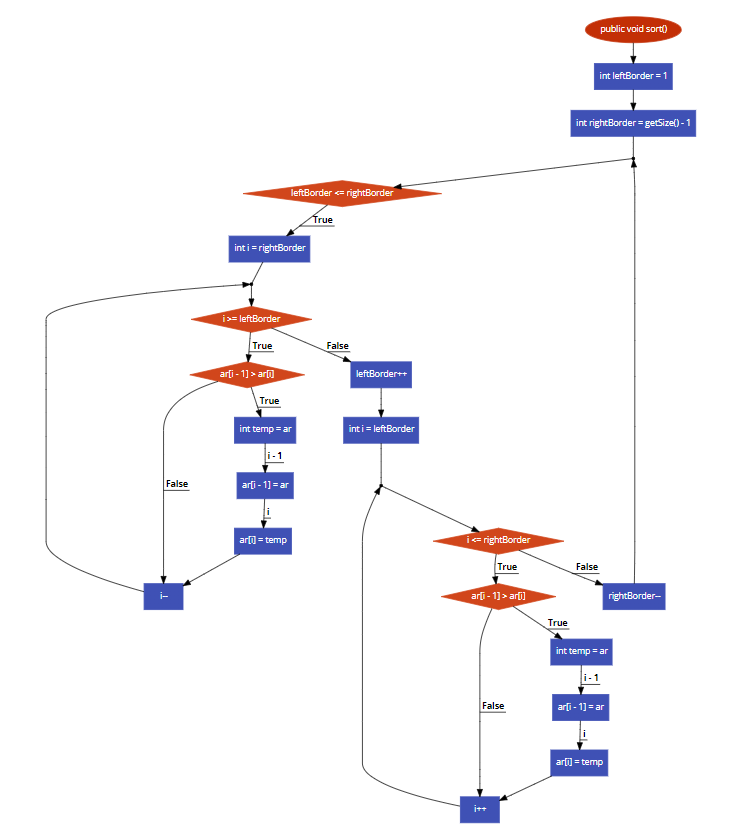


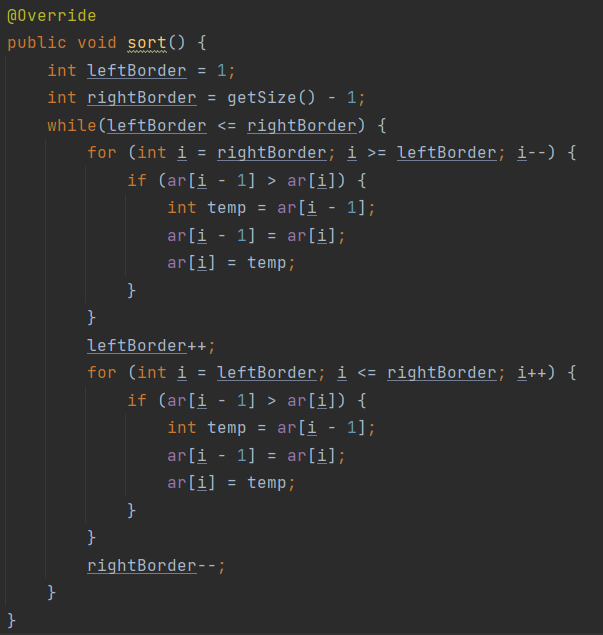


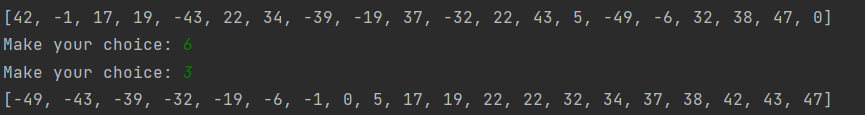




1. Шейкер сортування.







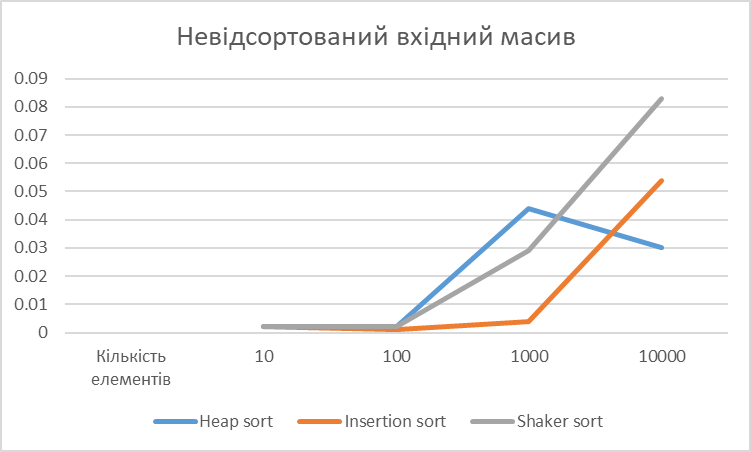
**Асимптотичний аналіз алгоритмів сортування**

|  |  |  |  |
| --- | --- | --- | --- |
| Алгоритм | Найкраще | В середньому | Найгірше |
| Heap sort |  |  |  |
| Insertion sort |  |  |  |
| Shaker sort |  |  |  |

Варто зазначити, що вимірювання проводяться в секундах.

Вхідний масив не відсортовано.

|  |  |  |  |
| --- | --- | --- | --- |
| Кількість елементів | Heap sort | Insertion sort | Shaker sort |
| 10 | 0.002 | 0.002 | 0.002 |
| 100 | 0.002 | 0.001 | 0.002 |
| 1000 | 0.044 | 0.004 | 0.029 |
| 10000 | 0.03 | 0.054 | 0.083 |

****

Вхідний масив відсортовано в порядку зростання.

|  |  |  |  |
| --- | --- | --- | --- |
| Кількість елементів | Heap sort | Insertion sort | Shaker sort |
| 10 | 0.002 | 0.002 | 0.027 |
| 100 | 0.002 | 0.002 | 0.005 |
| 1000 | 0.012 | 0.003 | 0.023 |
| 10000 | 0.051 | 0.02 | 0.031 |

Вхідний масив відсортовано в порядку спадання.

|  |  |  |  |
| --- | --- | --- | --- |
| Кількість елементів | Heap sort | Insertion sort | Shaker sort |
| 10 | 0.001 | 0.002 | 0.002 |
| 100 | 0.003 | 0.031 | 0.003 |
| 1000 | 0.006 | 0.003 | 0.003 |
| 10000 | 0.028 | 0.089 | 0.116 |

**Висновки**

Виконуючи дану роботу, я освоїв та застосував на практиці методи сортувань. А саме: сортування вставками, шейкер сортування та сортування кучею. З результатів тестування можна визначити, що пірамідальний алгоритм сортування значно ефективніший ніж сортування вставками та шейкер сортування.