Міністерство освіти і науки України

Кременчуцький національний університет   
імені Михайла Остроградського

Навчально-науковий інститут електричної інженерії   
та інформаційних технологій

Кафедра автоматизації та інформаційних систем

НаВчальна дисципліна  
«**Алгоритми та структури даних**»

Звіт

З Практичної роботи №3

Виконав

студент групи КН-24-1

Михайлик М. О.

Перевірив

доцент кафедри АІС

Сидоренко В. М.

Кременчук 2024

|  |  |
| --- | --- |
| Тема: | Алгоритми сортування та їх складність. Порівняння алгоритмів сортування |
| Мета: | Опанувати основні алгоритми сортування та навчитись методам аналізу їх асимптотичної складності. |

**Хід роботи**

*Ознайомлення з короткими теоритичними відомостями*

*Виконати завдання.*

1. 1. Вивчити самостійно і записати (будь-яким способом) алгоритм бульбашкового сортування. (рис.Рисунок 1).

2. Оцінити асимптотику алгоритму сортування методом бульбашки в найгіршому і в найкращому випадку.

3. Порівняти за цими показниками бульбашковий алгоритм з алгоритмом сортування вставлянням.

4. Чому на практиці бульбашковий алгоритм виявляється менш ефективним у порівнянні з сортуванням методом зливанням?

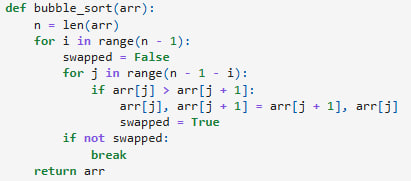


Рисунок 1 – «Бульбашкове сортування»

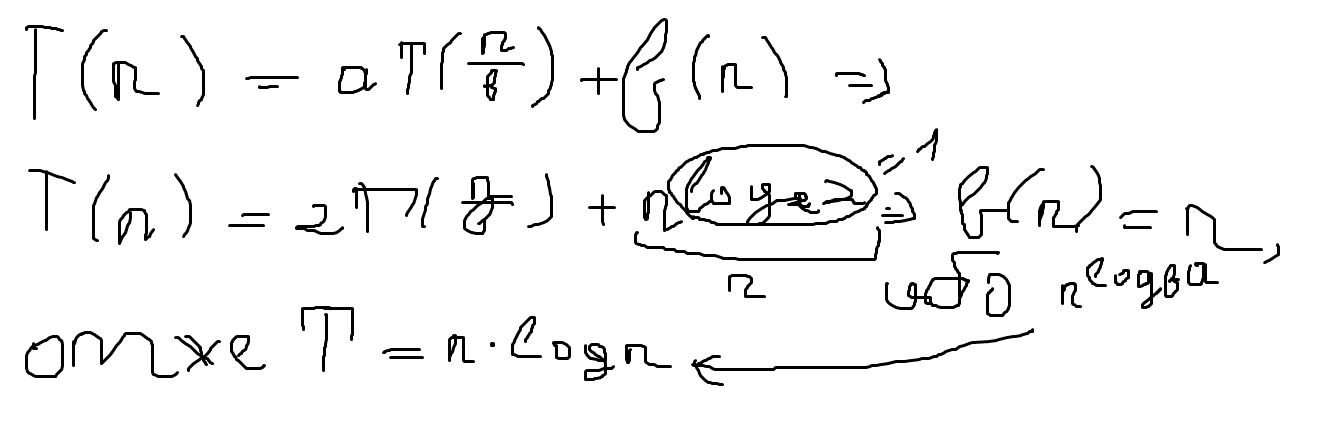
2. У найгіршому випадку, тобто якщо масив повністю не відсортований, алгоритм бульбашкового сортування матиме складність O(N^2).

Тоді як у найкращому: O(N), оскільки масив буде повністю впорядкований і ми лише один раз перевіримо елементи за умовою.

3. Алгоритм вставлянням та бульбашка мають однакову асимптотичну складність.

4. Бульбашка проста для розуміння але витрачає багато часу на виконання кожної дії по черзі, що робить її неефективною на великих обсягах інформації, тоді як алгоритм методом зливання доволі швидко перевіряє велику кількість інформації розділяючи її на 2 половини, а отже, навіть якщо об’єм роботи збільшиться у 2 рази, алгоритм виконає лише одну додаткову ітерацію, що робить його більш швидким та ефективним.

2. З основної теореми рекурсії: (рис.2)



3. Вивчити і записати (будь-яким способом) самостійно алгоритм швидкого сортування. Оцінити асимптотичну складність алгоритму швидкого сортування, скориставшись основною теоремою рекурсії. (рис. 1, 2)

В найгіршому випадку ми беремо найбільший або найменший елемент в початку виконання коду, тому швидкість зменушється за асимптотика змінюється, тоді як в усіх інших випадках асимптотика алгоритму як і в алгоритмі злиття.

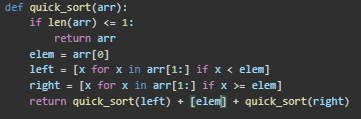


Рисунок 1 – «Швидке сортування»

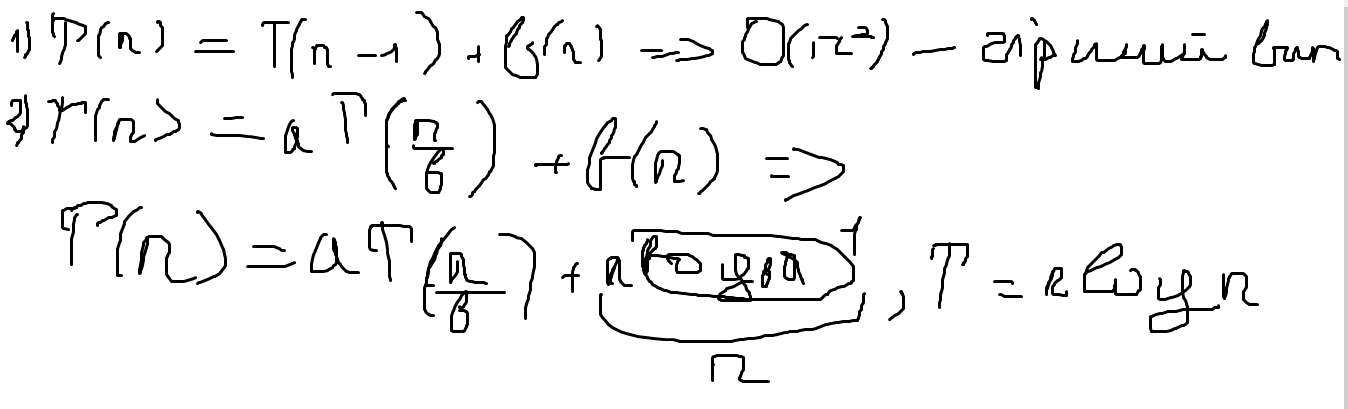


Рисунок 2 – «Асимптотика алгоритму»

*Додаткові питання*

1. Асимптотична складність алгоритму це характеристика , яка визначає відношення часу виконання певної дії до кількості вхідних даних у алгоритм. Вона оцінює складність алгоритму та виявляє його сильні та слабкі сторони.
2. Бульбашка, сортування вставкою, швидке сортування. Складність n^2 погано справляється з великою кількістю інформації оскільки дуже швидко зростає кількість операцій, а отже і час виконання алгоритму.
3. Сортування злиттям має перевагу надо алгоритмом вставки якщо задача має велику кількість інформації оскільки асимптотика злиття n\*log(n) набагато краще підходить для великих обсягів даних.
4. Python: Timsort – щось середнє з сортування злиттям та вставкою;

Java: Arrays.sort – трохи адаптоване швидке сортування;

C++ : sort() – схоже на швидке сортування;

1. Сортування злиттям використовується якщо вам потрібні точні та надійні розрахунки і ви не паритесь за час виконання, тоді як швидке сортування має більшу швидкість але при великих кількостях інформації значно втрачає в точності та надійності.
2. При виборі алгоритму сортування потрібно спиратися на об’єм інформації, потрібну вам швидкість, та точніть обрахунків.

**Висновки:** В ході цієї роботи ми опанували основні алгоритми сортування та навчились методам аналізу їх асимптотичної складності..