Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра комп’ютерної інженерії

КУРСОВА РОБОТА

Тема: “Порівняння методів сортування масивів”

з дисципліни «Програмування»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Керівник \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис) (прізвище, ініціали)

Студент гр. \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис) (прізвище, ініціали)

Харків 2016

РЕФЕРАТ

Записка пояснювальна до курсової роботи: 23 с., 5 рис., 4 розділи, 2 додатка.

Мета роботи — розробка програми для порівняння таких методів сортування, як: метод злиття, швидке сортування та за допомогою купи.

Метод вирішення задачі — для вирішення поставленої задачі були проаналізовані основні алгоритми сортування масивів.

Розроблено програму, яка реалізує основні алгоритми сортування та дозволяє заміряти час виконання цих алгоритмів, що дозволяю зручно порівнювати швидкість виконання реалізованих методів.

Програму складено мовою C++ у середовищі програмування Visual Studio 2013.

СОРТУВАННЯ, С++, ШВИДКЕ СОРТУВАННЯ, СОРТУВАННЯ ЗЛИТТЯМ, СОРТУВАННЯ ЗА ДОПОМОГОЮ КУПИ, АЛГОРИТМ.

ЗМІСТ

[ВСТУП 4](#_Toc452049707)

[1. ТЕОРЕТИЧНА ЧАСТИНА 5](#_Toc452049708)

[1.1 Постановка задачі 5](#_Toc452049709)

[1.2 Сортування злиттям 5](#_Toc452049710)

[ВИСНОВКИ 8](#_Toc452049711)

[ПЕРЕЛІК ПОСИЛАНЬ 9](#_Toc452049712)

[Додаток А Код програми 10](#_Toc452049713)

# ВСТУП

Алгоритм - набір інструкцій, що описують порядок дій виконавця для досягнення певного результату.

Часто в якості виконавця виступає комп'ютер, але поняття алгоритму необов'язково відноситься до комп'ютерних програм, так, наприклад, чітко описаний рецепт приготування страви також є алгоритмом, в такому випадку виконавцем є людина (а може бути і певний механізм, ткацький верстат, і ін.).

Поняття алгоритму належить до первісних, основним, базисним поняттям математики. Обчислювальні процеси алгоритмічного характеру (арифметичні дії над цілими числами, знаходження найбільшого загального дільника двох чисел і т. Д.) Відомі людству з глибокої давнини. Однак в явному вигляді поняття алгоритму сформувалося лише на початку XX століття.

Алгоритм сортування - це алгоритм для упорядкування елементів в списку. У разі, коли елемент списку має кілька полів, поле, що служить критерієм порядку, називається ключем сортування. На практиці в якості ключа часто виступає число, а в інших полях зберігаються будь-які дані, що не впливають на роботу алгоритму.

Алгоритми сортування оцінюються за швидкістю виконання та ефективності використання пам'яті:

Час - основний параметр, що характеризує швидкодію алгоритму. Називається також обчислювальною складністю. Для упорядкування важливі найгірша, середня і краща поведінка алгоритму.

Пам'ять - ряд алгоритмів вимагає виділення додаткової пам'яті під тимчасове зберігання даних. Як правило, ці алгоритми вимагають O (log n) пам'яті. При оцінці не враховується місце, яке займає вихідний масив і не залежать від вхідної послідовності витрати, наприклад, на зберігання коду програми (так як все це споживає O (1)).

# 1. ТЕОРЕТИЧНА ЧАСТИНА

* 1. Постановка задачі

Була поставлена задача розробити додаток, метою якого є порівняння методів сортування масивів. У якості цих методів були обрані такі алгоритмі:

* сортування злиттям;
* швидке сортування;
* сортування за допомогою купи.

Для реалізації можливості порівняння цих алгоритмів виникає необхідність в розробці засобів для виконання підрахунку витраченого часу на виконання впорядкування заданого масиву. Так як необхідно порівнювати декілька алгоритмів, то виникає необхідність реалізувати цей засіб таким чином, щоб виключити дуплікації коду та забезпечити зручне додавання нових методів для аналізу.

Також, для виконання порівнянь алгоритмів, необхідно буде для кожного методу надавати необхідні данні для сортування. Тому необхідно розробити можливість введення цих даних користувачем і збереження їх для подальшого використання у інших алгоритмах. Але на невеликому обсязі даних неможливо виконати перевірку швидкості виконання, тому цей додаток повинен надати можливість генерувати масиви з випадкових чисел заданої довжини, що суттєво збільшить швидкість роботи з цією програмою.

Розроблювана програма повинна мати простий користувальницький інтерфейс, який дозволив би виконувати усі необхідні завдання. Також програма повинна працювати у весь час, доки користувач не вибере необхідний пункт у меню для завершення програми.

* 1. Сортування злиттям

Сортування злиттям - алгоритм сортування, який впорядковує списки (або інші структури даних, доступ до елементів, яких можна отримувати тільки послідовно) в певному порядку. Ця сортування - хороший приклад використання принципу «розділяй і володарюй». Спочатку завдання розбивається на кілька підзадач меншого розміру. Потім ці завдання вирішуються за допомогою рекурсивного виклику або безпосередньо, якщо їх розмір досить малий. Нарешті, їх рішення комбінуються, і виходить рішення вихідної завдання.

Рішення даної задачі сортування можна зобразити у вигляді блок-схеми, представленої на рис. 1.1.

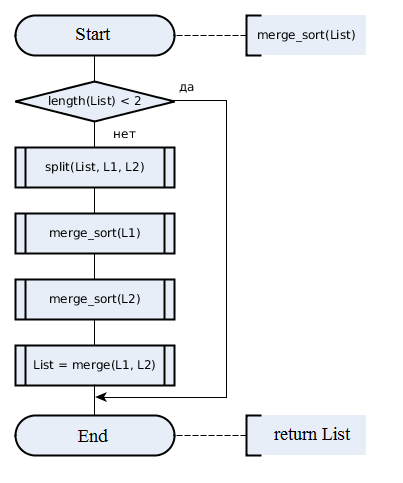


Рисунок 1.1 – «Блок-схема сортування злиттям»

Дана блок-схема має наступні кроки:

* Сортований масив розбивається на дві частини приблизно однакового розміру.
* Кожна з отриманих частин сортується окремо, наприклад - тим же самим алгоритмом. Рекурсивне розбиття на менші відбувається до тих пір, поки розмір масиву не досягне одиниці (будь-який масив довжини 1 можна вважати впорядкованим).
* Два упорядкованих масиву половинного розміру з'єднуються в один. Основну ідею злиття двох відсортованих масивів можна пояснити на наступному прикладі. Нехай ми маємо два вже відсортованих по зростанню підмасива. тоді злиття двох підмасивів у третій результуючий масив. На кожному кроці ми беремо менший з двох перших елементів підмасивів і записуємо його в результуючий масив. Лічильники номерів елементів результуючого масиву і підмасива, з якого був узятий елемент, збільшуємо на 1. Коли один з підмассивов закінчився, ми додаємо решту елементів другого підмасива в результуючий масив.

Алгоритм був винайдений Джоном фон Нейманом в 1945 році.

Час роботи алгоритму O (n \* log n) при відсутності деградації на невдалих випадках, які є болючим місцем швидкого сортування.

До переваг данного методу сортування можно віднести:

* Працює навіть на структурах даних послідовного доступу.
* Добре поєднується з підкачкою і кешуванням пам'яті.
* Непогано працює в паралельному варіанті: легко розбити завдання між процесорами порівну, але важко зробити так, щоб інші процесори взяли на себе роботу, в разі якщо один процесор затримається.
* Не має «важких» вхідних даних.

Але цей метод також має і свої недоліки:

* На «майже відсортованих» масивах працює так само довго, як на хаотичних.
* Вимагає додаткової пам'яті за розміром вихідного масиву.

## 1.3 Швидке сортування

Швидке сортування — алгоритм сортування, добре відомий, як алгоритм розроблений Чарльзом Гоаром, який не потребує додаткової пам'яті і виконує у середньому O(n\*log(n)) операцій. Однак, у найгіршому випадку робить O(n^2) порівнянь. Оскільки алгоритм використовує дуже прості цикли і операції, він працює швидше інших алгоритмів, що мають таку ж асимптотичну оцінку складності. Наприклад, зазвичай більш ніж удвічі швидший порівняно з сортуванням злиттям.

Ідея алгоритму полягає в переставлянні елементів масиву таким чином, щоб його можна було розділити на дві частини і кожний елемент з першої частини був не більший за будь-який елемент з другої. Впорядкування кожної з частин відбувається рекурсивно. Алгоритм швидкого сортування може бути реалізований як у масиві, так і в двозв'язному списку. Цей алгоритм може бути представлений у вигляді блок-схеми зображеної на рис. 1.2.

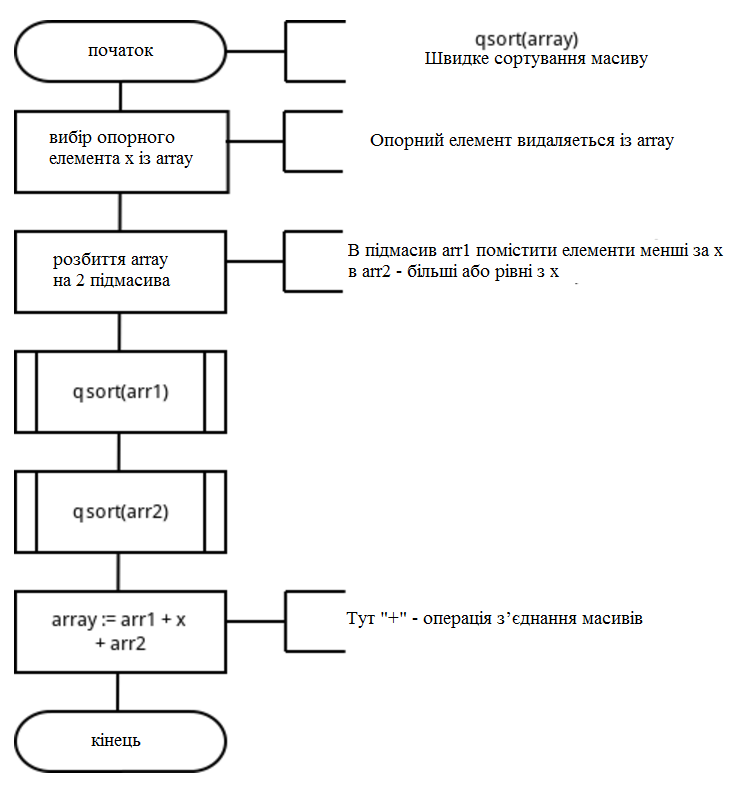


Рисунок 1.2 – «Блок-схема швидкого сортування»

Швидке сортування є алгоритмом на основі порівнянь, і не є стабільним.

Алгоритм швидкого сортування було розроблено Тоні Гоаром (C. A. R. Hoare) у 1962 під час роботи у маленькій британській компанії Elliott Brothers.

Час роботи алгоритму сортування залежить від збалансованості, що характеризує розбиття. Збалансованість, у свою чергу залежить від того, який елемент обрано як опорний (відносно якого елемента виконується розбиття). Якщо розбиття збалансоване, то асимптотично алгоритм працює так само швидко як і алгоритм сортування злиттям. У найгіршому випадку, асимптотична поведінка алгоритму настільки ж погана, як і в алгоритму сортування включенням.

Найгірша поведінка має місце у тому випадку, коли процедура, що виконує розбиття, породжує одну підзадачу з n-1 елементом, а другу — з 0 елементами. Нехай таке незбалансоване розбиття виникає при кожному рекурсивному виклику. Для самого розбиття потрібен час О(n). Тоді, рекурентне співвідношення для часу роботи, можна записати так:

T(n) = T(n-1)+T(0)+ θ (n) = T(n-1)+ θ(n).

Розв'язком такого співвідношення є T(n)=О(n^2).

В найкращому випадку процедура Partition ділить задачу на дві підзадачі, розмір кожної не перевищує n/2. Час роботи, описується нерівністю:

T(n) ≤ 2T(n/2) + θ(n).

Тоді:

T(n)= θ(n\*log n) — асимптотично найкращий час.

Математичне очікування часу роботи алгоритму на всіх можливих вхідних масивах є O(n\*log n), тобто середній випадок ближчий до найкращого.

В середньому алгоритм працює дуже швидко, але на практиці, не всі можливі вхідні масиви мають однакову імовірність. Тоді, шляхом додання рандомізації вдається отримати середній час роботи в будь-якому випадку.

## 1.4 Сортування за допомогою купи

Сортування купою або пірамідальне сортування — алгоритм сортування, працює в гіршому, в середньому і в кращому випадку (тобто гарантовано) за Θ(n log n) операцій при сортуванні n елементів. Кількість застосовуваної службової пам'яті не залежить від розміру масиву (тобто, O (1)).

Сортування пірамідою використовує бінарне сортувальне дерево. Бінарне сортувальне дерево — це структура даних, що є масивом, який можна розглядати як майже повне бінарне дерево. Кожен вузол цього дерева відповідає певному елементу масива. На всіх рівнях, крім, можливо останнього, дерево повністю заповнене (заповнений рівень — такий, що містить максимально можливу кількість вузлів). Останній рівень заповнюється послідовно зліва направо до тих пір, доки в масиві не закінчяться елементи.

Для масиву Array у корені дерева знаходиться елемент Array[1]. Далі дерево будується за наступним принципом: якщо якомусь вузлу відповідає індекс i, то індекс його батьківського вузла і/2, індекс лівого дочірнього вузла — 2і, а індекс правого дочірнього вузла 2і+1.

Розглядають два види бінарних куп: неспадні і незростаючі. В обох видах значення, що розташовані у вузлах купи, задовольняють властивості купи. Властивість незростаючої купи полягає в тому, що для кожного вузла крім кореневого виконується нерівність Array[i/2] ≥ Array[i]. Іншими словами, значення вузла не перевищує значення батьківського вузла. Таким чином найбільший елемент знаходиться в корені дерева.

Принцип побудови неспадної купи протилежний. Властивість неспадної купи полягає в тому, що кожен елемент крім кореневого є неменшим за свій батьківський елемент Array[i/2] ≤ Array[i].

Зручна структура даних для сортувального дерева — такий масив Array, що Array[1] — елемент в корені, а нащадки елемента Array[i] є Array[2i] і Array[2i+1].

Алгоритм сортування складатиметься з двох основних кроків:

* Вибудовуємо елементи масиву у вигляді сортувального дерева. Цей крок вимагає O(n) операцій.
* Будемо видаляти елементи з кореня по одному за раз і перебудовувати дерево. Тобто на першому кроці обмінюємо Array[1] і Array[n], перетворюємо Array[1], Array[2], … , Array[n-1] в сортувальне дерево. Потім переставляємо Array[1] і Array[n-1], перетворюємо Array[1], Array[2], … , Array[n-2] в сортувальне дерево. Процес продовжується до тих пір, поки в сортувальному дереві не залишиться один елемент. Тоді Array[1], Array[2], … , Array[n] — впорядкована послідовність. Цей крок вимагає O(n \ log n) операцій.

# ВИСНОВКИ

У процесі виконання курсової роботи крім закріплення старих знань були придбані так само і нові.

Був розробленій додаток, який дозволяє виконувати сортування будь-якої послідовності чисел різними алгоритмами а також заміряти час їх виконання, що може допомогти у подальшому аналізі та дослідженні продуктивності алгоритмів сортування.

Також був розроблений універсальний засіб для заміру часу виконання вибраних алгоритмів. Це дозволить у майбутньому з легкістю розширювати кількість алгоритмів, які може аналізувати розроблений додаток.

Серед запропонованих алгоритмів сортування, завдяки розробленому додатку, був обраний найшвидший алгоритм. ???

Всі ці набуті навички послужать міцної основою для подальшого отримання знань.

# ПЕРЕЛІК ПОСИЛАНЬ

1. П. Путятiн, В.П. Степанов, В.П. Пчелiнов, Т.Г. Долженкова, О.О. Матат “Основи програмування мовою С++” – Харьков: СМИТ, 2005г.
2. Джесс Либерти «Освой самостоятельно С++ за 21 день»
3. В.М. Бондарев “Програмирование на С++” – Харьков: СМИТ, 2005г.

# Додаток А Код програми