

Завдання на курсову роботу
з дисципліни
“Структури даних та алгоритми”
Тема: “Дослідження ефективності методів сортування
на багатовимірних масивах”

1. Описати принцип роботи кожного з методів, що досліджуються для одновимірного випадку.
2. Скласти алгоритми сортування в багатовимірному масиві методами, що задані за варіантом.
3. Провести практичні дослідження швидкодії складених алгоритмів.
4. За результатами досліджень скласти порівняльні таблиці за різноманітними ознаками.
5. Зробити висновки про порівняння отриманих результатів.

P.S. При бажанні, можна дослідити вплив геометричних розмірів масиву на швидкість сортування, тобто не змінюючи загальної кількості елементів змінювати геометричні розміри масиву. Для випадку перестановки перерізів, не повинна змінюватись загальна кількість перерізів та загальна кількість елементів масиву, а іншими параметрами розмірності можна вар'ювати. Для випадку перестановки рядків/стовпців у кожному перерізі – не повинна змінюватись кількість рядків/стовпців у кожному перерізі та загальна кількість елементів масиву.

Задачі

1. Впорядкувати кожний переріз (зміна третього індекса змінює номер перерізу) тривимірного масиву $A[m, n, p]$ наступним чином: переставити стовпці перерізу за незростанням сум їх елементів.
2. Впорядкувати тривимірний масив $A[m, n, p]$ наступним чином: переставити перерізи масиву (зміна третього індекса змінює номер перерізу) за неспаданням сум їх елементів.
3. Впорядкувати за неспаданням кожну побічну діагональ всіх перерізів (зміна третього індекса змінює номер перерізу) масиву $A[n, n, p]$.
4. Наскрізне сортування матриці $A[m, n]$ по рядках за неспаданням.
5. Наскрізне сортування матриці $A[m, n]$ по стовпцях за незростанням.
6. Впорядкувати кожний переріз (зміна першого індекса змінює номер перерізу) тривимірного масиву $A[m, n, p]$ наступним чином: переставити стовпці перерізу за незростанням сум їх елементів.
7. Впорядкувати тривимірний масив $A[m, n, p]$ наступним чином: переставити перерізи масиву (зміна другого індекса змінює номер перерізу) за неспаданням сум їх елементів.
8. Впорядкувати за неспаданням кожну головну діагональ всіх перерізів (зміна другого індекса змінює номер перерізу) масиву $A[n, p, n]$.
9. Впорядкувати кожний переріз (зміна першого індекса змінює номер перерізу) тривимірного масиву $A[m, n, p]$ наступним чином: переставити рядки перерізу за неспаданням перших елементів кожного рядка перерізу.
10. Впорядкувати тривимірний масив $A[m, n, p]$ наступним чином: переставити перерізи масиву (зміна першого індекса змінює номер

перерізу) за неспаданням сум елементів перших рядків кожного перерізу.

Методи

1. Вибір.
2. Вставка з лінійним пошуком місця вставки від початку відсортованої частини (“зліва”).
3. Вставка з лінійним пошуком місця вставки від взятого елемента (“справа”) без бар’єру.
4. Прямий обмін.
5. Обмін з використанням прапорця.
6. Обмін із запам’ятовуванням місця останньої перестановки.
7. Шейкерне сортування.
8. Сортування Шелла.

Способи обходу

1. Переписати елементи вихідного масиву в додатковий одномірний масив. Виконати сортування. Повернути результат у вихідний масив.
2. Не використовуючи додатковий масив, виконати сортування перетворюючи один індекс елементів “уявного” вектора у відповіднійому індекси елементів вихідного масиву.
3. Виконати сортування, виконуючи обхід безпосередньо по елементам вихідного масиву, не використовуючи додаткових масивів та перетворень індексів.

Стан масиву

1. Вихідний масив впорядкований, як задано за варіантом.
2. Елементи вихідного масиву невпорядковані.
3. Вихідний масив впорядкований протилежно, ніж задано за варіантом.

Звіт має містити:

1. Титульний аркуш оформленний за зразком.
2. Технічне завдання.
3. Опис теоретичних положень.
4. Структурна схема виклику підпрограм.
5. Призначення процедур та функцій.
6. Текст програми з коментарями.
7. Тести (для не великих розмірів масиву).
8. Результати (великі розміри, приблизно четвертого-п'ятого порядку за кожною координатою).

Результати оформити у вигляді порівняльних таблиць.

	Впорядкований	Невпорядкований	Зворотно впорядкований
Метод №1			
Метод №2			
Метод №3			

9. Висновки за отриманими результатами.
10. Список використаної літератури.

Розподіл варіантів

Варіант №	Задача	Методи	Способи обходу	Стан масиву
1	4	1, 2, 7	1, 2, 3	1, 2, 3
2	3	1, 3, 7		1, 2, 3
3	7	2, 4, 8		1, 2, 3
4	1	3, 5, 8		1, 2, 3
5	9	2, 5, 7		1, 2, 3
6	3	1, 6, 7		1, 2, 3
7	2	3, 6, 8		1, 2, 3
8	5	1, 3, 6	1, 2, 3	1, 2, 3
9	8	2, 5, 8		1, 2, 3
10	5	1, 2, 7	1, 2, 3	1, 2, 3
11	10	1, 6, 8		1, 2, 3
12	6	2, 4, 5		1, 2, 3
13	2	3, 5, 8		1, 2, 3
14	8	1, 4, 7		1, 2, 3
15	10	2, 6, 7		1, 2, 3
16	7	1, 2, 8		1, 2, 3
17	6	2, 6, 7		1, 2, 3
18	4	3, 5, 8	1, 2, 3	1, 2, 3
19	1	1, 6, 8		1, 2, 3
20	9	1, 5, 7		1, 2, 3

Рекомендацій

Для практичного дослідження швидкодії алгоритмів на комп'ютері рекомендується реалізовувати алгоритми на Borland Pascal 7.1 та виконувати заміри часу запустивши Borland Pascal 7.1 під DosBox, щоб унеможливити не контролюваний вплив певних процесів операційної системи на результати заміру часу. Рекомендується використовувати процедуру GetTime, яка повертає встановлений в операційній системі поточний час.

```
procedure GetTime (var години, хвилини, секунди, соти_секунди: word);

    "години" 0..23
    "хвилини"0..59
    "секунди" 0..59
    "соти_секунди" 0..99

program MYPROG;
uses dos;
var hour, min, sec, hsec : word;
begin
    {початок алгоритму}
    GetTime(hour,min,sec,hsec);
    Writeln(hour:2,'.',min:2,'.',sec:2,'.',hsec:2);
    {частина алгоритма, що досліджується}
    GetTime(hour,min,sec,hsec);
    Writeln(hour:2,'.',min:2,'.',sec:2,'.',hsec:2);
    {кінець алгоритму}
end.
```