НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ ТА СПЕЦІАЛІЗОВАНИХ КОМП’ЮТЕРНИХ СИСТЕМ

**КУРСОВА РОБОТА  
з дисципліни «Структури даних та алгоритми»**

Виконав:

Горпинич-Радуженко І.О.

Група КВ-41

Залікова книжка: КВ-4106

Допущений до захисту

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 семестр 2014/2015 навч. року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ ТА СПЕЦІАЛІЗОВАНИХ КОМП’ЮТЕРНИХ СИСТЕМ

|  |  |
| --- | --- |
| *Узгоджено*  *Керівник роботи*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Марченко О.І./* | *Захищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_р.*  *з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Марченко О.І./* |

**Дослідження ефективності методів сортування  
на багатовимірних масивах:  
Гібридний алгоритм "вставка – обмін", здійснюючи обхід з використанням додаткового одновимірного масиву;**

**Гібридний алгоритм "вставка – обмін", здійснюючи бохід перетворюючи один індекс елементів "уявного" вектора у відповідні індекси елементів заданого двовимірного масиву;**

**Гібридний алгоритм "вставка – обмін", здійснюючи обхід безпосередньо по елементах заданого двовимірного масиву**

Виконавець роботи:\_\_\_\_\_\_\_\_\_\_\_\_

Гопинич-Радуженко Іван Олександрович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_р.

**ПЛАН РОБОТИ**

1. Технічне завдання.
2. Теоретичні положення.
3. Схема імпорту/експорту модулів та структурна схема взаємовикликів процедур та функцій.
4. Опис призначення процедур та функцій.
5. Текст програми.
6. Тести.
7. Таблиці виміру часу.
8. Порівняльний аналіз.
9. Висновки.

**Технічне завдання на курсову роботу**

I. Описати принцип та схему роботи досліджуваного методу сортування для одновимірного масиву.

II. Скласти алгоритми сортування в багатовимірному масиві заданим методом, згідно з варіантом, та написати відповідну програму на мові програмування.

Програма повинна задовольняти наступні вимоги:

1.Всі алгоритми повинні бути реалізовані в рамках ОДНІЄЇ програми з діалоговим інтерфейсом для вибору варіантів тестування та виміру часу кожного алгоритму.

2.Одним з варіантів запуску програми має бути режим запуску виміру часу всіх алгоритмів у пакетному режимі, тобто запуск всіх алгоритмів для всіх випадків і побудова результуючої таблиці за наведеним нижче зразком для масиву з заданими геометричними розмірами.

3.При реалізації програми повинні бути використані модулі (unit).

4.Програма повинна мати коментарі для всіх структур даних, процедур та функцій, а також до основних смислових фрагментів алгоритмів.

III. Виконати налагодження та тестування коректності роботи написаної програми.

IV. Провести практичні дослідження швидкодії складених алгоритмів.

V. За результатами досліджень скласти порівняльні таблиці за різними ознаками.

VI. Виконати порівняльний аналіз поведінки заданих алгоритмів за отриманими результатами:

1.Для одномірного масиву відносно загальновідомої теорії.

2.Для багатовимірних масивів відносно результатів для одномірного масиву.

3.Для заданих алгоритмів на багатовимірних масивах між собою.

4.Дослідити вплив різних геометричних розмірів багатовимірних масивів на поведінку алгоритмів та їх взаємовідношення між собою.

5.Для всіх вищезазначених пунктів порівняльного аналізу пояснити, ЧОМУ алгоритми в розглянутих ситуаціях поводять себе саме так, а не інакше.

VII. Зробити висновки за виконаним порівняльним аналізом.

**Варіант №106**

**Задача**

Впорядкувати окремо кожен переріз тривимірного масиву А[p,m,n] наскрізно по рядках за незменшенням.

**Досліджувані методи та алгоритми**

1. Гібридний алгоритм №17 "вставка – обмін".

**Способи обходу**

1. Переписати елементи заданого двовимірного масиву у додатковий одновимірний масив. Виконати сортування. Повернути результат у початковий масив.
2. Не використовуючи додаткового масиву, виконати сортування перетворюючи один індекс елементів "уявного" вектора у відповідні індекси елементів заданого двовимірного масиву.
3. Виконати сортування, здійснюючи обхід безпосередньо по елементах заданого двовимірного масиву, не використовуючи додаткових масивів і перетворень індексів.

**Випадки дослідження**

1. Елементи початкового масиву впорядковані відповідно до заданої ознаки.

2. Елементи початкового масиву невпорядковані.

3. Елементи початкового масиву впорядковані за протилежно заданою ознакою.

Окремо з кожним випадком дослідження необхідно провести дослідження залежності часу виконання алгоритму від розмірів масиву:

1. Кількість перерізів: p = const = 4;

Розміри перерізу:

1. m=4; n=400;
2. m=40; n=40;
3. m=400; n=4;
4. Розміри масиву: m = const = 20; n = const = 20;

Кількість перерізів:

1. p=1;
2. p=2;
3. p=4;
4. p=8;
5. p=16;
6. Кількість перерізів: p = const = 4;

Розміри перерізу:

1. m = n = 4;
2. m = n = 8;
3. m = n = 16;
4. m = n = 32;

**ТЕОРЕТИЧНІ ПОЛОЖЕННЯ**

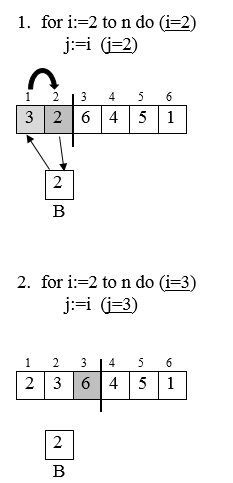
Розглянемо роботу гібридного алгоритму №17 "вставка – обмін" на прикладі сортування одновимірного масиву (вектора).

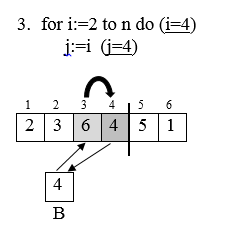
Принцип роботи:

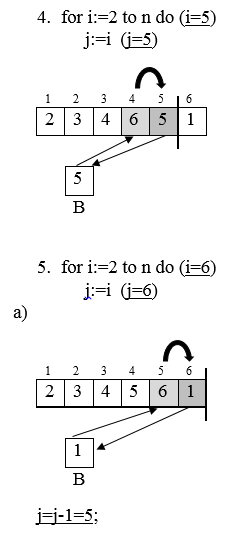
1. Починаючи з **2** і до ***n*** {for i:=2 to n}:
2. Змінна місця проходу **j** спочатку має значення лічильника, тобто **j:=i**;
3. Якщо змінна місця проходу більша за одиницю і якщо даний елемент **A[j]** менший за попередній **A[j-1]**, то:
4. Запам’ятовуємо елемент **A[j]** як **B**;
5. На місце елемента **A[j]**стає **A[j-1]**, а елементу **A[j-1]**присвоюється значення **B**, та місце проходу зменшуємо на 1.
6. Якщо змінна місця проходу дорівнює одиниці або ж даний елемент більший за попередній, то переходимо до (1).

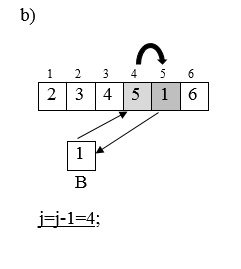
З початку кожної ітерації цикла **for** з індексом j масив складається з двох частин. Підмасив **A[1..j-1]** складається з елементів, які початково знаходились у **A[1..j-1]**, але тепер розташовані в відсортованому порядку, а елементи **A[j+1..n]** відповідають не відсортованій частині.

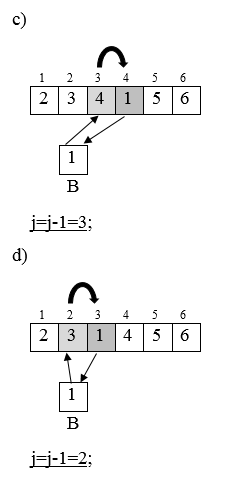
**Схема роботи алгоритму**

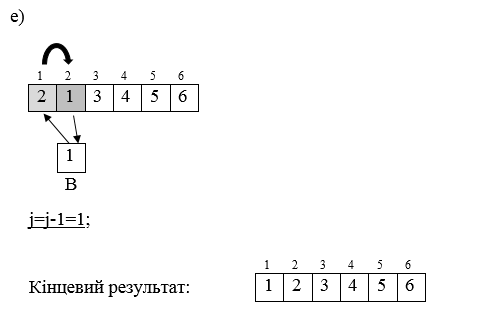
****

****

****

****

****

****

**СТРУКТУРНІ СХЕМИ ПРОГРАМИ**

Схема імпорту/експорту модулів та структурна схема взаємовикликів процедур і функцій

**GLOBAL**

**Експорт**

**ShowArrayyyyy**

**TIMER**

**Експорт**

**MAIN**

**Імпорт**

**MENU**

**GLOBAL**

**Імпорт**

**SortArray**

**UnSortArray**

**UnSortCopy**

**Time**

**SortVect**

**BackSortArray**

**UnSortVect**

**TimeVector**

**BackSortVect**

**WORKAR**

**Імпорт**

**Експорт**

**TEST**

**Імпорт**

**MENU**

**Імпорт**

**GLOBAL**

**GLOBAL**

**GLOBAL**

**WORKAR**

**WORKAR**

**WORKAR**

**1711**

**TIMER**

**WORKAR**

**1712**

**TIMER**

**TEST**

**WORKAR**

**1711**

**Main\_1**

**Експорт**

**Main\_2**

**Menu\_1**

**Експорт**

**SortVector**