 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №13**

з дисципліни «Алгоритмізація та програмування»

на тему: «Алгоритми сортування»

Варіант №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

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

(підпис викладача)

Київ-2019

**Завдання:**

1. Ознайомитися з теоретичними відомостями сортування масивів, розглянувши алгоритми бульбашкового, коктейльного сортування, сортування гребінцем, швидкого сортування, сортування простими вставками, сортування Шелла, злиттям, вибором та пірамідального сортування.
2. Побудувати блок-схему алгоритму для вирішення задачі відповідно до свого варіанту.
3. Відповідно до свого варіанту розробити програмний застосунок, що включає створення масиву цілих чисел, заповнення його елементів випадковими згенерованими значеннями з діапазону [0, 50], сортування і форматований вивід на екран початкового та відсортованого масиву (глобальні змінні не використовувати, розміри матриці задавати з клавіатури, функція main має бути призначена тільки для виклику функцій користувача):

***Мається одновимірний масив довжиною N=50. Відсортувати алгоритмом сортування Шелла елементи масиву, що знаходяться на непарних позиціях, і в такий чин, щоб вони розташовувалися за збільшенням.***

1. Скласти і захистити звіт по роботі.

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Блок-схема:**





**Код:**

#include <iostream>

#include <cstdlib>

#include <time.h>

#include <conio.h>

using namespace std;

/\*

функция для корректного ввода колличества елементов массива

нет параметров

возвращает: целочисленное значение длины массива

\*/

int number\_of\_elements(void)

{

cout << "Enter a number of elements in random array : ";

int n = 0;

while (!(cin >> n) || n <= 0)

{

cout << "Try correct number!\n";

cin.clear();

cin.ignore();

fflush(stdin);

}

return n;

}

/\*

функция для заполнения одномерного массива случайными целочисленными значениями в промежутке [0, 50]

параметры: массив для заполнения, длинна массива

ничего не возвращает, изменяет массив заполняя его случайными значениями

\*/

void fill\_array(int\* arr, int arr\_len = 50)

{

srand(time(NULL));

for (int i = 0; i < arr\_len; i++)

arr[i] = rand() % 50;

}

/\*

функция для вывода одномерного массива

параметры: массив для заполнения, длинна массива

ничего не возвращает, выводит одномерный массив

\*/

void output(int\* arr, int arr\_len = 50)

{

cout << "\nRandom array is { ";

for (int i = 0; i < arr\_len; i++)

cout << arr[i] << " ";

cout << "}\n";

}

/\*

функция для сортировки по возрастанию непарных элементов массива с помощью алгоритма сортировки Шелла

параметры: массив для заполнения, длинна массива

ничего не возвращает, сортирует массив

\*/

void shell\_sort(int\* arr, int arr\_len = 50)

{

for (int interval = (arr\_len/2) - ((arr\_len/2)%2); interval > 1; interval = (interval/2) - ((interval/2)%2))

for (int i = interval; i < arr\_len; i += 2)

{

int tmp = arr[i];

int j = 0;

for (j = i; j >= interval && arr[j - interval] > tmp; j -= interval)

arr[j] = arr[j - interval];

arr[j] = tmp;

}

cout << "\n\*Magic algorithm happened\*\n";

}

/\*

функция для создания, заполнения, вывода и сортировки по возрастанию непарных элементов массива с помощью алгоритма сортировки Шелла

параметры: длинна массива

ничего не возвращает

\*/

void array\_func(int arr\_len = 50)

{

int\* rand\_array = new int[arr\_len];

fill\_array(rand\_array, arr\_len);

output(rand\_array, arr\_len);

shell\_sort(rand\_array, arr\_len);

output(rand\_array, arr\_len);

}

void main()

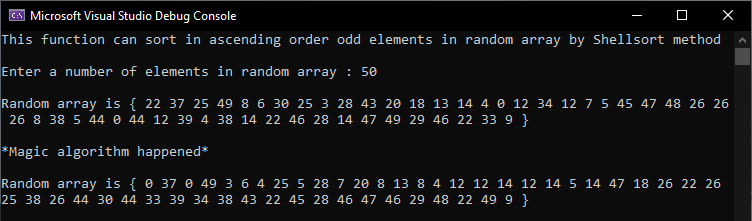
{

array\_func(number\_of\_elements());

\_getch();

}

**Результати:**

****

**Контрольні питання:**

1. ***Що таке сортування, яка його основна мета?***

У загальному випадку сортування - це процес перегрупування заданої множини об’єктів, що приводить до їх впорядкованого розташування щодо ключа.

Мета сортування - полегшити подальший пошук елементів.

1. ***Чи впливає тип даних на час виконання алгоритму сортування?***

Так, бо порівняння та інші операції з різними типами даних відрізняються за часом. Таким чином і фінальний час алгоритму сортування буде відрізнятись.

1. ***Чому не існує універсального алгоритму сортування?***

Тому що у різних ситуаціях усі алгоритми сортування мають як плюси, так і мінуси.

1. ***Чим можна пояснити різноманітність алгоритмів сортування?***

Для різних задач існують найбільш оптимальні алгоритми, які у інших задачах можуть бути менш ефективними, звідси і велика їх різноманітність.

1. ***Як визначити, яким алгоритмам сортування надати перевагу при вирішенні завдання?***

Для визначення найефективніших алгоритмів сортування, необхідно врахувати велику кількість факторів, які допоможуть вибрати певний - оптимальний: кількість даних, доступна пам’ять, типи даних, початкова сортованість, розподіл.

1. ***Які на сьогоднішній день найефективніші методи сортування?***

Quick Sort, Merge Sort, Heap Sort.

1. ***Чому алгоритми швидкого сортування не дають великого виграшу при малих розмірах масивів?***

При малих об’ємах даних, додаткові операції, такі як врахування граничних умов циклів, копіювання даних у тимчасові змінні, нівелюють швидкість роботи з масивами, які потребують мінімальну кількість дій для відсортованості.

1. ***За рахунок чого в алгоритмах швидкого сортування відбувається виграш при виконанні операцій порівняння і перестановок?***

Швидке сортування на практиці дійсно швидше ніж інші алгоритми, так як воно міняє місцями тільки ті елементи, які необхідно міняти.

1. ***У чому переваги і недоліки по відношенню один до одного наступних алгоритмів сортування: пірамідальне сортування, сортування злиттям, сортування Шелла та сортування Хоара?***

*Пірамідальне сортування*:

Переваги: має доведену оцінку гіршого випадку O ( n \ cdot \ log n ).Сортує на місці, тобто вимагає всього O ( 1 ) додаткової пам'яті.

Недоліки: складний в реалізації. Нестійкий - для забезпечення стійкості потрібно розширювати ключ.

*Сортування злиттям*:

Переваги: при витраті пам'яті O (n) швидше (O (n \ cdot \ log n) з меншою константою) і не схильна деградації на невдалих даних (порівняно з пірамідальним сортуванням)

Недоліки: через складність алгоритму виграш виходить тільки на великих n (порівняно з пірамідальним сортуванням).

*Сортування Шелла*:

Переваги: На невеликих n (до декількох тисяч) швидше за сортування злиттям- відсутність потреби в пам'яті під стек; відсутність деградації при невдалих наборах даних - швидке сортування (Хоара) легко деградує до O (n ²), що гірше, ніж найгірше гарантований час для сортування Шелла.

Недоліки: сортування Шелла в багатьох випадках повільніше, ніж швидке сортування Хоара.

*Сортування Хоара*:

Переваги: якщо розбиття збалансоване, то асимптотично алгоритм працює так само швидко як і алгоритм сортування злиттям. простий в реалізації.

Недоліки: нестійкий.

1. ***Чим відрізняється внутрішнє сортування від зовнішнього?***

Алгоритми внутрішнього сортування - працюють з даними в оперативній пам’яті з довільним доступом.

Алгоритми зовнішнього сортування - впорядковують інформацію, розташовану на зовнішніх носіях.