**Лабораторна робота № 2**

**ФБ-95 Прохоренко Ярослав**

Варіант 8

«lab2.h»

#pragma once

#include <iostream>

#include <chrono>

#include <map>

#include <string>

#define container map<string, long double>

using namespace std;

void fillArray(int\* arr, int n);

void displayArray(int\* arr, int n);

container methodOfChoice(int\* arr, int n);

container methodOfShell(int\* arr, int n);

void ConsoleLog(container results);

void selectMethod(int\* arr,int n);

«lab2.cpp»

#include "lab2.h"

void fillArray(int\* arr, int n)

{

for (int i = 0; i < n; i++)

{

arr[i] = rand() % 654;

}

}

void displayArray(int\* arr, int n)

{

char a;

cout << "\nОтобразить массив? (y/n) ";

cin >> a;

if (a != 'y') return;

cout << "\n";

for (int i = 0; i < n; i++)

{

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

}

}

container methodOfChoice(int\* arr, int n)

{

chrono::duration<double> tm;

int compare = 0;

int swaps = 0;

container \_data;

auto start = chrono::high\_resolution\_clock::now();

for (int i = 0; i < n - 1; ++i)

{

int min = i;

for (int j = i + 1; j < n; ++j)

{

if (arr[j] < arr[min])

min = j;

compare++;

}

swap(arr[i], arr[min]);

swaps++;

}

auto end = chrono::high\_resolution\_clock::now();

tm = end - start;

\_data["Time"] = tm.count();

\_data["Compare"] = compare;

\_data["Swaps"] = swaps;

return \_data;

}

container methodOfShell(int\* arr, int n)

{

chrono::duration<double> tm;

int compare = 0;

int swaps = 0;

container \_data;

auto start = chrono::high\_resolution\_clock::now();

for (int k = n / 2; k >= 1; k /= 2)

{

for (int i = k; i < n; i++)

{

for (int j = i; j >= k && arr[j - k] > arr[j]; j -= k)

{

swap(arr[j], arr[j - k]);

swaps++;

compare++;

}

}

}

auto end = chrono::high\_resolution\_clock::now();

tm = end - start;

\_data["Time"] = tm.count();

\_data["Compare"] = compare;

\_data["Swaps"] = swaps;

return \_data;

}

void ConsoleLog(container results)

{

cout << "\nРезультаты: \n - Время работы: " << to\_string(results["Time"]) << " секунд\n - Количество сравнений: " << (int)results["Compare"] << "\n - Количество обменов: " << (int)results["Swaps"] << endl;

}

void selectMethod(int\* arr, int n)

{

container results;

int ans;

cout << "\n\nВыберите метод сортировки: " << "\n1. Методом выбора" << "\n2. Методом Шелла" << "\n\n" << "> ";

cin >> ans;

switch (ans)

{

case 1:

results = methodOfChoice(arr, n);

ConsoleLog(results);

break;

case 2:

results = methodOfShell(arr, n);

ConsoleLog(results);

break;

default:

break;

}

}

«main.cpp»

#include "lab2.h"

int main()

{

srand(time(0));

setlocale(0, "ru");

int count = 0;

cout << "Введите кол-во элементов массива: ";

cin >> count;

while (cin.fail() || count < 0)

{

cin.clear();

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

cout << "Ошибка ввода! Попробуйте ещё раз\nВведите кол-во элементов массива: ";

cin >> count;

}

int\* array = new int[count];

fillArray(array, count);

displayArray(array, count);

selectMethod(array, count);

displayArray(array, count);

delete[] array;

return 0;

}

Результати порівняння методів сортування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Метод вибору | | | | |
| N | Кількість копіювань  (М) | | Кількість порівнянь  (С) | | Час |
| Теор. | Експерим. | Теор. | Експерим. |
| 100 | 99 | 99 | 4950 | 4950 | 0,000053 |
| 1000 | 999 | 999 | 4995\*102 | 4995\*102 | 0,0027 |
| 10000 | 9999 | 9999 | 49995\*103 | 49995\*103 | 0,171 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Метод Шелла | | | | |
| N | Кількість копіювань  (М) | | Кількість порівнянь  (С) | | Час |
| Теор. | Експерим. | Теор. | Експерим. |
| 100 | 251 | 371 | 251 | 371 | 0,000059 |
| 1000 | 3981 | 7145 | 3981 | 7145 | 0,001809 |
| 10000 | 63096 | 121542 | 63096 | 121542 | 0,018639 |

Як можемо бачити за даними с таблиці, прості методи сортування можуть конкурувати з швидкими методами сортування лише при малій кількості елементів. Тому доцільно використотувати прості методи сортування лише на невелику кількість елементів, якщо ж необхідно відсортувати великий масив данних, то краще підійдуть швидкі методи.

**Контакти**

Почта: [yarpro-ipt23@lll.kpi.ua](mailto:yarpro-ipt23@lll.kpi.ua) /

[prohorenko.yaroslav01@gmail.com](mailto:prohorenko.yaroslav01@gmail.com)

Телеграм: @AugFitzR