Георги Филев – F104081

Домашно 1.Б 2021-22есен.

Използвайки дадения в презентацията Lab 1 - 2020 примерен код, реализирайте измерване на времето на изпълнение на алгоритмите за сортиране пряка селекция, пряко вмъкване и мехурче. Измерете времената на всеки от тях за 10 000, 20 000, ... 100 000 случайно наредени елементи и ги въведете в електронна таблица, за да изчертаете графиката за тях.

#include <iostream>

#include <stdlib.h>

#include <time.h>

#include <sys/types.h>

using namespace std;

void swap(long int \*xp, long int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

// A function to implement bubble sort

void bubbleSort(long int arr[], long int n)

{

int i, j;

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

// Last i elements are already in place

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

if (arr[j] > arr[j + 1])

swap(&arr[j], &arr[j + 1]);

}

//

void selectionSort(long int arr[], long int n)

{

int i, j, min\_idx;

// One by one move boundary of unsorted subarray

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

{

// Find the minimum element in unsorted array

min\_idx = i;

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

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

min\_idx = j;

// Swap the found minimum element with the first element

swap(&arr[min\_idx], &arr[i]);

}

}

//

void insertionSort(long int arr[], long int n)

{

int i, key, j;

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

{

key = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > key)

{

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

/\* Function to print an array \*/

void printArray(long int arr[], long int n)

{

int i;

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

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

cout << endl;

}

int main()

{

time\_t t0, t1;

clock\_t c0, c1;

srand(time(NULL));

const long int ARRAY\_SIZE = 10000;

long int arr[ARRAY\_SIZE];

for (int i = 0; i < ARRAY\_SIZE; i++) {

arr[i] = rand() % 100 + 1;

}

t0 = time(NULL);

c0 = clock();

bubbleSort(arr, ARRAY\_SIZE);

c1 = clock();

t1 = time(NULL);

//cout << "Sorted array bubble: \n";

//printArray(arr, n);

cout << "elapsed wall clock time: " << ((long)(t1 - t0)) << endl;

cout << "elapsed wall clock time:" << (float)(c1 - c0) / CLOCKS\_PER\_SEC << endl;

//cout << "is sorted" << std::is\_sorted(a, ARRAY\_SIZE);

//selectionSort(arr, ARRAY\_SIZE);

//cout << "Sorted array selection: \n";

//printArray(arr, n);

//insertionSort(arr, ARRAY\_SIZE);

//cout << "Sorted array insertion: \n";

//printArray(arr, n);

return 0;

}

Bubble sort /time

Graphical user interface, application

Description automatically generated

Selection sort /time

A screenshot of a computer

Description automatically generated

Insertion sort /time

A screenshot of a computer

Description automatically generated