Георги Филев F104081

Домашна работа към лабораторно и практически реализации – тема 4.

1. Съставете обща графика на времето за изпълнение на два алгоритъма: 1. сортиране по дялове (Quick Sort) и 2. сортиране с пряк избор (Selection sort), базирана на мерене и записване в електронна таблица на времето за изпълнение по двата алгоритъма за входни масиви с 200, 300, 400, ... 1000 елемента със случайни стойнности (един и същи случаен набор се подава на двата алгоритъма за всяко генериране на случаен масив).

#include <iostream>

#include <stdlib.h>

#include <time.h>

#include <sys/types.h>

using namespace std;

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

int i,key,j;

for(int 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;

}

}

void QSort(int arr[], int li ,int di){

int j=li , k = di;

if(di > li){

int etalon = arr[(di+li)/2];

do{

while(arr[j]<etalon){

j++;

}

while(arr[k] > etalon){

k--;

}

if(j<=k){

int temp = arr[k];

arr[k]= arr[j];

arr[j]= temp;

k--;

j++;

}

}while(j<=k);

QSort(arr,li,k);

QSort(arr,j,di);

}

else{

cout<<" napuskame cikula"<<endl;

}

}

int main()

{

time\_t t0, t1;

clock\_t c0, c1;

srand(time(NULL));

int arrSize = 2000;

int arr[arrSize] ;

for(int i =0; i<2000; i++){

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

cout<< arr[i];

}

t0 = time(NULL);

c0 =clock();

for(int i =0 ;i <=2000; i++){

insertionSort(arr,i);

i+=500;

}

c1 = clock();

t1 = time(NULL);

cout << endl;

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

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

for(int i =0; i<2000; i++){

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

cout<< arr[i];

}

t0 = time(NULL);

c0 =clock();

for(int i =0 ;i <=2000; i++){

QSort(arr,0 ,i);

i+=500;

}

t1 = time(NULL);

c1 = clock();

cout << endl;

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

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

return 0;

}

Text

Description automatically generated

A picture containing text, screenshot, computer, indoor

Description automatically generated

1. Съставете по същия начин обща графика на времето за изпълнение при вход от **нареден**и масиви с 200, 300 и т.н. елементи, на 1. сортиране по дялове, като еталонът е най-левия елемент от дяла и 2. сортиране на същия нареден масив с пряк избор. (можете да ползвате за вход изхода от задача 1).

A picture containing text, screenshot, computer, monitor

Description automatically generated

Приложете код, изход и графиките.

1. Коментирайте писмено резултатите, като обосновете разсъжденията си със специфичните особености на двата сравнявани алгоритъма и това дали те се влияят от състоянието на входните данни.

Резултата се влияе от входните данни със сигурност ,защото колкото по голям е входа толкова повече време отнема на алгоритъма да я обработи , но QSort се държи много по добре във времето, като логаритъм при основа 2, Но ако взимаме най-левия елемент при нареден масив тормозим алгоритъма да прави най много сравнения като n квадрат.

1. Съставете програма за сливане на два наредени масива в трети, на изхода от програмата да се разпачатват сливаните масиви, как последователно се вземат стайностите от единия и от другия масив , както и слетият и нареден масив.

void Merge(int li , int di){

int j = li, k = (li+di)/2 +1;

int r = li;

while(j<(li + di)/2 +1 && k<di + 1){

if(arr[j] < arr[k]){

Carr[r] = arr[j];

j++;

}

else{

Carr[r] = arr[k];

k++;

}

r++;

}

for(int i = j; i<(li+di)/2 + 1; i++){

Carr[r]= arr[i];

r++;

}

for(int i =k; i < di+1;i++){

Carr[r]= arr[i];

}

for(int i = li; di+1; i++){

cout<<Carr[i]<<" "<< endl;

arr[i] =Carr[i];

}

}

int main()

{

Merge(0,9);

return 0;

}

(не успявам да засека времето за изпълнение)посоянно е 0 дори с по големи стойности

Съставете слеването по два варинта – 1. с междинна проверка за това кой от двата масива се е изчерпал и 2. Направао два последователни цикъла за пренасяна на стойностите от неизчелпания масив в слетия, като измерите време за изполнение на двата варианта.