**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Algorytmy i struktury danych | **Zadanie** | 11\_02 |
| **Autor** | Kamil Borkowski | **Grupa** | WCY22IY1S1 |
| **Temat** | Łączenie naturalne | | |

1. Treść

Zaimplementuj algorytm sortowania metodą łączenia naturalnego.

* 1. Metoda realizacji

Program wykorzystuje prosty interfejs użytkownika do wyboru operacji, wczytywania danych i sortowania za pomocą algorytmu łączenia naturalnego.

* 1. Założenia / ograniczenia dotyczące danych:
     1. Dane wejściowe *(opis)*

Wartości do sortowania – wczytywana z pliku/klawiatury/generatora

* + 1. Dane wyjściowe *(opis)*

Wartości posortowane – wyświetlone na ekranie.

1. Realizacja
   1. Algorytm *(lista kroków lub schemat blokowy)*
2. Utworzenie tablic pomocniczych arr1 i arr2, które będą przechowywać dane podczas procesu sortowania.
3. Przechodzenie przez wejściową tablicę arr.
4. Elementy są dodawane naprzemiennie do arr1 i arr2 zgodnie z regułą łaczenia naturalnego.
5. Gdy natrafi na punkt, w którym wartość następnego elementu jest mniejsza niż obecnej, przechodzi do uzupełniania drugiej tablicy.
6. Porównywanie elementów na odpowiednich pozycjach w arr1 i arr2.
7. Mniejszy element jest przenoszony do wejściowej tablicy arr.
8. Po scaleniu jednej pary tablic pomocniczych, funkcja sprawdza, czy cała tablica jest posortowana. Jeśli nie, proces sortowania jest powtarzany rekurencyjnie.

Kod źródłowy

//================================================

//Zadanie 11\_02 jezyk C++

//Laczenie Naturalne

//WCY22IY1S1 Borkowski Kamil

//================================================

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <cstdio>

#include <fstream>

using namespace std;

//funkcja pomocnicza sprawdzajaca czy tablica jest juz posortowana

bool czyPosortowana(int arr[], int n)

{

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

{

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

{

return false; // Znaleziono niesortowaną parę

}

}

return true; // Tablica jest posortowana

}

//funkcja laczenia naturalnego sort

void laczNat(int arr[], int n)

{

int arr1[10000] = {};//tablice pomocnicze

int arr2[10000] = {};

int n1=0,n2=0;

bool piszdoa1=true;

for (int i = 0; i < n; i++)//uzupelnianie tablic pomocniczych

{

if (piszdoa1)//uzupelnianie arr1

{

arr1[n1] = arr[i];

n1++;

if (i < n - 1 && arr[i + 1] < arr[i])

piszdoa1 = false;

}

else//uzupelnianie arr2

{

arr2[n2] = arr[i];

n2++;

if (i < n - 1 && arr[i + 1] < arr[i])

piszdoa1 = true;

}

}

int pom1 = 0, pom2 = 0;//zmienne pomocnicze do indeksow tablic pomocniczych

for (int i = 0; i < n; i++)//scalanie tablic pomocniczych

{

if (pom1 < n1 && (pom2 >= n2 || arr1[pom1] < arr2[pom2]))

{

arr[i] = arr1[pom1];

pom1++;

}

else

{

arr[i] = arr2[pom2];

pom2++;

}

}

//czy posortowana

if(!czyPosortowana(arr,n))laczNat(arr,n);

}

int main()

{

srand((time(0)));

int arr[10000] = {};

int i,k=0,n,g,d,w;

int pom=1,z;//pom oblusguje interfejs, z= nr komendy z interfejsu

cout << "WCY22IY1S1 Borkowski Kamil" << endl;

cout << "Program sortowania Laczenia Naturalnego" << endl;

cout << "[0] - wyjscie z programu" << endl;

cout << "[1] - wczytaj dane z pliku(we.txt)" << endl;

cout << "[2] - wczytaj dane z generatora" << endl;

cout << "[3] - wczytaj dane z klawiatury" << endl;

cout << "=========================================" << endl;

while(pom){

cout << "Wprowadz nr operacji: ";

cin >> z;

switch (z){

case 0:

cout<<"Wychodzenie..."<<endl;

pom = 0;

break;

case 1:

{

k=0;

ifstream in;

in.open("we.txt");

int p;

i=0;

while(in>>p)

{

arr[i]=p;

i++;

}

n=i;

cout<<"Tablica zostala usupelniona danymi z pliku we.txt"<<endl<<endl;

cout <<"Przed sortowaniem metoda laczenia naturalnego: "<<endl;;

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

{

cout<<arr[i]<<'\t';

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

laczNat(arr,n);

k=0;

cout<<endl<<"Po sortowaniu metoda laczenia naturalnego: "<<endl;;

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

cout << arr[i] <<'\t';

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

cout<<endl;

in.close();

break;

}

case 2:

{

k=0;

cout<<"Podaj ilosc liczb do sortowania"<<endl;

cin>>n;

cout<<"Podaj zakres liczb do sortowania"<<endl;

cout<<"Granica dolna: ";

cin>>d;

cout<<"Granica gorna: ";

cin>>g;

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

arr[i]=(rand() % (g-d+1))+d; // liczby od 1 do 100

}

cout <<"Przed sortowaniem metoda laczenia naturalnego: "<<endl;;

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

{

cout<<arr[i]<<'\t';

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

laczNat(arr,n);

k=0;

cout<<endl<<"Po sortowaniu metoda laczenia naturalnego: "<<endl;;

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

cout << arr[i] <<'\t';

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

cout<<endl;

break;

}

case 3:

{

k=0;

cout<<"Podaj ilosc liczb do sortowania"<<endl;

cin>>n;

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

{

w=0;

cout<<"Podaj argument: ";

cin>>w;

arr[i]=w;

}

cout <<"Przed sortowaniem metoda laczenia naturalnego: "<<endl;;

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

{

cout<<arr[i]<<'\t';

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

laczNat(arr,n);

k=0;

cout<<endl<<"Po sortowaniu metoda laczenia naturalnego: "<<endl;;

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

cout << arr[i] <<'\t';

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

cout<<endl;

break;

}

default:

{

cout<<"Podano bledny nr operacji"<<endl<<endl;

break;

}

}

}

return 0;

}

* 1. Dane wejściowe *(ekran uruchomienia programu)*

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* 1. Dane wyjściowe *(ekran uruchomienia programu)*

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

1. Wnioski *(złożoność asymptotyczna algorytmu przy użyciu notacji* ***O lub innej i inne wnioski)***

Złożoność operacyjna algorytmu sortowania przez łączenie naturalne wynosi O(n log n).