**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Algorytmy i struktury danych | **Zadanie** | 10\_01 |
| **Autor** | Kamil Borkowski | **Grupa** | WCY22IY1S1 |
| **Temat** | Heap Sort | | |

1. Treść

Zaimplementuj algorytm sortowania metodą Heap Sort.

* 1. Metoda realizacji

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

* 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. Funkcja Heap Sort wywołuje funkcję NaKop do aranżowania elementów tablicy.
3. Pierwsza pętla heapSort inicjalizuje kopiec, druga pętla sortuje tablice

Kod źródłowy

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

//Zadanie 10\_01 jezyk C++

//Heap Sort

//WCY22IY1S1 Borkowski Kamil

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

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <cstdio>

#include <fstream>

using namespace std;

//funkcja pomocnicza przearanzujaca elementy tablicy w kolejnosci kopca

void NaKop(int arr[], int n, int i)

{

int pom;//pomocnicza do zamiany elementow

int max = i;//najwiekszy element jako root

int l = 2 \* i + 1;//lewy potomek

int r = 2 \* i + 2;//prawy potomek

if (l < n && arr[l] > arr[max])//gdy lewy potomek jest wiekszy od roota

max = l;

if (r < n && arr[r] > arr[max])//gdy prawy potomek jest wiekszy od najwiekszego

max = r;

if (max != i) {//gdy najwiekszy nie jest rootem

pom=arr[i];//zamiena elementow

arr[i]=arr[max];

arr[max]=pom;

NaKop(arr, n, max);//przearanzuj pozostale elementy w kolejnosci kopca

}

}

//funkcja heap sort

void heapSort(int arr[], int n)

{

int pom;

for (int i = n / 2 - 1; i >= 0; i--)//przearanzuj elementy w kolejnosci kopca

NaKop(arr, n, i);

for (int i = n - 1; i > 0; i--) {//wybierz element z kopca

pom=arr[i];//zamien element z rootem

arr[i]=arr[0];

arr[0]=pom;

NaKop(arr, i, 0);//przearanzuj pozostale elementy w kolejnosci kopca

}

}

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 Heap Sort" << 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 heap sort: "<<endl;;

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

{

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

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

heapSort(arr,n);

k=0;

cout<<endl<<"Po sortowaniu metoda heap sort: "<<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 heap sort: "<<endl;;

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

{

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

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

heapSort(arr,n);

k=0;

cout<<endl<<"Po sortowaniu metoda heap sort: "<<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 heap sort: "<<endl;;

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

{

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

k++;

if(k==10)

{

k=0;

cout<<endl;

}

}

heapSort(arr,n);

k=0;

cout<<endl<<"Po sortowaniu metoda heap sort: "<<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, oprogramowanie

Opis wygenerowany automatycznie

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

Złożoność obliczeniowa programu jest zdominowana przez złożoność sortowania przez kopcowanie (Heap Sort) i wynosi O(n log n).