Universitatea de Stat din Tiraspol

Facultatea Fizică, Matematică și Tehnologii Informaționale

Catedra Informatică și Tehnologii Informaționale

Programele de studiu: Matematică și Informatică, Informatică și Matematică, Informatică, Fizică și Informatică

Grupa academică: 2MI

Numele și prenume studentului: Razloga Anastasia

Evaluarea 2

la cursul Structuri de date și algoritmi

Varianta 1

*1. Scrieți un program care, utilizând o funcție recursivă, va afișa reprezentarea binară a numărului natural dat n (n < 30 000).*

#include<iostream>

using namespace std;

void fb (int n) {

if (n>0) {

fb (n/2);cout << " " << n%2;

}

}

int main(){

int n;

cout<<"( n<30 000 ) n = " ; cin >> n;

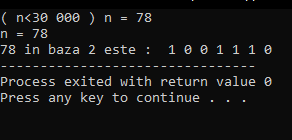
cout << "n = "<< n << endl;

cout << n << " in baza 2 este : ";

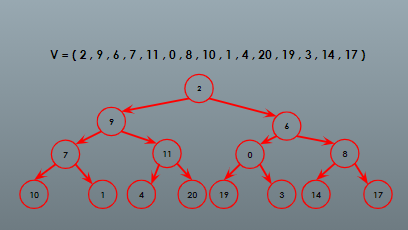
if (n) fb(n); else cout <<"0";

return 0;

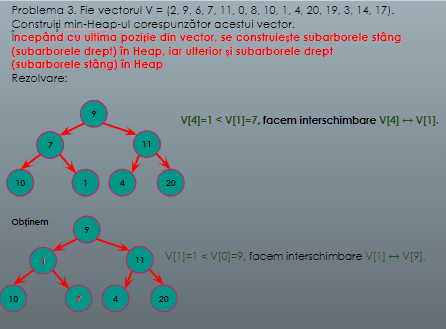
}

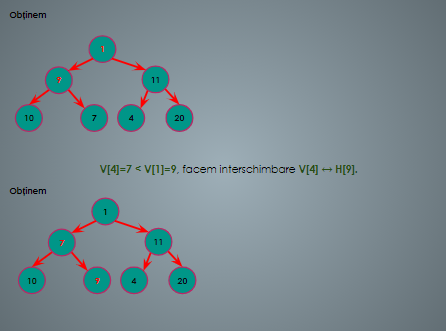
****

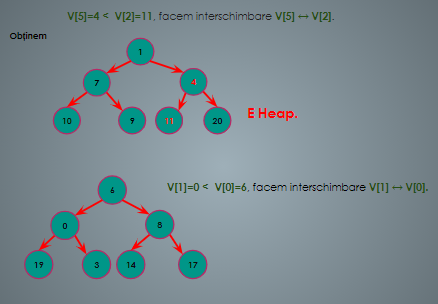
*2. Vectorul V = (2, 9, 6, 7, 11, 0, 8, 10, 1, 4, 20, 19, 3, 14, 17) este reprezentarea liniară a arborelui binar complet A. Reprezentați grafic arborele A.*

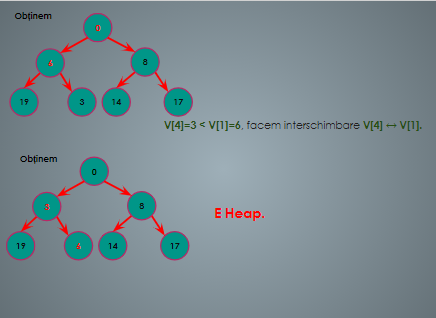
**

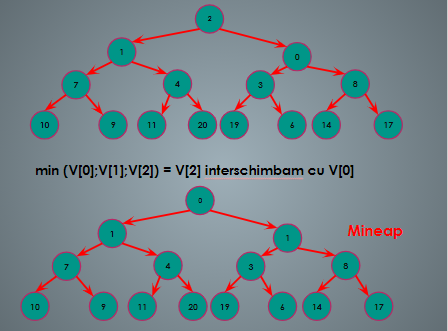
*3. Fie vectorul V = (2, 9, 6, 7, 11, 0, 8, 10, 1, 4, 20, 19, 3, 14, 17). Construiți min-Heap-ul corespunzător acestui vector.*

**

**

**

**

**

*4. Se dau coordonatele carteziene ale punctelor A, B, C și D. Să se scrie un algoritm care verifică dacă punctul D aparține interiorului triunghiului ABC.*

#include <iostream>

using namespace std;

typedef struct t\_point {

double x, y;

} TPoint;

double ab(TPoint X, TPoint Y, TPoint Z)

{

return (Y.y - Z.y) \* X.x + (Z.x - Y.x) \* X.y + Y.x \* Z.y - Y.y \* Z.x;

}

int main()

{

int n = 3;

TPoint M;

TPoint \*A;

bool f;

cout << "Coordonatele A , B , C: " << endl ;

A = new TPoint[n + 2];

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

cout << "Coordonata x"<< i+1 << ": ";

cin >> A[i].x;

cout << "Coordonata Y"<< i+1 << ": ";

cin >> A[i].y;

}

cout << "Coordonatele punctului D : " ;

cin >> M.x >> M.y;

f = true;

A[n].x = A[0].x;

A[n].y = A[0].y;

A[n + 1].x = A[1].x;

A[n + 1].y = A[1].y;

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

if (ab(M, A[i], A[i + 1]) \* ab(A[i+2], A[i], A[i+1]) <= 0) {

f = false;

}

}

if (f) {

cout << "DA";

} else {

cout << "NU";

}

return 0;

}

