

Tema 3 - Introducere in programare

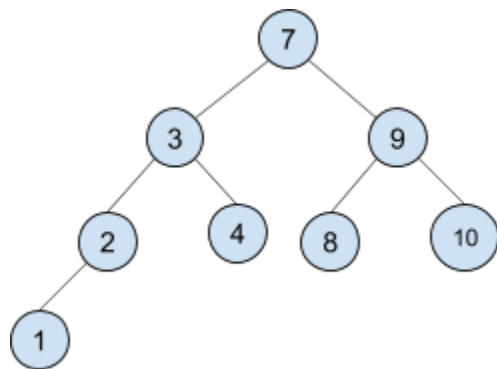
(3 pct) 1. Se primește un vector de elemente, pe baza carora trebuie construit un arbore binar de cautare. Sa se implementeze functia `nrNoduriSpeciale` (data mai jos) care returneaza numarul de noduri speciale din arborele creat. Daca un nod are exact doi fii nenuli, ce contin numere pare, vom spune ca acel nod este "special".

```
int nrNoduriSpeciale(int vectorNoduri[MAX_NODURI], int nrNoduri);
```

Exemplu:

Pentru `vectorNoduri[]={7,3,9,2,8,1,4,10}`;

Arborele creat va fi:



7 fiind prima valoare este radacina arborelui. 3 va fi adaugat in stanga $3 < 7$, 9 in dreapta lui 7 $9 > 7$ etc.

3 are exact doi fii (2,4) ambii avand valori pare deci este un nod "special"

9 are exact doi fii (8,10) ambii avand valori pare deci este un nod "special"

Rezultatul este asadar 2

(3 pct) 2. Se dau `nrListe` liste simplu inlantuite. Sa se combine listele intr-o singura lista ce are toate elementele listelor in ordine crescatoare.

```
struct nod {
    int val;
    nod* urm;
};
nod* combinareSortareListe(int nrListe, nod* liste[MAX_LISTE]);
```

(3pct) **3. Se da un graf in formatul explicat mai jos. Sa se construiasca grafurile si sa se returneze toate subgrafurile conexe de ordin dat ce contin noduri de valoare para din graf.**

```
struct compConexe {  
    int nrComponente;  
    int componente[100][100];  
};
```

```
struct grafListaMuchii {  
    int nrNoduri;  
    int nrMuchii;  
    int muchii[2][MAX_MUCHII];  
};
```

```
compConexe CompConexe(grafListaMuchii graf, int ordinCompConexe);
```

grafListaMuchii.nrNoduri -numarul nodurilor din graf

grafListaMuchii.nrMuchii - numarul muchiilor din graf

grafListaMuchii.muchii[][] -o matrice cu 2 linii in care pe fiecare coloana este specificata o muchie din graf

ordinCompConexe - se vor cauta componentele conexe de acest ordin

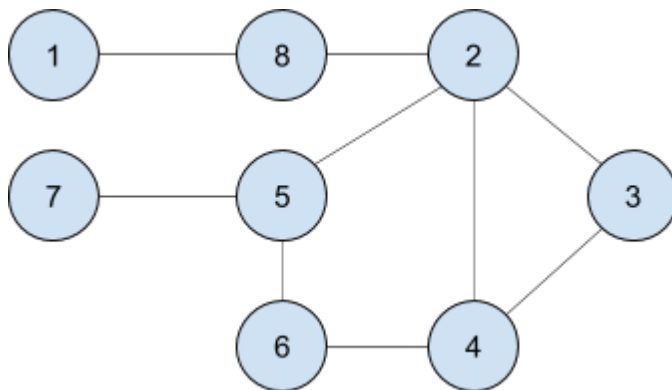
compConexe - lista componentelor conexe de returnat

compConexe.nrComponente - numarul de componente conexe gasite

compConexe.componente - matrice in care pe fiecare linie se afla nodurile din componenta conexa

Exemplu:

Pentru graful de jos si ordinCompConexe=3



Subgraful format din nodurile 8,2,4 este o componenta conexa cu toate nodurile pare. De asemenea si subgraful 2,4,6 este corect.

Prin urmare in structura rezultata:
compConexe.nrComponente=2
compConexe.componente[0][0]=8
compConexe.componente[0][1]=2
compConexe.componente[0][2]=4
compConexe.componente[1][0]=2
compConexe.componente[1][1]=4
compConexe.componente[1][2]=6

Observatie:

Orice ordine pentru nodurile returnate este corecta, 8,2,4 = 8,4,2 = 2,4,8 = 2,8,4 = 4,2,8 = 4,8,2. Returnati doar una din variante.

(1 pct) 4. Se da un graf neorientat in formatul explicat mai jos. Sa se gaseasca un arbore binar (subgraf al grafului dat) cu numar de noduri maxim, sa se creeze folosind structura ArbBin si sa se returneze.

```
struct ArbBin {
    int val;
    ArbBin* st;
    ArbBin* dr;
};
```

```
ArbBin* maxArbBin(int nrNoduri, int nrMuchii, int muchii[2][MAX_MUCHII]);
```

Mai jos aveti exemplu de un graf si doi posibili arbori binari valizi.

