

Arbore cu rădăcină. Metode de reprezentare

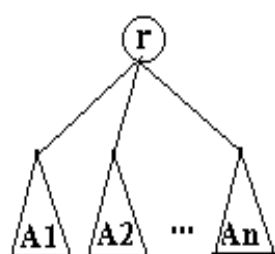
Definiție

Un **arbore cu rădăcină** este o structură de date constituită dintr-o mulțime finită de elemente numite **noduri**, mulțime care îndeplinește una din condițiile următoare:

1. este vidă → este cazul **arborelui vid** sau **arborelui nul**;
2. conține un nod special denumit **rădăcina arborelui**, toate celelalte noduri fiind partiționate în $n \geq 0$ arbori cu rădăcină **A1, A2, A3, ..., An**.

Arborii **A1, A2, A3, ..., An** sunt denumiți subarbori ai nodului rădăcină.

În **Figura 1** avem reprezentarea grafică a unui arbore cu rădăcină.



r este nodul rădăcină

Nodurile conținute în subarbori sunt considerate **descendenți** ai **nodului rădăcină**.

Rădăcinile arborilor **A1, A2, A3, ..., An** sunt considerate **descendenți direcți** sau **fii** ai rădăcinii **r** a arborelui.

Figura 1: Arbore cu rădăcină

Terminologie

- ✚ muchiile unui arbore cu rădăcină se mai numesc **ramuri** sau **arce**;
- ✚ alte denumiri pentru nodul rădăcină sunt **tulpină** sau **vârf**;
- ✚ într-un nod intră un singur arc care îl leagă de un alt nod numit **părintele** sau **predecesorul** său, excepție făcând nodul rădăcină în care nu intră nici un arc;
- ✚ dintr-un nod pot să iasă nici un arc, unul sau mai multe arce care îl leagă de un alt nod numit **fiu** sau **succesor**
- ✚ există un drum unic de la rădăcina **r** a arborelui la fiecare nod **i** din arbore. Dacă nodul **j** se găsește pe drumul unic de la rădăcină la nodul **i**, atunci **j** se numește **ascendent**(**strămoș**);
- ✚ **nodurile terminale**(sau **frunzele**) reprezintă nodurile care nu au descendenți. Nodurile care nu sunt terminale se numesc **noduri de ramificare**;
- ✚ două noduri care au același părinte se numesc **frați**.

Definiție

Pentru un arbore cu rădăcină **ordinul unui nod** reprezintă numărul de **descendenți direcți**(**fii**).

Nodurile unui arbore cu rădăcină sunt organizate pe niveluri.

Fie un arbore cu rădăcină reprezentat în **Figura 2**.

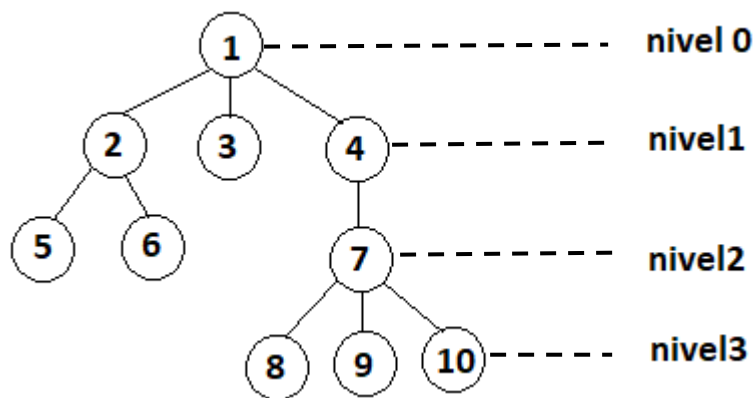


Figura 2 Nivelurile unui arbore cu rădăcină

Rădăcina arborelui este nodul 1 și constituie nivelul 0 în arbore

Fiii rădăcinii sunt nodurile 2, 3 și 4 și constituie nivelul 1 în arbore

Fiii fiilor nodului rădăcină sunt nodurile 5, 6 și 7 ce se găsesc pe nivelul 2 în arbore

-
-
-

Fiii nodurilor de pe nivelul i constituie nivelul i+1 în arbore

Definiție1

Înălțimea unui arbore cu rădăcină reprezintă nivelul maxim în arbore

Definiție2

Înălțimea unui arbore cu rădăcină se poate calcula astfel: $1 +$ valoarea maximă dintre înălțimile subarborilor săi $A_1, A_2, A_3, \dots, A_n$.

Pentru arborele cu rădăcină de mai sus înălțimea = 3

Pentru **Figura 2** mai putem adăuga:

Nodul 2 este predecesorul nodurilor 5 și 6

Nodul 8 este succesorul nodului 7

Nodul 4 este ascendent al nodului 8

Nodurile 7, 8, 9 și 10 sunt descendenți ai nodului 4

Nodurile terminale sunt 3, 5, 6, 8, 9 și 10

Metode de reprezentare a arborilor cu rădăcină

Pentru un arbore cu rădăcină un nod poate avea mai mulți succesori, însă nu poate avea decât un predecesor (tată). Deci implementarea structurii de date de tip arbore cu rădăcină se poate face static, prin referințe ascendente.

Se va utiliza un vector cu numele **tata** în care memorăm pentru fiecare **nod i** eticheta nodului **părinte j**.

Astfel **elementul tata[i]** poate lua una dintre valorile:

- **j** dacă **nodul j** este **tatăl nodului i**
- **0** dacă **nodul i** este **rădăcina arborelui**

Pentru arborele din **Figura 2** valorile elementelor vectorului **tata** sunt:

Vectorul **tata**

0	1	1	1	2	2	4	7	7	7
---	---	---	---	---	---	---	---	---	---

Nodul **i**

1 2 3 4 5 6 7 8 9 10

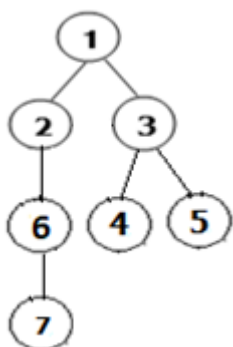
Observații

- ✚ Nodul rădăcină este **nodul i** pentru care **tata[i]=0**;
- ✚ Nodurile frunză sunt nodurile a căror **etichetă nu există** în vectorul **tata**.

Aplicații

1. Din fișierul **graf.in** se citesc de pe prima linie numărul de **noduri n**, iar de pe următoarele **n linii** matricea de adiacență a unui graf neorientat care reprezintă un **arbore**. De la tastatură se citește un **număr natural r**, $1 \leq r \leq n$. Se cere să se afișeze pe ecran vectorul **tata**, corespunzător **arborelui cu rădăcină r**.

Ex



graf.in 7 0 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 De la tastatură r=1	Se va afisa pe ecran: Vectorul tata contine: 0 1 1 3 3 2 6
---	--

Rezolvare:

Se realizează o parcurgere în adâncime DF, de la grafuri neorientate, cu nodul de pornire $np=r$. De câte ori se selectează o muchie se completează datele din vectorul tata:

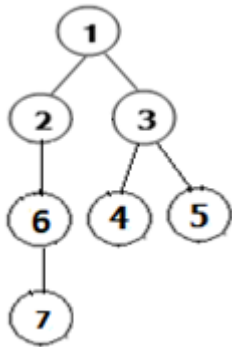
```
void DF(int np)
{
    int i;
    viz[np]=1;
    for(i=1;i<=n;i++)
        if(a[np][i]==1&&viz[i]==0)
            {
                tata[i]=np;
                DF(i);
            }
}
```

Programul C++ este următorul. Realizați-l pe calculator.

<pre>#include<iostream> #include<fstream> using namespace std; ifstream fin("graf.in"); int a[30][30],viz[30],tata[30],n,r; void CitireMatr() { int i,j; fin>>n; for(i=1; i<=n; i++) for(j=1; j<=n; j++) fin>>a[i][j]; fin.close(); }</pre>	<pre>void DF(int np) { int i; viz[np]=1; for(i=1;i<=n;i++) if(a[np][i]==1&&viz[i]==0) { tata[i]=np; DF(i); } } int main () { int i; CitireMatr(); cout<<"Radacina="<<cin>>r; DF(r); cout<<"Vectorul tata contine:\n"; for(i=1; i<=n; i++) cout<<tata[i]<<" "; return 0; }</pre>
---	--

2. Se dă un arbore cu rădăcină pentru care se cunosc **numărul de noduri n**, **matricea de adiacență** și **rădăcina r** (citire din fișierul **date.in**). Să se afișeze în fișierul **date.out**, pentru fiecare nod în parte, **nivelul** pe care acesta se află.

Ex



<u>date.in</u>	<u>date.out</u>
7	Nodul 1 niv 0
0 1 1 0 0 0 0	Nodul 2 niv 1
1 0 0 0 0 1 0	Nodul 3 niv 1
1 0 0 1 1 0 0	Nodul 4 niv 2
0 0 1 0 0 0 0	Nodul 5 niv 2
0 0 1 0 0 0 0	Nodul 6 niv 2
0 1 0 0 0 0 1	Nodul 7 niv 3
0 0 0 0 0 1 0	
1	

Se utilizează un vector **niv** cu **n** elemente și se realizează o **parcurgere în lățime BF**, de la grafuri neorientate cu nodul de pornire **np=r**.

Pentru fiecare **vecin i găsit**, al nodului **c[pi]**:

- Se introduce în coadă **c[pi]=i**;
- Se marchează ca fiind vizitat **viz[i]=1**;
- Se dă elementului **niv** valoarea lui **niv[c[pi]]+1**: **niv[c[pi]]=niv[c[pi]]+1**;

La final se afișează valorile vectorului **niv**.

Programul C++ este următorul. Realizați-l pe calculator.

<pre>#include<fstream> using namespace std; ifstream fin("date.in"); ofstream fout("date.out"); int a[30][30],viz[30],c[30],niv[30],n,r; void CitireMatr() { int i,j; fin>>n; for(i=1; i<=n; i++) for(j=1; j<=n; j++) fin>>a[i][j]; fin>>r; fin.close(); } void BF(int np) {int pi,pf,i; pi=1; pf=1; c[pi]=np; viz[np]=1;</pre>	<pre>while(pi<=pf) { i=1; while(i<=n) { if(a[c[pi]][i]==1&&viz[i]==0) { pf++; c[pf]=i; viz[i]=1; niv[c[pf]]=niv[c[pi]]+1; } i++; } pi++; } int main() { int i; CitireMatr(); BF(r); for(i=1; i<=n; i++) fout<<"Nodul "<<i<<" niv "<<niv[i]<<"\n"; fout.close(); return 0; }</pre>
--	---