

# Algoritmi pe grafuri - 1

Ștefan Trăușan-Matu

## Tipuri de grafuri

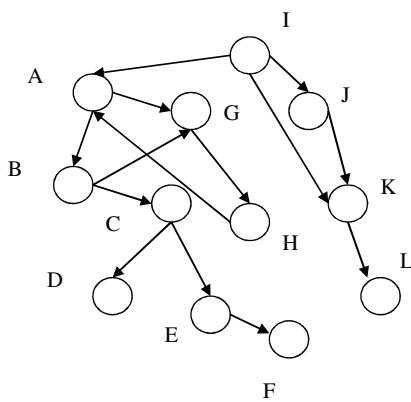
- Orientate
- Neorientate
- DAG (aciclice orientate)
- Cu ponderi (lungime, cost...) pe arce
- Ordonate
- Multigrafuri
- Arbori, păduri, liste

## Reprezentarea internă a grafurilor

- Matrice de adiacență
- Vector de adiacență
- Perechi de arce
- .....?

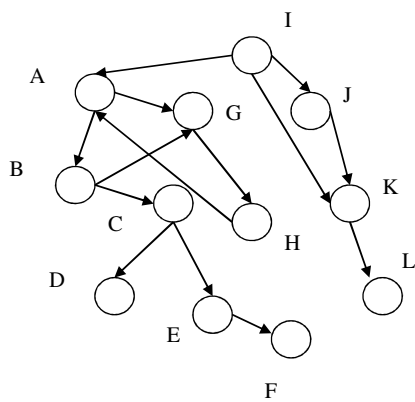
# Matrice de adiacență

Matricea este rară!



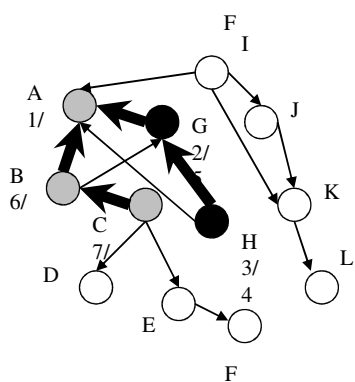
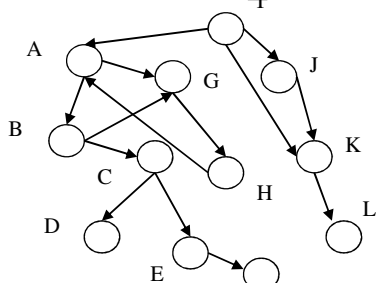
	A	B	C	D	E	F	G	H	I	J	K	L
A		1					1					
B			1				1					
C				1	1							
D												
E						1						
F												
G								1				
H	1											
I	1									1	1	
J											1	
K												1
L												

## Vector de adiacență



<b>A</b>	<b>G</b>	<b>B</b>	
<b>B</b>	<b>C</b>	<b>G</b>	
<b>C</b>	<b>D</b>	<b>E</b>	
<b>D</b>			
<b>E</b>	<b>F</b>		
<b>F</b>			
<b>G</b>	<b>H</b>		
<b>H</b>	<b>A</b>		
<b>I</b>	<b>A</b>	<b>J</b>	<b>K</b>
<b>J</b>	<b>K</b>		
<b>K</b>	<b>L</b>		
<b>L</b>			

## O posibilă reprezentare

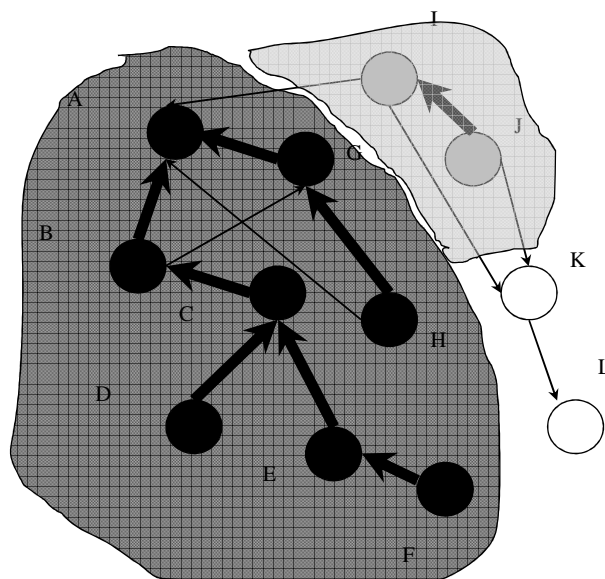


graf=[	
['a','alb','nil',' ',' ','[6,1]],	0
['b','alb','nil',' ',' ','[2,6]],	1
['c','alb','nil',' ',' ','[3,4]],	2
['d','alb','nil',' ',' ','[]],	3
['e','alb','nil',' ',' ','[5]],	4
['f','alb','nil',' ',' ','[]],	5
['g','alb','nil',' ',' ','[7]],	6
['h','alb','nil',' ',' ','[0]],	7
['i','alb','nil',' ',' ','[0,9,10]],	8
['j','alb','nil',' ',' ','[10]],	9
['k','alb','nil',' ',' ','[11]],	10
['l','alb','nil',' ',' ','[]],	11

[nod, culoare, tată, debut, finis, lista\_adiacențe]

## Parcurgeri sistematice ("scanări")

- În adâncime
- În lățime
- În adâncime iterativ
- După "optim"
- .....

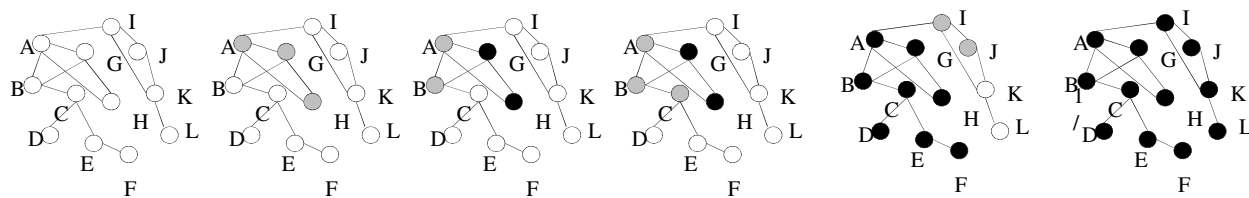


## Parcurgerea în adâncime

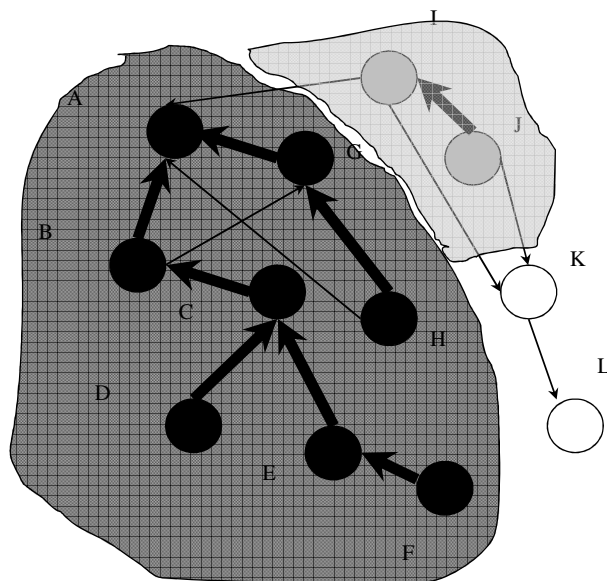
```
def parc_ad(g,n):  
    for i in range(0,n-1):  
        g[i][1]='alb'  
        g[i][2]='nil'  
    for i in range(0,n-1):  
        if culoare(i)=='alb':  
            parc_ad_rec(g,i)
```

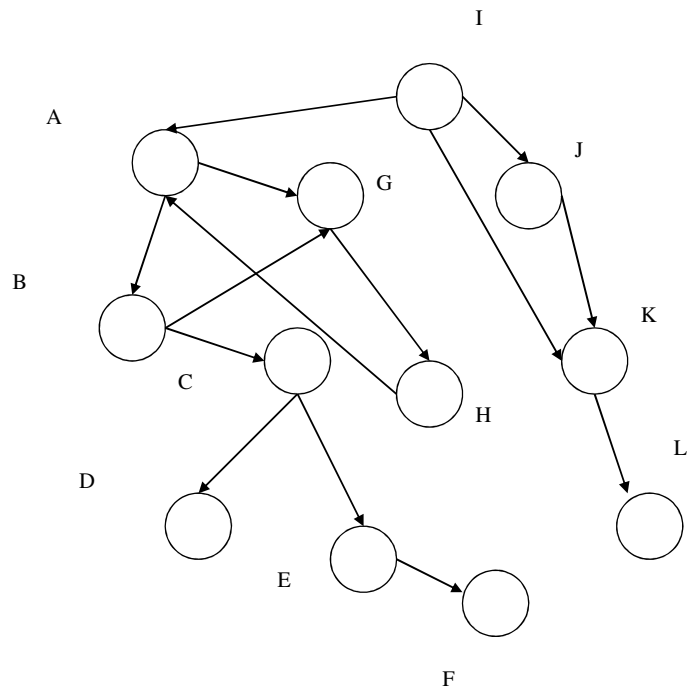
```
def parc_ad_rec(g,i):  
    graf[i][3]=timp  
    timp+=1  
    graf[i][1]='gri'  
    for j in adiacente (i):  
        if culoare(j)=='alb':  
            pune_tata(j,i)  
            parc_ad_rec(g,j)  
    graf[i][4]=timp  
    timp+=1  
    graf[i][1]='negru'
```

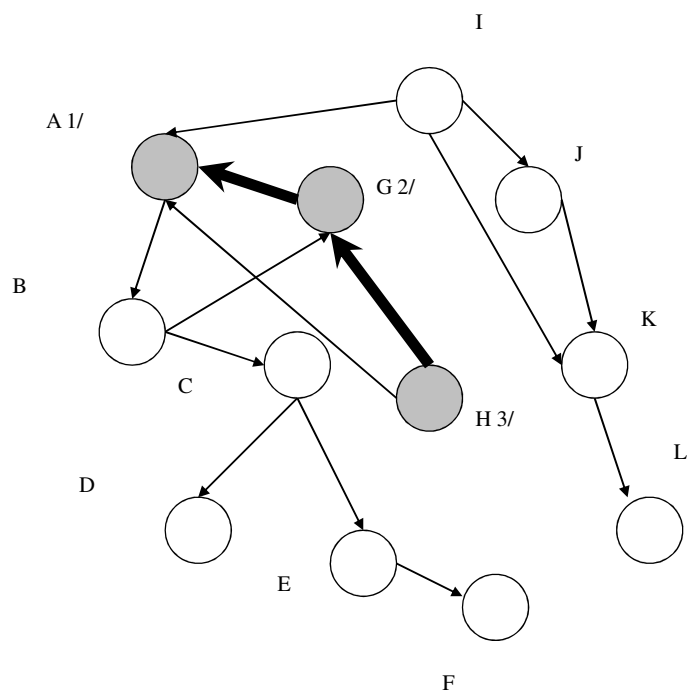


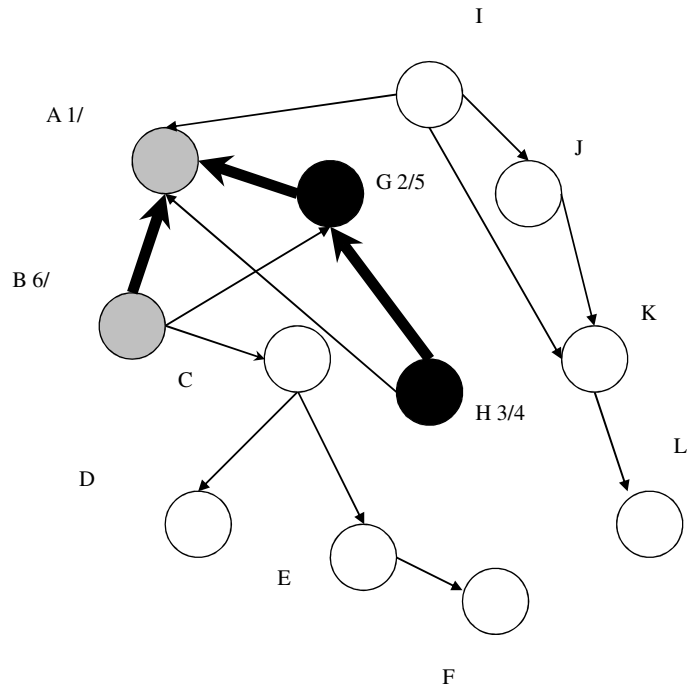


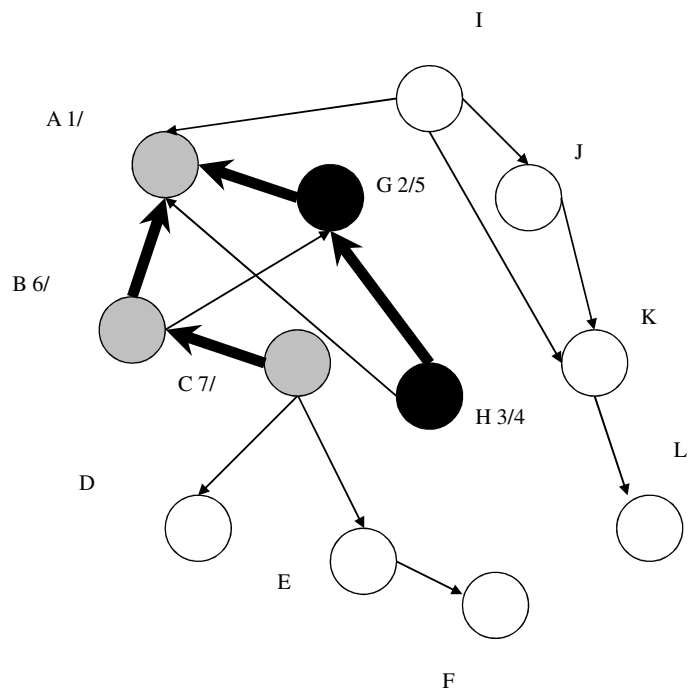
# Zonele parcurgerii grafului

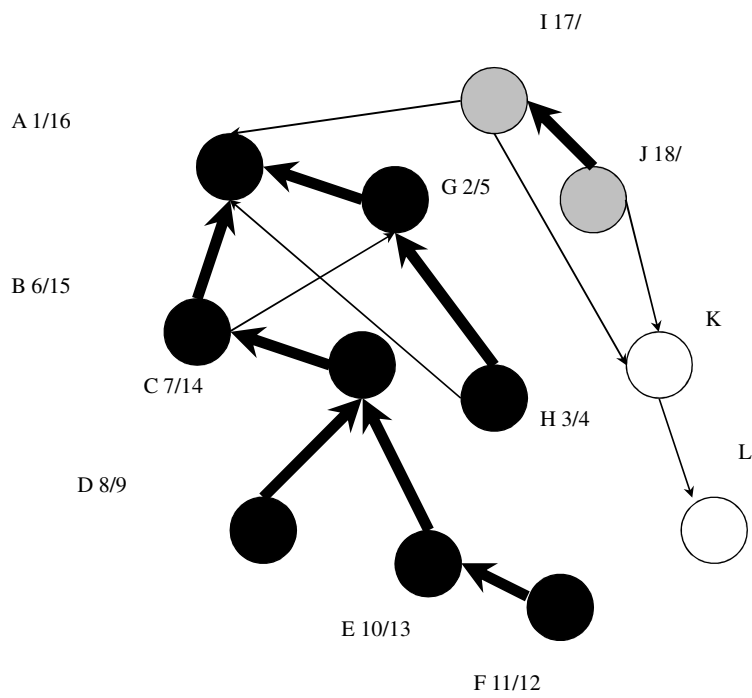


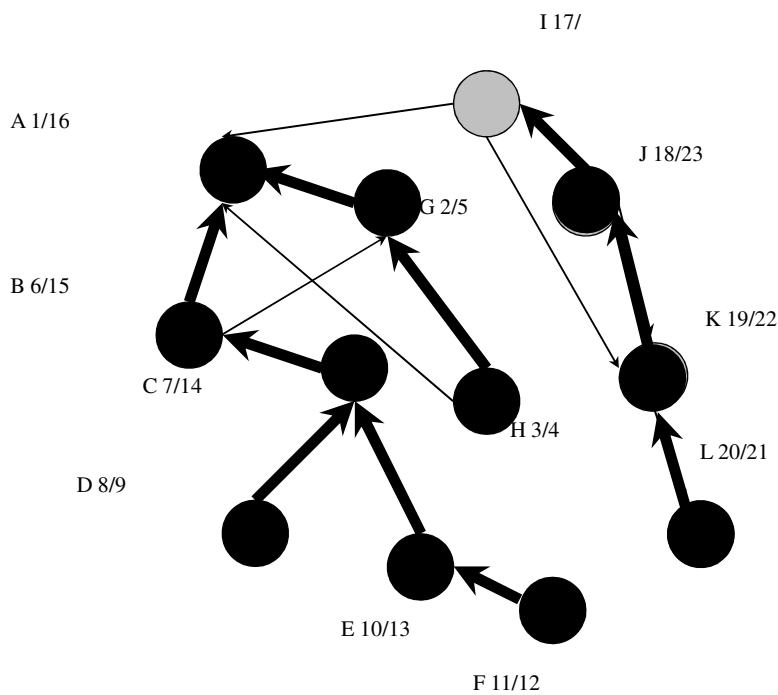




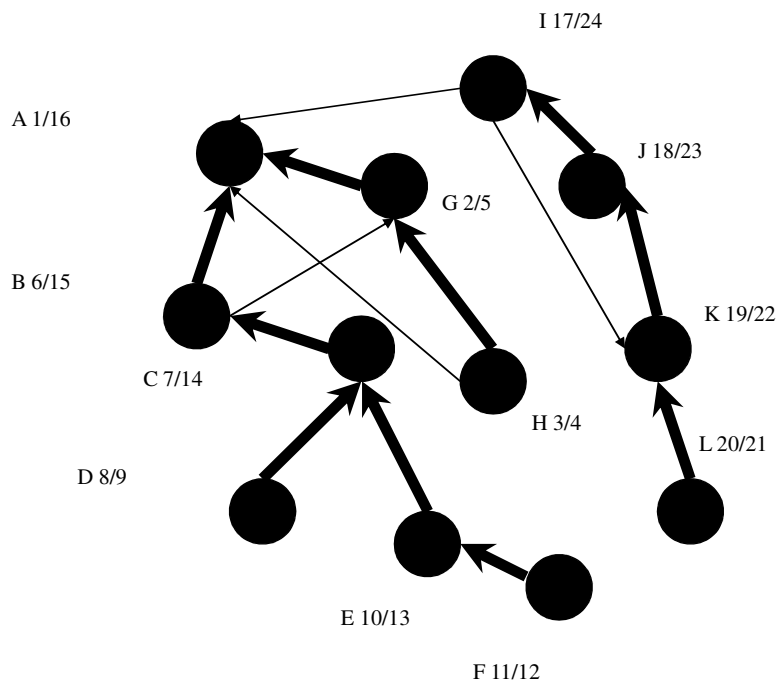








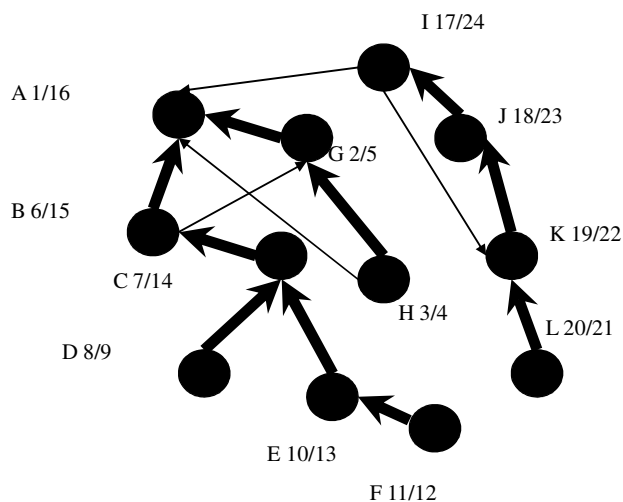




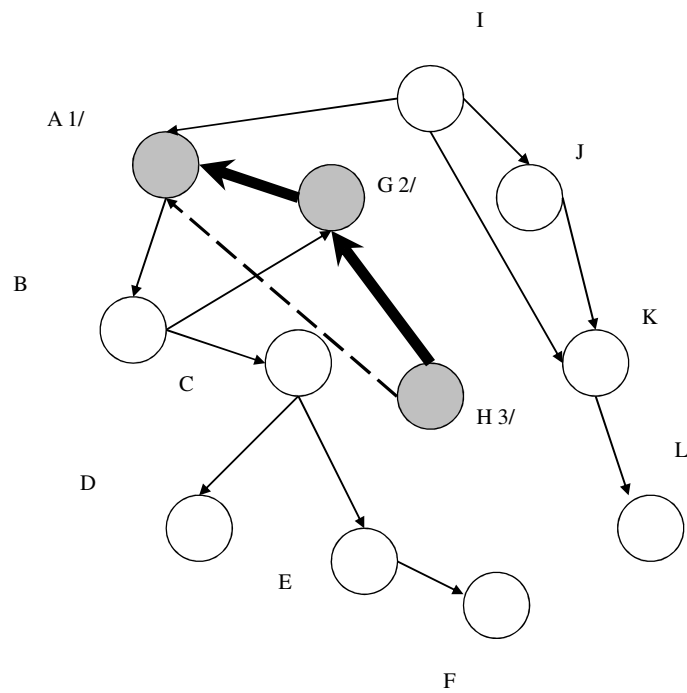
| | | ( a este gri debut/finis= 1 /  
 | | | | | ( g este gri debut/finis= 2 /  
 | | | | | | | ( h este gri debut/finis= 3 /  
 | | | | | | | | ) h este negru debut/finis= 3 / 4  
 | | | | | | ) g este negru debut/finis= 2 / 5  
 | | | | | ( b este gri debut/finis= 6 /  
 | | | | | | | | ( c este gri debut/finis= 7 /  
 | | | | | | | | | | ( d este gri debut/finis= 8 /  
 | | | | | | | | | | | ) d este negru debut/finis= 8 / 9  
 | | | | | | | | | | | | ( e este gri debut/finis= 10 /  
 | | | | | | | | | | | | | | ( f este gri debut/finis= 11 /  
 | | | | | | | | | | | | | | | ) f este negru debut/finis= 11 / 12  
 | | | | | | | | | | | | | | | ) e este negru debut/finis= 10 / 13  
 | | | | | | | | | | | | ) c este negru debut/finis= 7 / 14  
 | | | | | | | | ) b este negru debut/finis= 6 / 15  
 | | | ) a este negru debut/finis= 1 / 16  
 | | | ( i este gri debut/finis= 17 /  
 | | | | | ( j este gri debut/finis= 18 /  
 | | | | | | | | ( k este gri debut/finis= 19 /  
 | | | | | | | | | | ( l este gri debut/finis= 20 /  
 | | | | | | | | | | | ) l este negru debut/finis= 20 / 21  
 | | | | | | | | | | | ) k este negru debut/finis= 19 / 22  
 | | | | | | | | ) j este negru debut/finis= 18 / 23  
 | | | ) i este negru debut/finis= 17 / 24

## Pădurea de arbori de parcurgere în adâncime

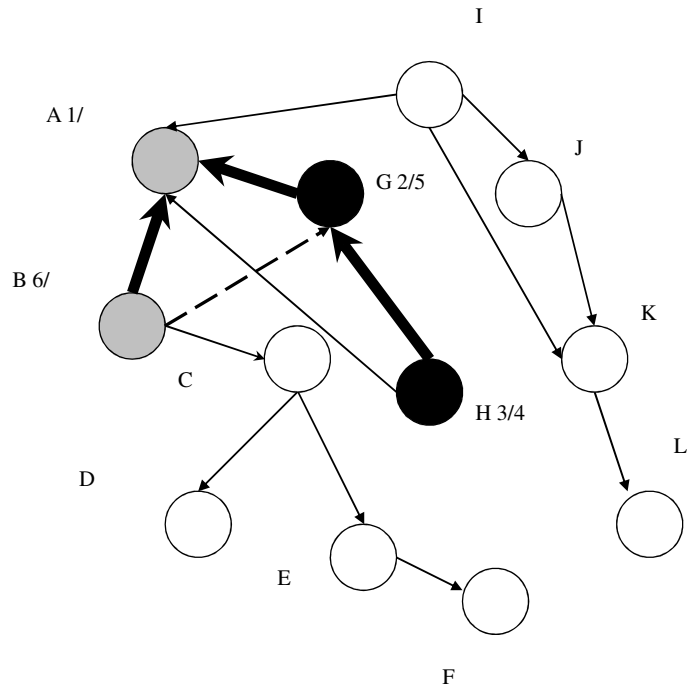
### Arce "în arbore"



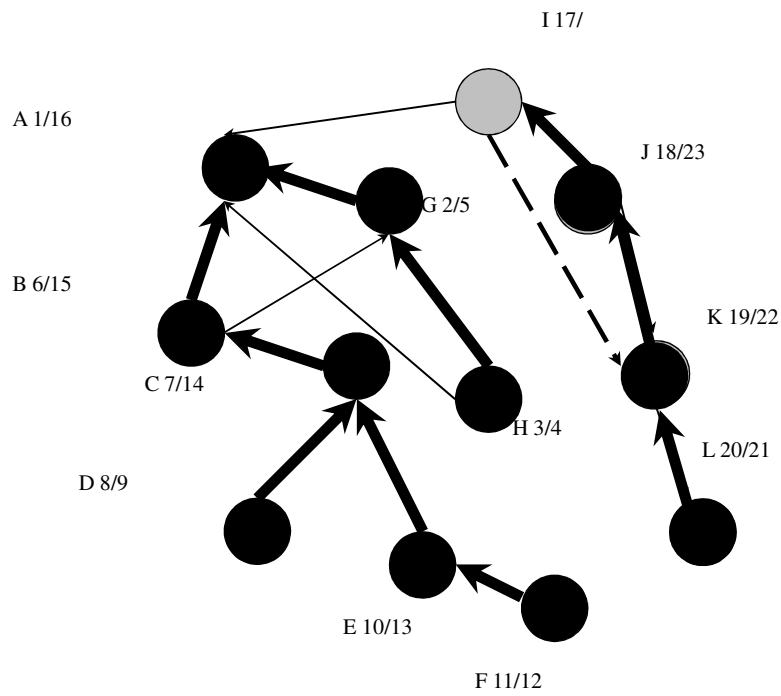
# Arc înapoi



# Arc de traversare



# Arc înainte



## Tipuri de arce

```
if culoare(j)=='alb':
    pune_tata(j,i)
    parc_ad_rec(g,j)
elif culoare(j)=='gri':
    print 'arc inapoi',g[i][0],g[j][0],' - ciclu'
elif culoare(j)=='negru' and g[j][3] < g[i][3]:
    print 'arc de traversare',g[i][0],g[j][0]
elif culoare(j)=='negru' and g[j][3] > g[i][3]:
    print 'arc inainte',g[i][0],g[j][0]
```