Metoda Branch and Bound

Metoda Branch and Bound

Asemănări cu backtracking

- se aplică problemelor care pot fi reprezentate printr-un arbore: la un pas avem de ales între mai multe variante

- Vârfurile arborelui (configuraţiile) corespund stărilor posibile în dezvoltarea soluţiei (soluţii parţiale)

Metoda Branch and Bound

Asemănări cu backtracking

- Pentru o configuraţie se poate **estima** dacă **nu** poate fi completată până la o soluție (**mai bună** decât cea mai bună soluţie determinată până la momentul curent, în cazul problemelor de optim) -> **nu mai este explorată**

  nu se mai parcurge subarborele/ramificaţiile care îl au ca rădăcină ⇒ branch and bound

Metoda Branch and Bound

- **Diferențe față de backtracking**
  - ordinea de parcurgere a arborelui (nu neapărat DF)
  - modul în care sunt eliminaţi subarborii care nu pot conduce la o soluţie
  - arborele poate fi infinit
  - util în probleme de optim

Metoda Branch and Bound

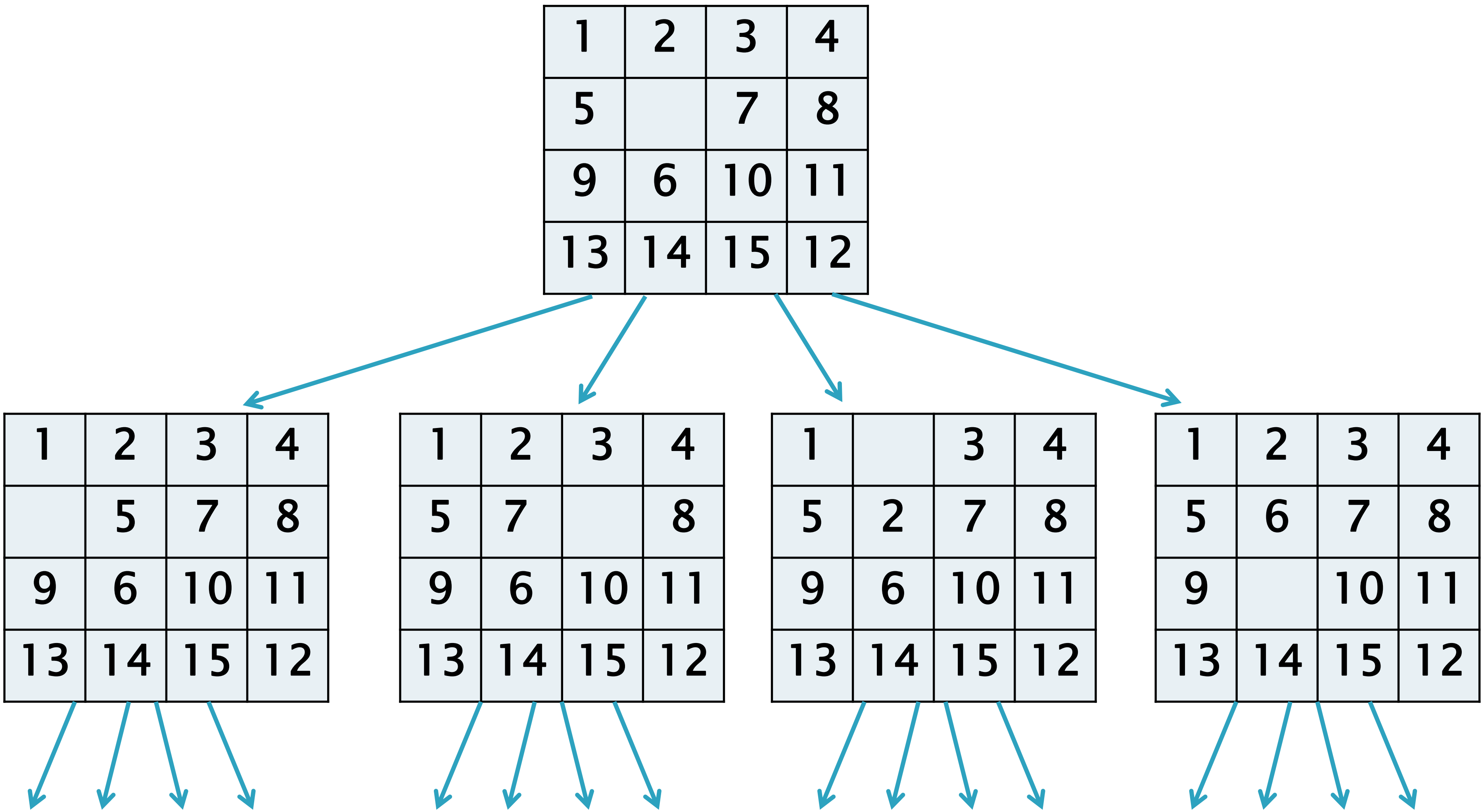
- Două tipuri de probleme la care se poate utiliza:
    - Se caută o anumită soluţie = un anumit ***vârf rezultat*** (final, frunză)
    - Se caută o **<u>soluție optimă</u>** (există mai multe vârfuri finale)
        - **probleme de optim – principalele aplicaţii**

Exemplu – Jocul 15 (Perspico)

1	2	3	4
5		7	8
9	6	10	11
13	14	15	12

$\Rightarrow$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

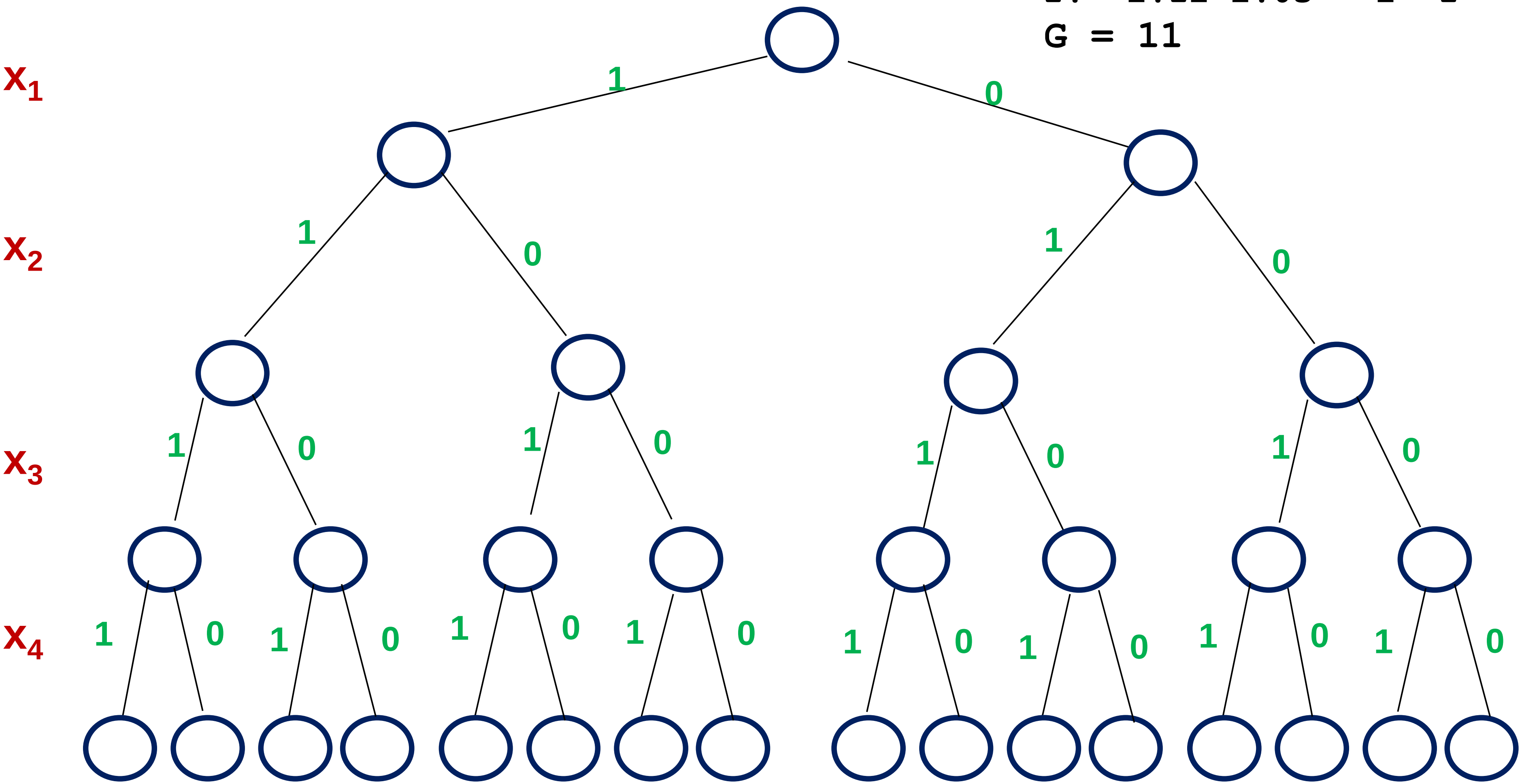

1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
1 3 4
5 2 7 8
9 6 10 11
13 14 15 12
1 2 3 4
5 6 7 8
9 10 11
13 14 15 12

Exemplul 2 – Problema rucsacului

Exemplul 2 – Problema rucsacului

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11



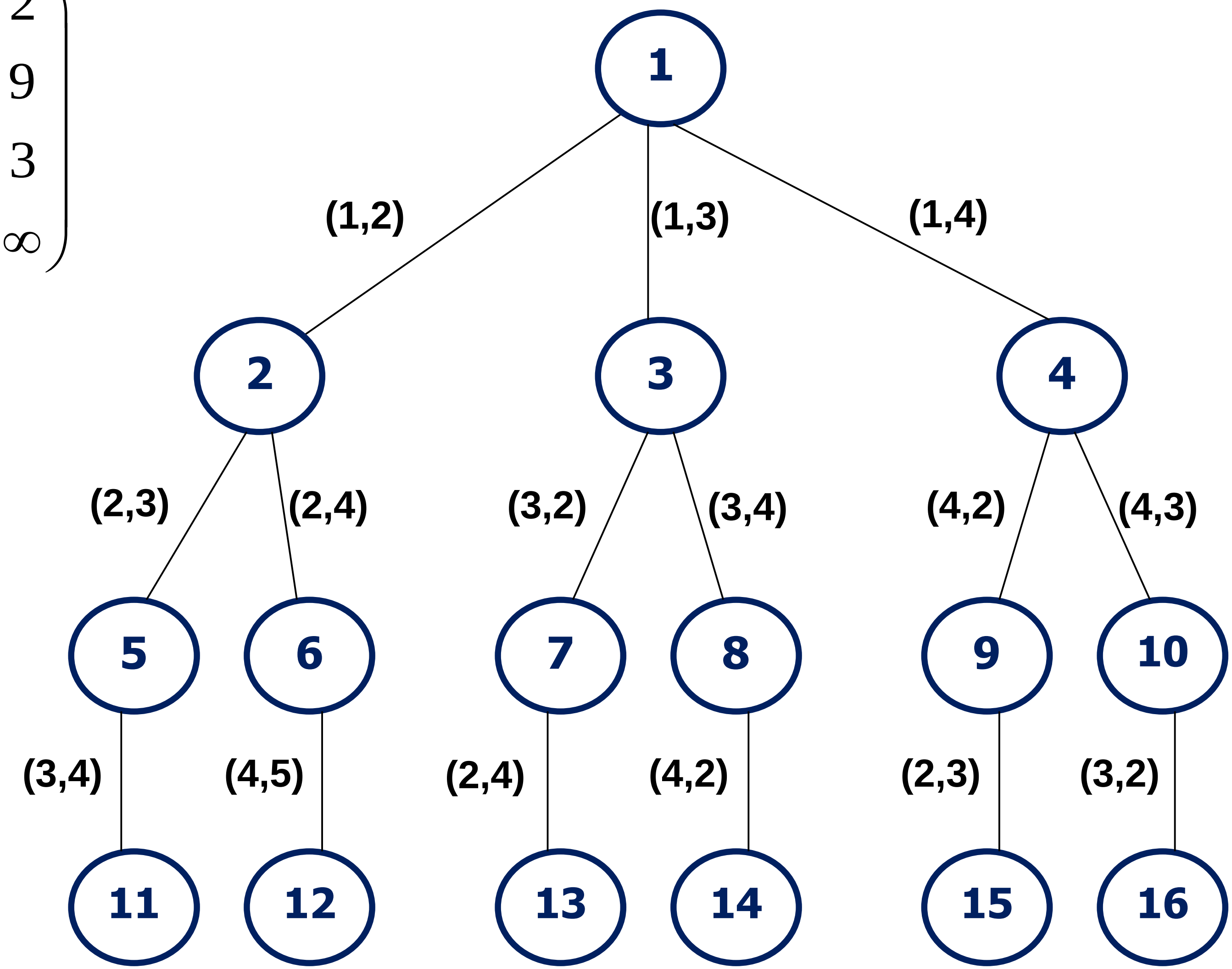
Arborele de stări

Exemplul 3 – circuit hamiltonian minim

Matricea costurilor: $C = \begin{pmatrix} \infty & 3 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$

Exemplul 3 – circuit hamiltonian minim

$$C = \begin{pmatrix} \infty & 3 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$



Metoda Branch and Bound

Schemă (pentru căutarea unui vârf rezultat)

L –lista de vârfuri **active** din arbore (care mai pot fi explorate)

- Se inserează în L vârful inițial
- **Repetă**
  - Se **alege** un vârf din L care devine **curent**
  - Se generează fiii săi şi se adaugă în L

  până când vârful curent este final

Metoda Branch and Bound

- Cum se alege vârful curent (cum parcurgem arborele de stări parțiale)?

Metoda Branch and Bound

Cum se alege vârful curent

- **DF**

  - soluţia căutată poate fi de exemplu un fiu al rădăcinii diferit de primul fiu

Metoda Branch and Bound

Cum se alege vârful curent

- **DF**

  – soluţia căutată poate fi de exemplu un fiu al rădăcinii diferit de primul fiu

- **BF**

  – conduce totdeauna la soluţie, dar poate fi ineficientă dacă vârfurile au mulţi fii

Metoda Branch and Bound

- **Compromis**:

  Vârf x – asociat un **cost** pozitiv c(x)

  = **măsură a costului soluției care trece prin starea x (conține x) /**

  măsură a distanței de la rădăcină la o soluție trecând prin starea x

Metoda Branch and Bound

- **Compromis**:

  Vârf x – asociat un **cost** pozitiv c(x)

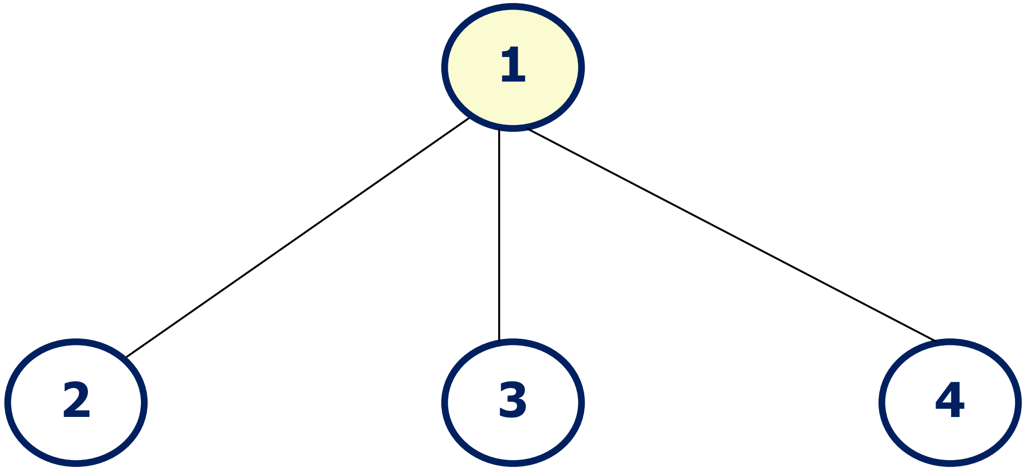
  = măsură a costului soluției care trece prin starea x (conține x)

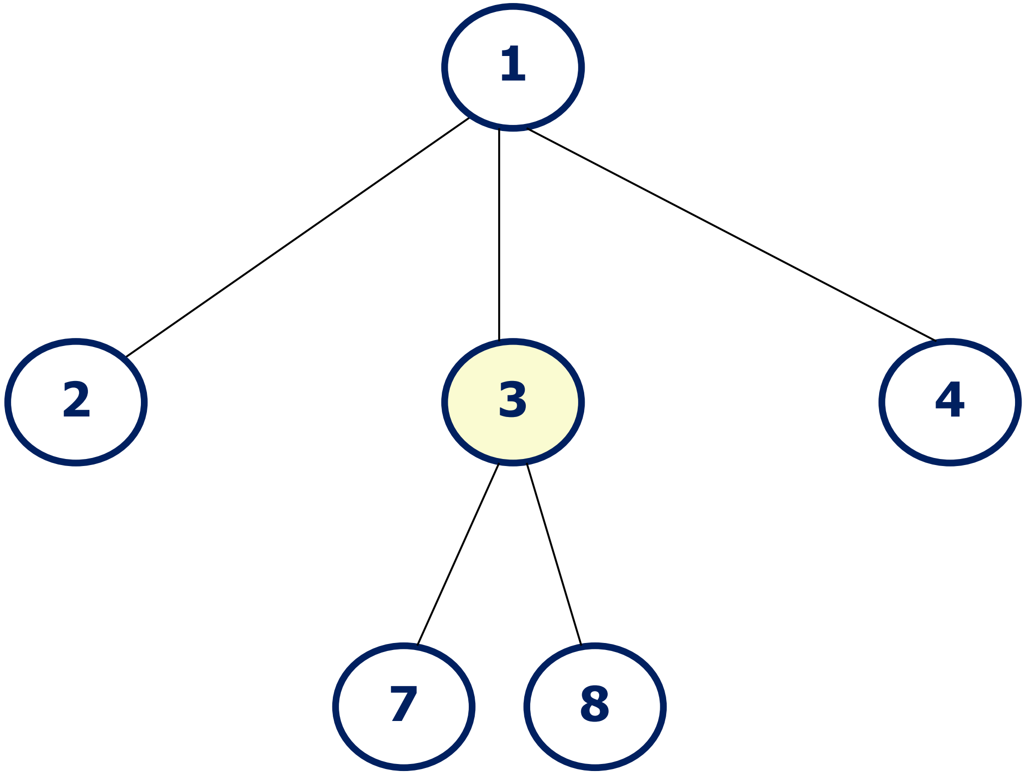
- Este ales vârful de cost minim (! care **nu este neapărat fiu al vârfului curent anterior**)

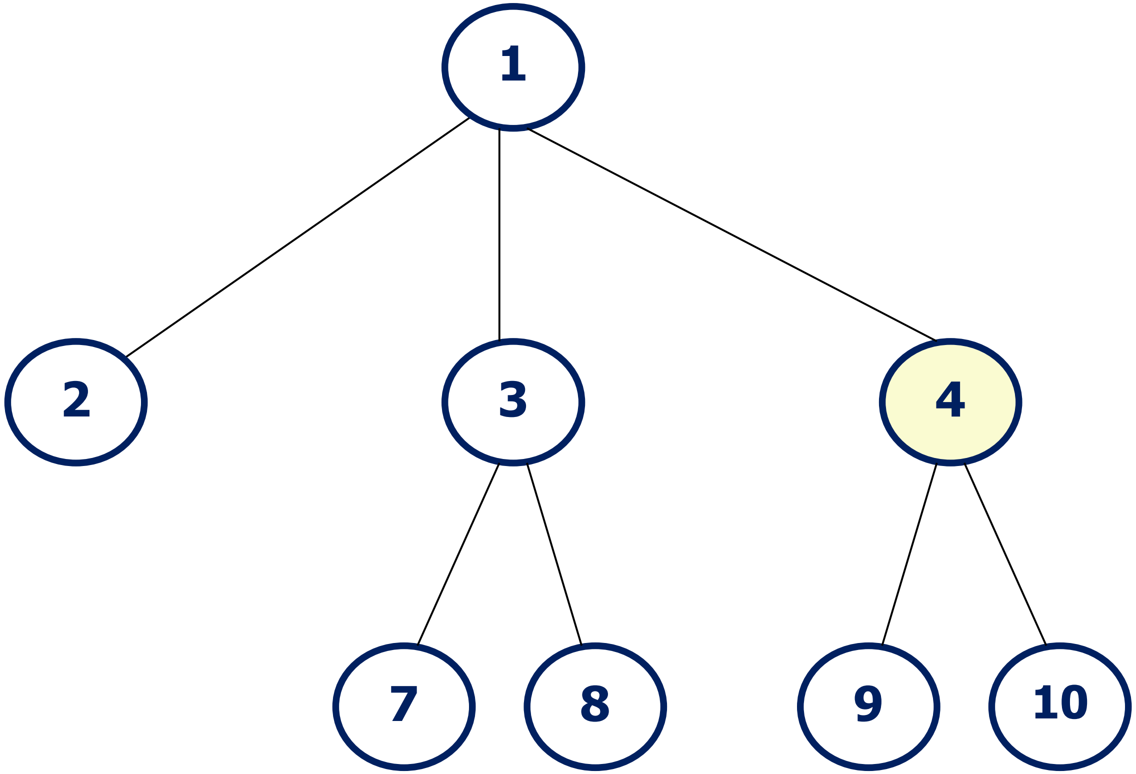
- c(varf) $\leq$ c(fiu)

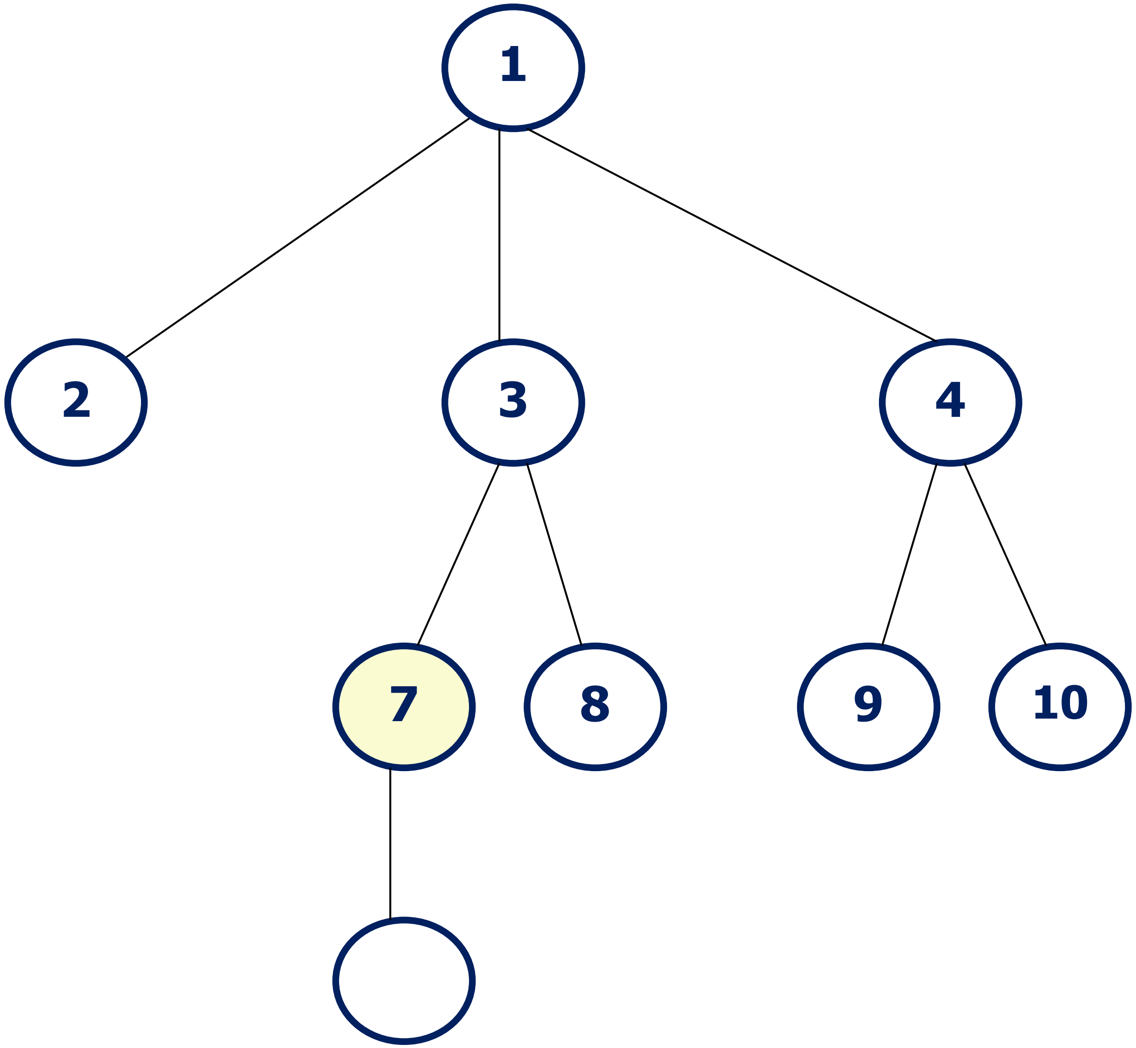
- **c(frunza) = cost real**

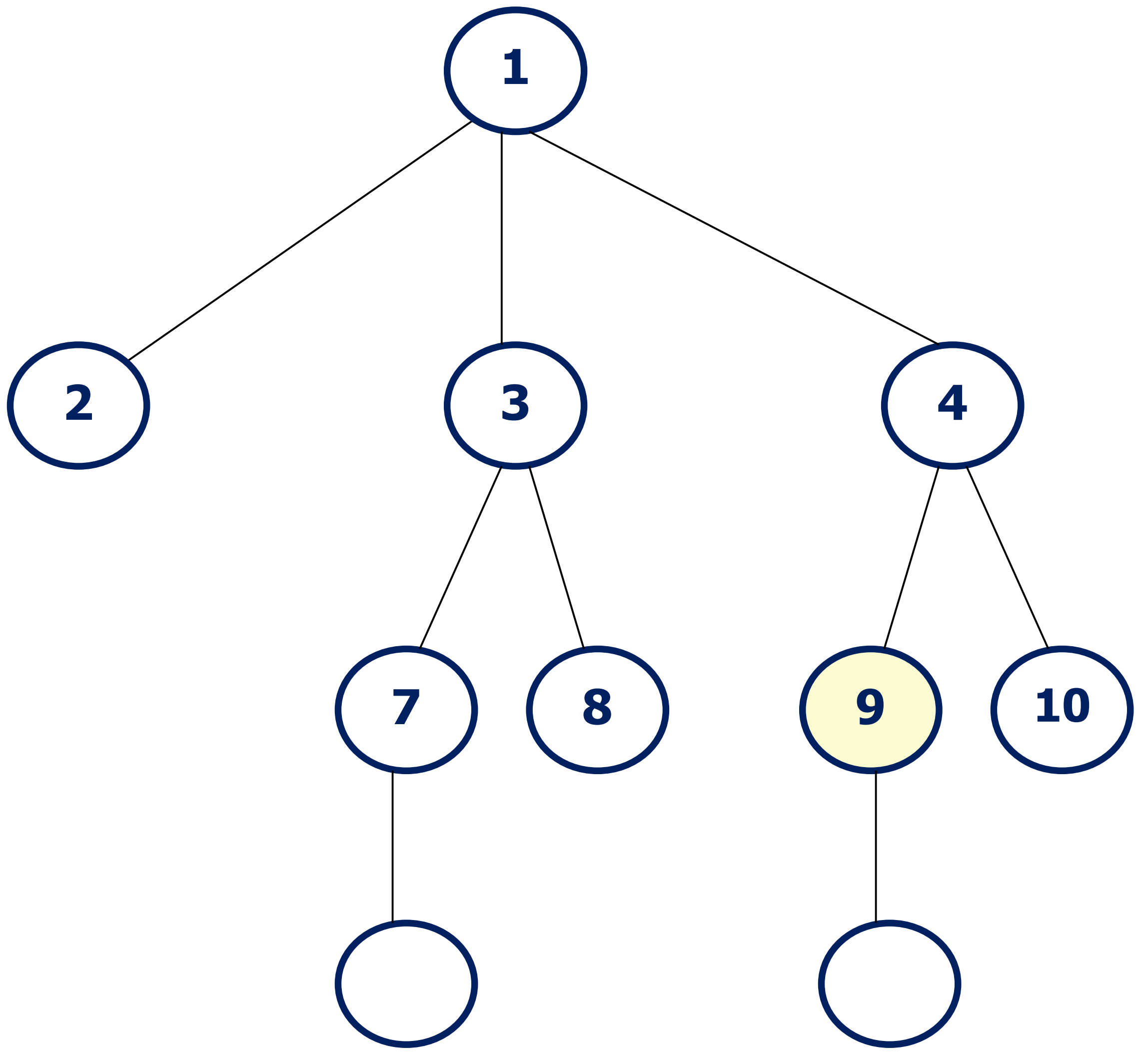
L va fi în general un min-ansamblu


1
2
3
4


1
2
3
4
7
8


1
2
3
4
7
8
9
10


1
2
3
4
7
8
9
10


1
2
3
4
7
8
9
10

Metoda Branch and Bound

- **Funcție de cost ideală**

c(x) =

- **cost**(x), dacă x este vârf **rezultat = costul soluției x =**

  costul pentru a ajunge de la rad la x (de exp nivelul vârfului)

- +∞**,** dacă x este vârf **final**, diferit de vârf rezultat

- **min** {c(y) | y fiu al lui x }**,** dacă x nu este vârf final

  - **Trebuie parcurs arborele în întregime pentru a o calcula**

Metoda Branch and Bound

- **Aproximație f a lui c**
  -
  -

Metoda Branch and Bound

- **Aproximație f a lui c**
  - f(x) să poată fi calculată doar pe baza informaţilor din drumul de la rădăcină la x
  - Pentru corectitudine (!probleme de minimizare)

  – trebuie ca **$f \leq c$** (aproximare optimistă)

  – f(frunza) = cost real

Metoda Branch and Bound

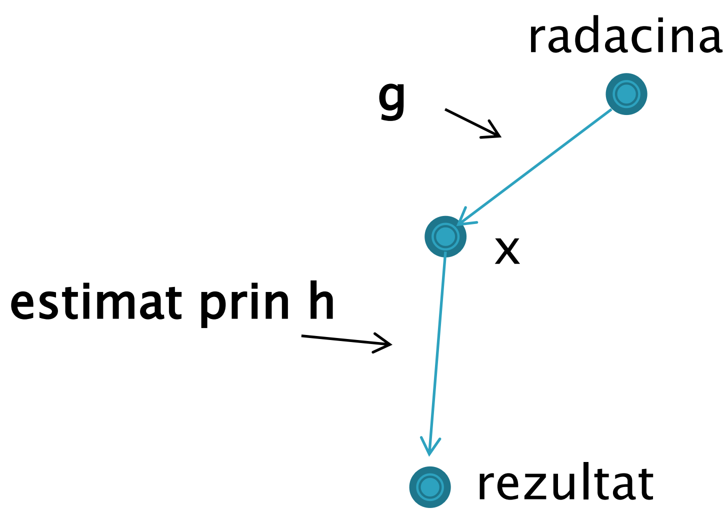
- **Cum definim f?**

Metoda Branch and Bound

- **Cum definim f? O posibililitate ar fi:**

$$\mathbf{f(x) = g(x)+h(x)}$$

- **g(x)** = distanţa de la rădăcină la nodul x (costul soluției parțiale)
- **h(x)** = estimarea distanţei de la x la vârful rezultat (**"subestimare"**)



Metoda Branch and Bound

- Cum definim f?

- Estimări cu algoritmi de tip greedy

  - Exemplu:

valoarea maximă problema discretă a rucsacului $\leq$

valoarea maximă problema fracționară a rucsacului corespunzătoare (cu aceleași date de intrare)

Metoda Branch and Bound

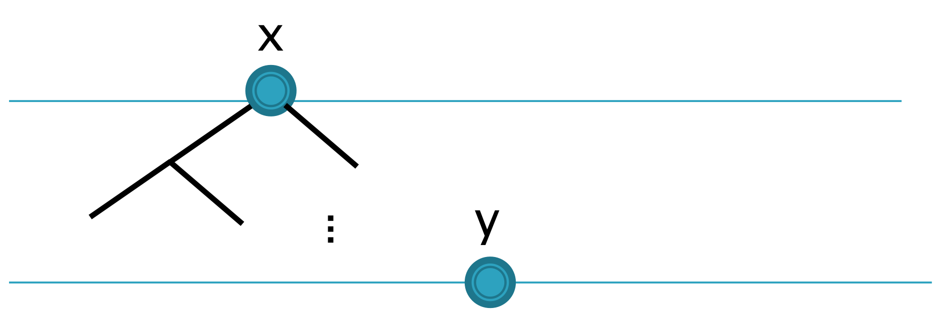
- **O modalitate de a asigura compromisul între DF şi BF este de a alege f astfel încât**

  există un k natural a,î. pentru orice vârf x situat pe un nivel nx şi orice vârf y situat pe un nivel

  $$ny \geq nx+k,$$

  să avem

  $$f(x) < f(y)$$



Exemplu – Jocul 15 (Perspico)

1	2	3	4
5		7	8
9	6	10	11
13	14	15	12

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Se cere: o succesiune de mutări pentru a ajunge de la prima configurație la cea de a doua

Exemplu – Jocul 15 (Perspico)

- **Condiție de existență**

  – căsuța liberă 16, pe poziția (l, c)

  – pentru plăcuța etichetată cu i calculăm

n(i) = numărul locaşurilor care îi urmează şi care conţin o plăcuţă cu etichetă mai mică decât i

Există soluție ⇔

n(1)+n(2)+...+n(16)+(l+c) %2 este par.

Exemplu – Jocul 15 (Perspico)

- h(t) = ?

1	2	3	4
5	6	7	8
9	14	10	11
13		15	12

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

în câți pași minim?

Exemplu – Jocul 15 (Perspico)

- Putem considera euristici precum:
  - $h_1(t)$ = numărul de plăcuţe care nu sunt la locul lor
    distanţa Hamming

  - $h_2(t)$ = **distanţa Manhattan** = suma pentru fiecare plăcuţă a **numărului de mutări** pentru a o putea aduce în poziţia finală
    - dacă o cifră este pe poziţia (i,j) şi trebuie adusă pe poziţia (r,s) distanţa este

$$|r-i|+|s-j|$$

h = câte plăcuţe nu sunt la locul lor

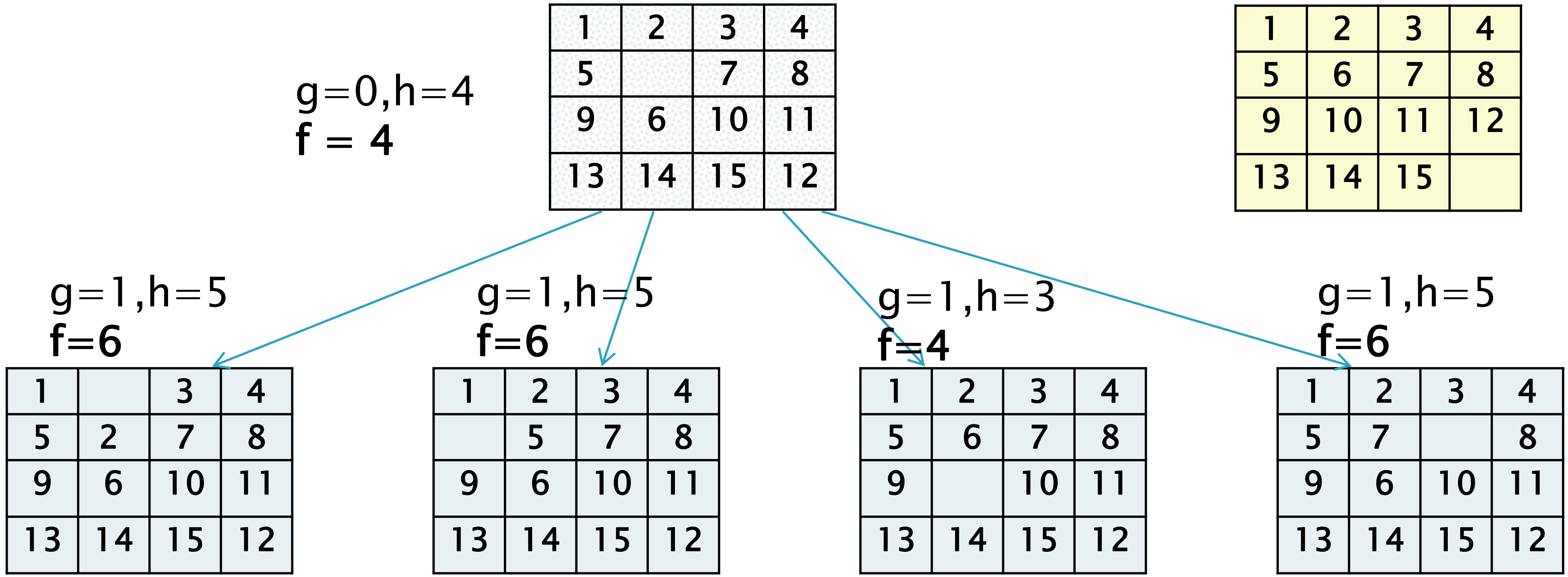
1	2	3	4
5		7	8
9	6	10	11
13	14	15	12

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

g = 0
h = 4

1	2	3	4
5		7	8
9	6	10	11
13	14	15	12

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	


g=0,h=4
f = 4
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15
g=1,h=5
f=6
1 3 4
5 2 7 8
9 6 10 11
13 14 15 12
g=1,h=5
f=6
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
g=1,h=3
f=4
1 2 3 4
5 6 7 8
9 10 11
13 14 15 12
g=1,h=5
f=6
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12

g=0,h=4

f = 4

1	2	3	4
5		7	8
9	6	10	11
13	14	15	12

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

g=1,h=5

f=6

1		3	4
5	2	7	8
9	6	10	11
13	14	15	12

g=1,h=5

f=6

1	2	3	4
	5	7	8
9	6	10	11
13	14	15	12

g=1,h=3

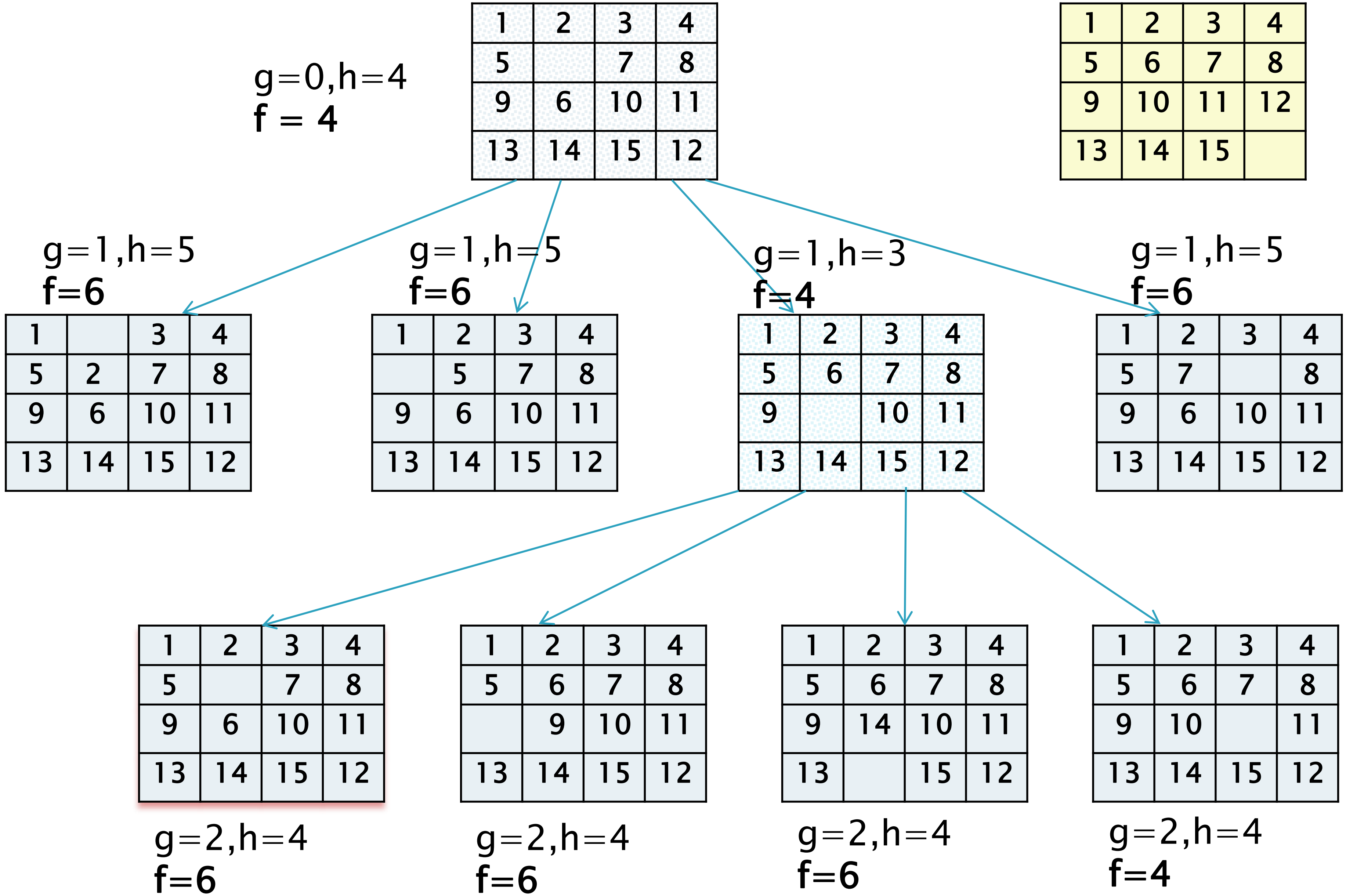
f=4

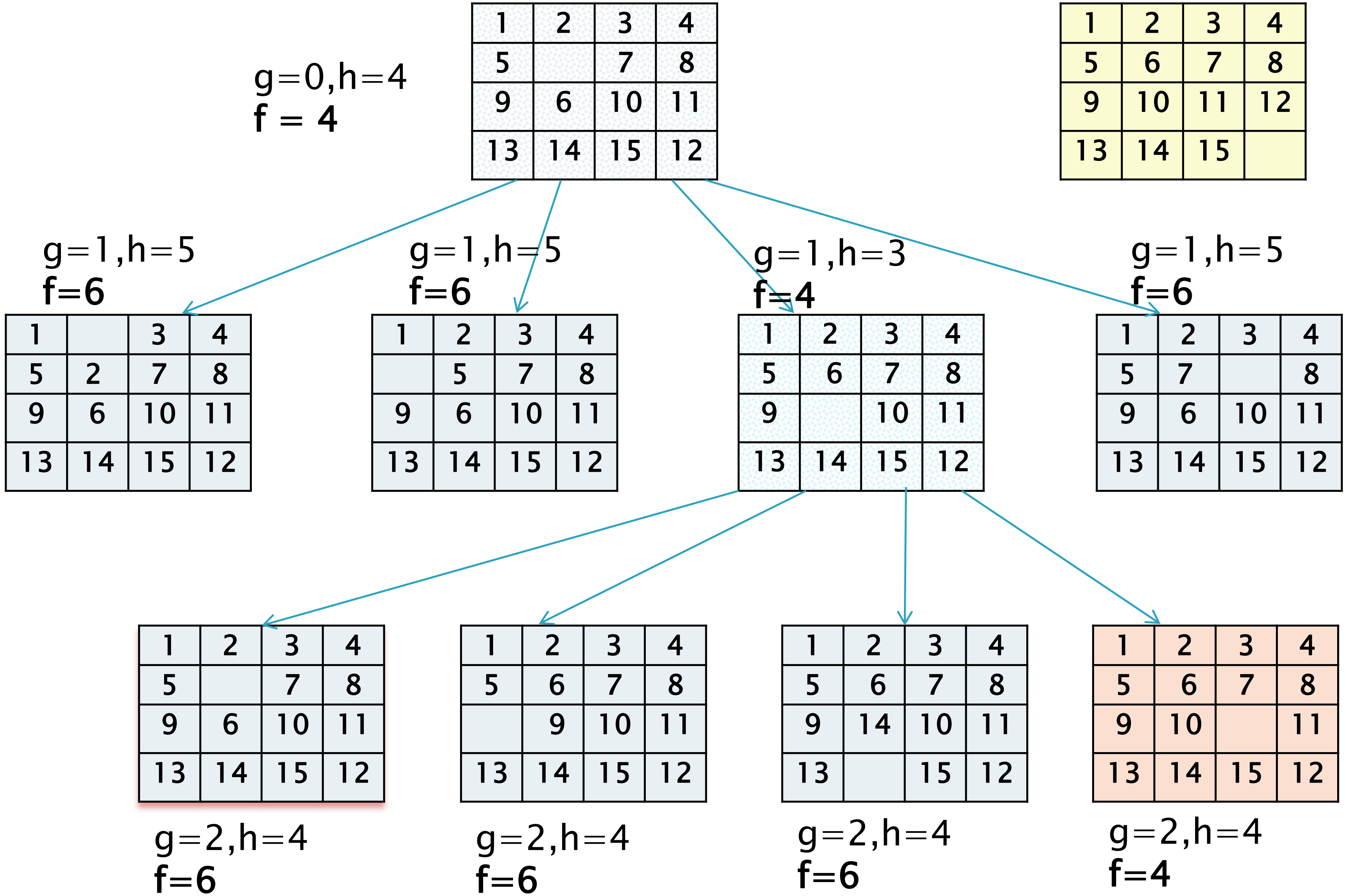
1	2	3	4
5	6	7	8
9		10	11
13	14	15	12

g=1,h=5

f=6

1	2	3	4
5	7		8
9	6	10	11
13	14	15	12


g=0,h=4
f = 4
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15
g=1,h=5
f=6
1 3 4
5 2 7 8
9 6 10 11
13 14 15 12
g=1,h=5
f=6
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
g=1,h=3
f=4
1 2 3 4
5 6 7 8
9 10 11
13 14 15 12
g=1,h=5
f=6
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
1 2 3 4
5 7 8
9 6 10 11
13 14 15 12
g=2,h=4
f=6
1 2 3 4
5 6 7 8
9 10 11
13 14 15 12
g=2,h=4
f=6
1 2 3 4
5 6 7 8
9 14 10 11
13 15 12
g=2,h=4
f=6
1 2 3 4
5 6 7 8
9 10 11
13 14 15 12
g=2,h=4
f=4


g=0,h=4
f = 4
g=1,h=5
f=6
g=1,h=5
f=6
g=1,h=3
f=4
g=1,h=5
f=6
g=2,h=4
f=6
g=2,h=4
f=6
g=2,h=4
f=6
g=2,h=4
f=4

Metoda Branch and Bound

- **Pentru probleme de optim**
  - **Vom considera probleme de minimizare**
  - Maximizare – similar

    (maximizare obiectiv g -> minimizare obiectiv -g)

Pseudocod

Metoda Branch and Bound

- **Pentru probleme de optim (minim)**
  - Considerăm `lim` – aproximaţie prin adaos a minimului căutat
    - **se actualizează pe parcursul algoritmului = costul celei mai bune soluţii găsite până la pasul curent**

Metoda Branch and Bound

- **Pentru probleme de optim (minim)**
  - Considerăm **`lim`** – aproximaţie prin adaos a minimului căutat
    - **se actualizează pe parcursul algoritmului = costul celei mai bune soluţii găsite până la pasul curent**
  - Configuraţiile cu costul estimat mai mare decât **`lim`** nu mai trebuie considerate:

    cost real ≥ cost estimat > lim

Metoda Branch and Bound

- **Pentru probleme de optim (minim)**
  - Notăm rad – vârful corespunzător configurației inițiale

```
i ← rad; L ⇐ {i}; min ← lim; i_final ← ∞
calculăm f(rad); tata(i) ← 0
```

```
i ← rad; L ⇐ {i}; min ← lim; i_final ← ∞
calculăm f(rad); tata(i) ← 0
while L ≠ ∅
    i ⇐ L {scos vârful i din L, de obicei cu f(i) minim}
    for toţi j fii ai lui i
```

```
i ← rad; L ⇐ {i}; min ← lim; i_final ← ∞
calculăm f(rad); tata(i) ← 0
while L ≠ ∅
    i ⇐ L
    for toţi j fii ai lui i
        calculăm f(j); calcule locale asupra lui j;
        tata(j) ← i
```

```
i ← rad; L ⇐ {i}; min ← lim; i_final ← ∞
calculăm f(rad); tata(i) ← 0
while L ≠ ∅
    i ⇐ L
    for toţi j fii ai lui i
        calculăm f(j); calcule locale asupra lui j;
        tata(j) ← i
        if j este vârf final
            if f(j)<min
                min ← f(j); i_final ← j
                elimină din L vârfurile k cu f(k)≥ min
```

```
i ← rad; L ⇐ {i}; min ← lim; i_final ← ∞
calculăm f(rad); tata(i) ← 0
while L ≠ ∅
    i ⇐ L
    for toţi j fii ai lui i
        calculăm f(j); calcule locale asupra lui j;
        tata(j) ← i
        if j este vârf final
                if f(j)<min
                        min ← f(j); i_final ← j
                        elimină din L vârfurile k cu f(k)≥ min
        else
```

```
i ← rad; L ⇐ {i}; min ← lim; i_final ← ∞
calculăm f(rad); tata(i) ← 0
while L ≠ ∅
    i ⇐ L
    for toţi j fii ai lui i
        calculăm f(j); calcule locale asupra lui j;
        tata(j) ← i
        if j este vârf final
                if f(j)<min
                        min ← f(j); i_final ← j
                        elimină din L vârfurile k cu f(k)≥ min
        else
                if f(j)<min
                        j ⇒ L
```

```
i ← rad; L ⇐ {i}; min ← lim; i_final ← ∞
calculăm f(rad); tata(i) ← 0
while L ≠ ∅
    i ⇐ L
    for toţi j fii ai lui i
        calculăm f(j); calcule locale asupra lui j;
        tata(j) ← i
        if j este vârf final
             if f(j)<min
                  min ← f(j); i_final ← j
                  elimină din L vârfurile k cu f(k)≥ min
        else
             if f(j)<min
                  j ⇒ L

if i_final = ∞
      write('Nu există soluţie')
else writeln(min); i ← i_final
     while i ≠ 0
         write(i); i ← tata(i)
```

Metoda Branch and Bound – pseusocod

- **De obicei L – min-ansamblu**

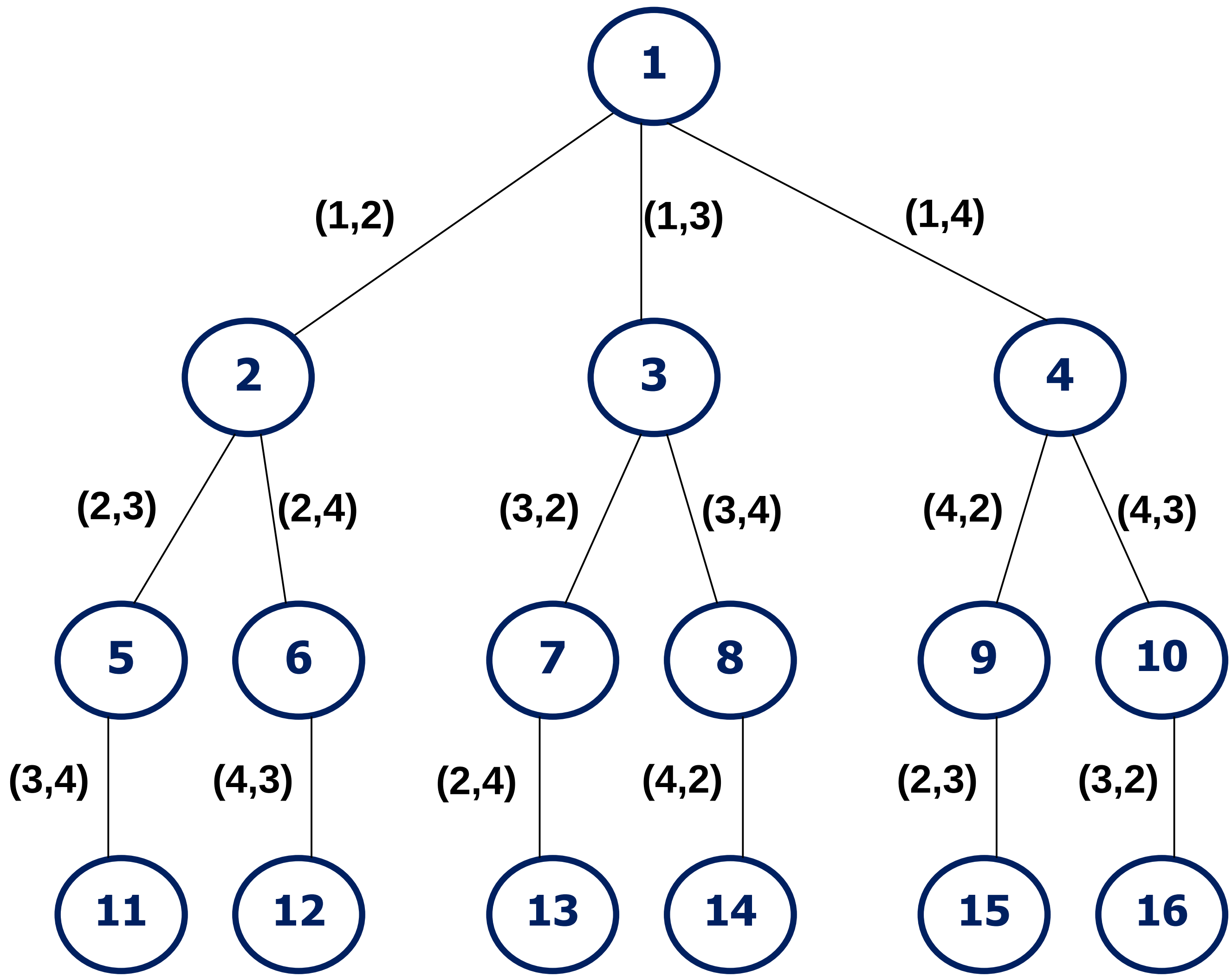
  ($\Rightarrow$ nu mai este neapărată nevoie de eliminare)

Exemplu – circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Exemplu – circuit hamiltonian minim TSP

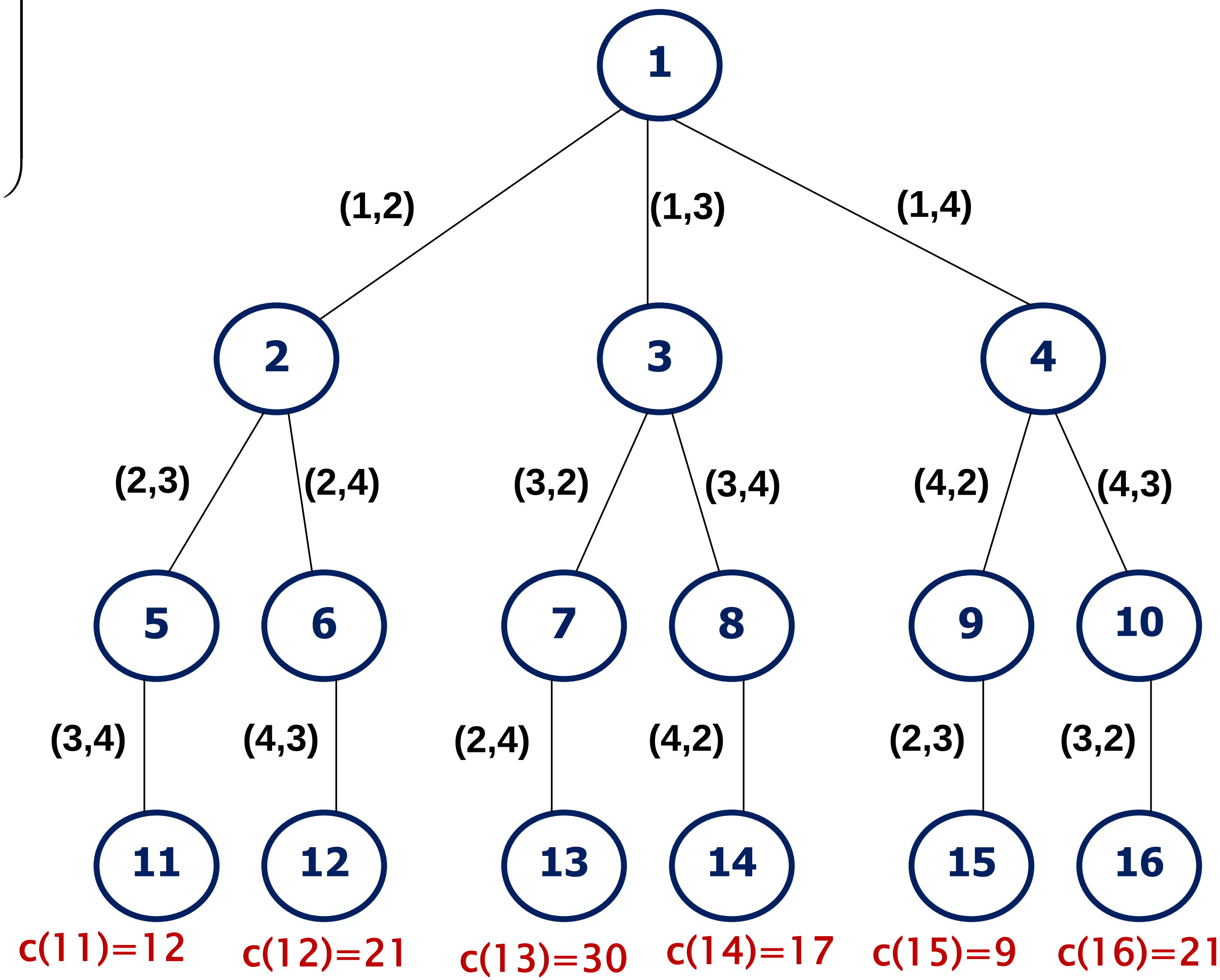
$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$



Exemplu – circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

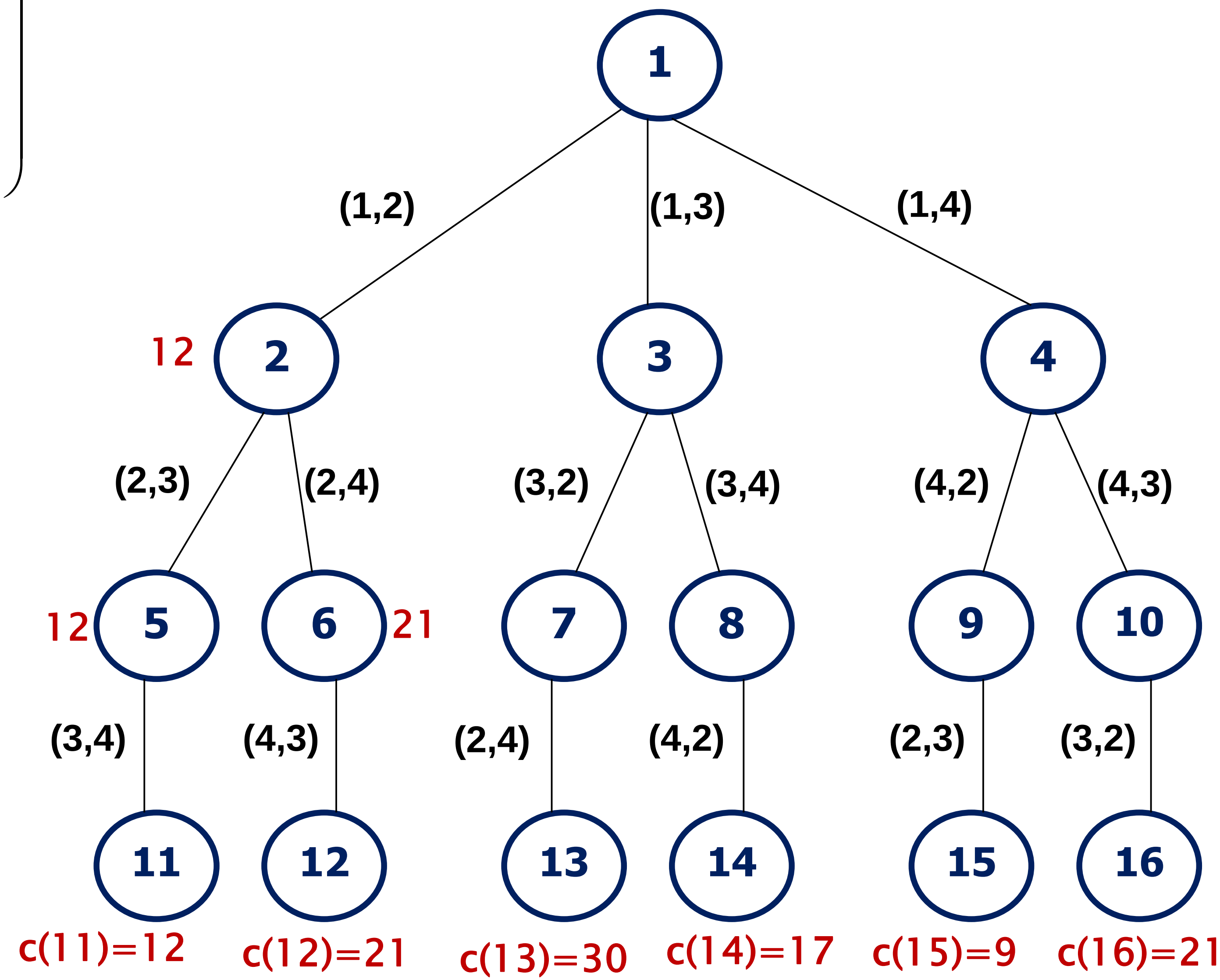
Funcția de cost ideală



Exemplu – circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

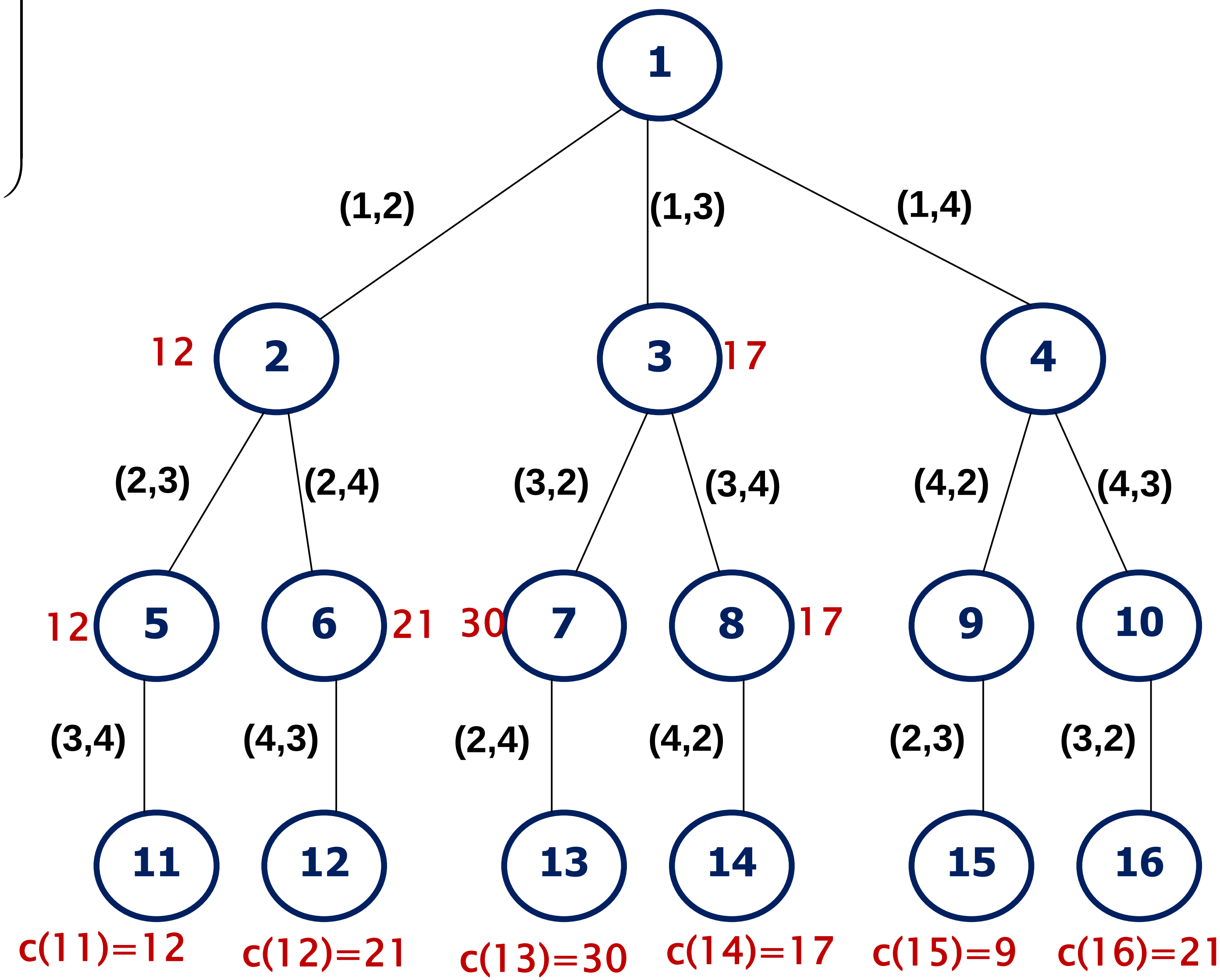
Funcția de cost ideală



Exemplu – circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

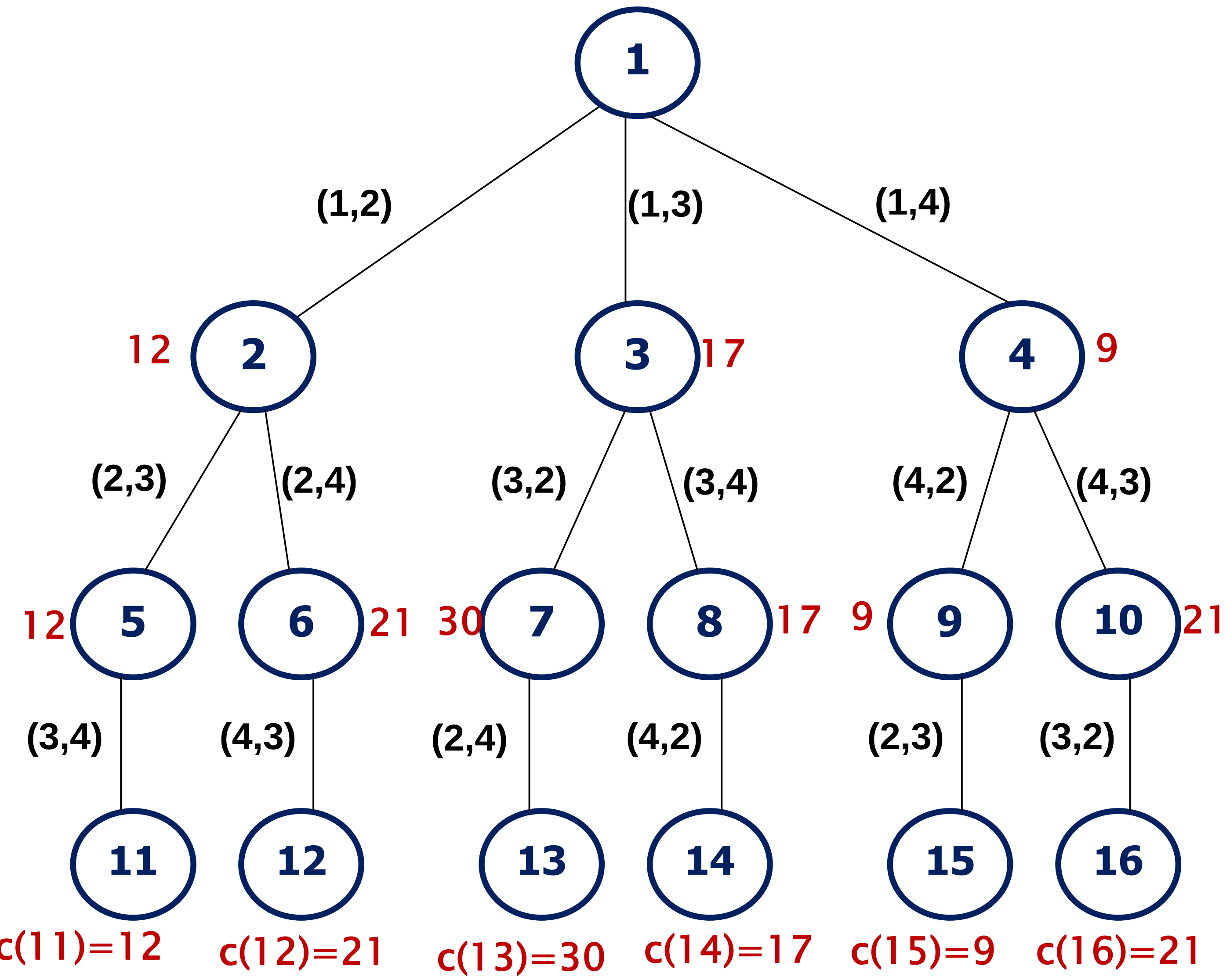
Funcția de cost ideală



Exemplu – circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

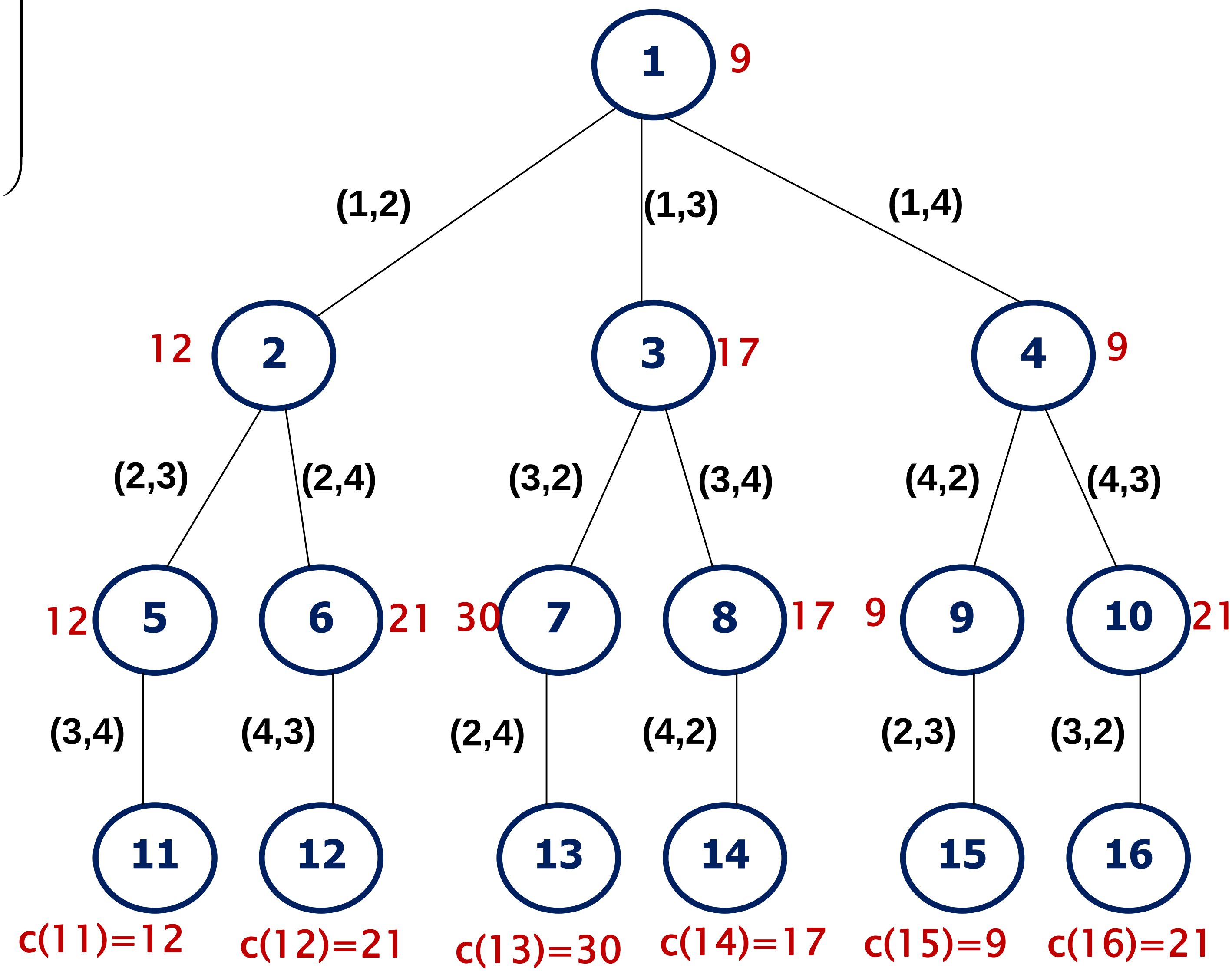
Funcția de cost ideală



Exemplu – circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Funcția de cost ideală



Exemplu – circuit hamiltonian minim TSP

Pentru un vârf x din arbore valoarea c(x) dată de funcţia de cost **ideală** este:

- lungimea circuitului corespunzător lui x
  dacă x este frunză

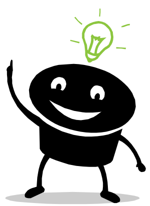
- min {c(y) | y fiu al lui x } altfel.

Exemplu – circuit hamiltonian minim TSP

- Cum determinăm limita inferioară f(x) pentru circuitului minim corespunzător vârfului x din arborele Branch and Bound

Exemplu – circuit hamiltonian minim TSP

Cum determinăm limita inferioară f(x) pentru circuitului minim corespunzător vârfului x din arborele Branch and Bound



Idee simplă – lungimea circuitului minim care corespunde lui x este $\leq$ lungimea drumului de la rădăcină la x (arcelor alese pentru a ajunge la x)

`f(x) = g(x) + h(x)`

- **`g(x)`** = distanţa de la rădăcină la nodul x
- **`h(x)`** = 0 -> este o **"subestimare"**

Exemplu – circuit hamiltonian minim TSP

Cum determinăm limita inferioară f(x) pentru circuitului minim corespunzător vârfului x din arborele Branch and Bound

- **`h(x)`** = 0 -> este o **"subestimare“**
- **Problemă – cu cât estimarea este mai puţin precisă (mai depărtată de valoarea reală) sunt excluse mai puţine vârfuri din parcurgere ⇒ algoritm mai lent**

L = {1}

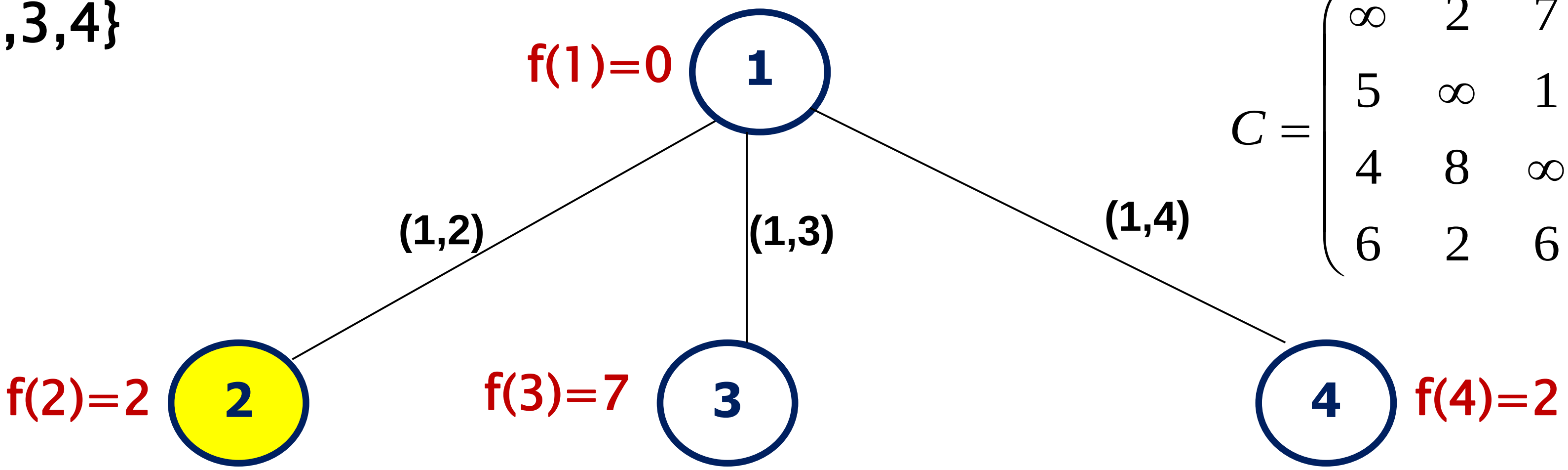
f(1)=0 1

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

lim = ∞

min= ∞

L = {2,3,4}



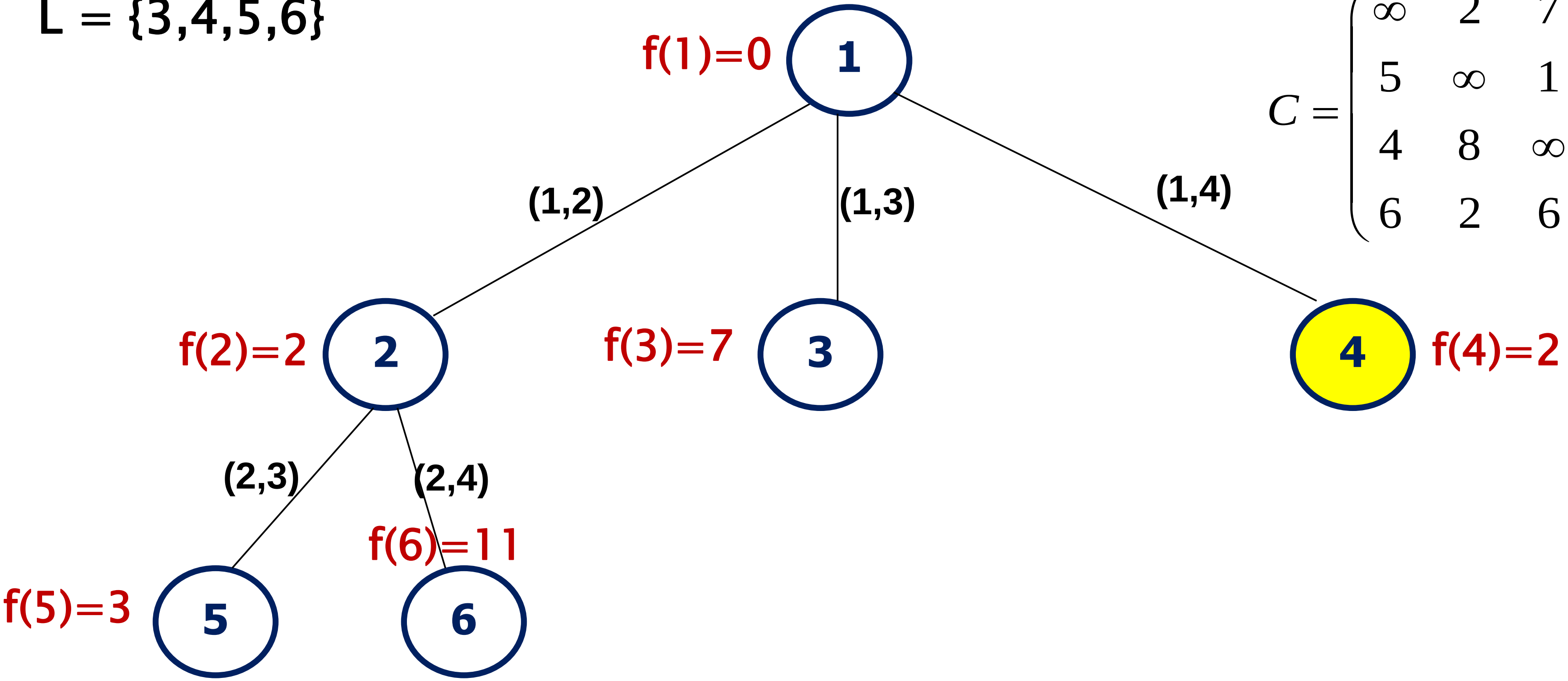
$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Extragem din L vârful cu f minim şi îi generăm fii

min= ∞

L = {3,4,5,6}

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

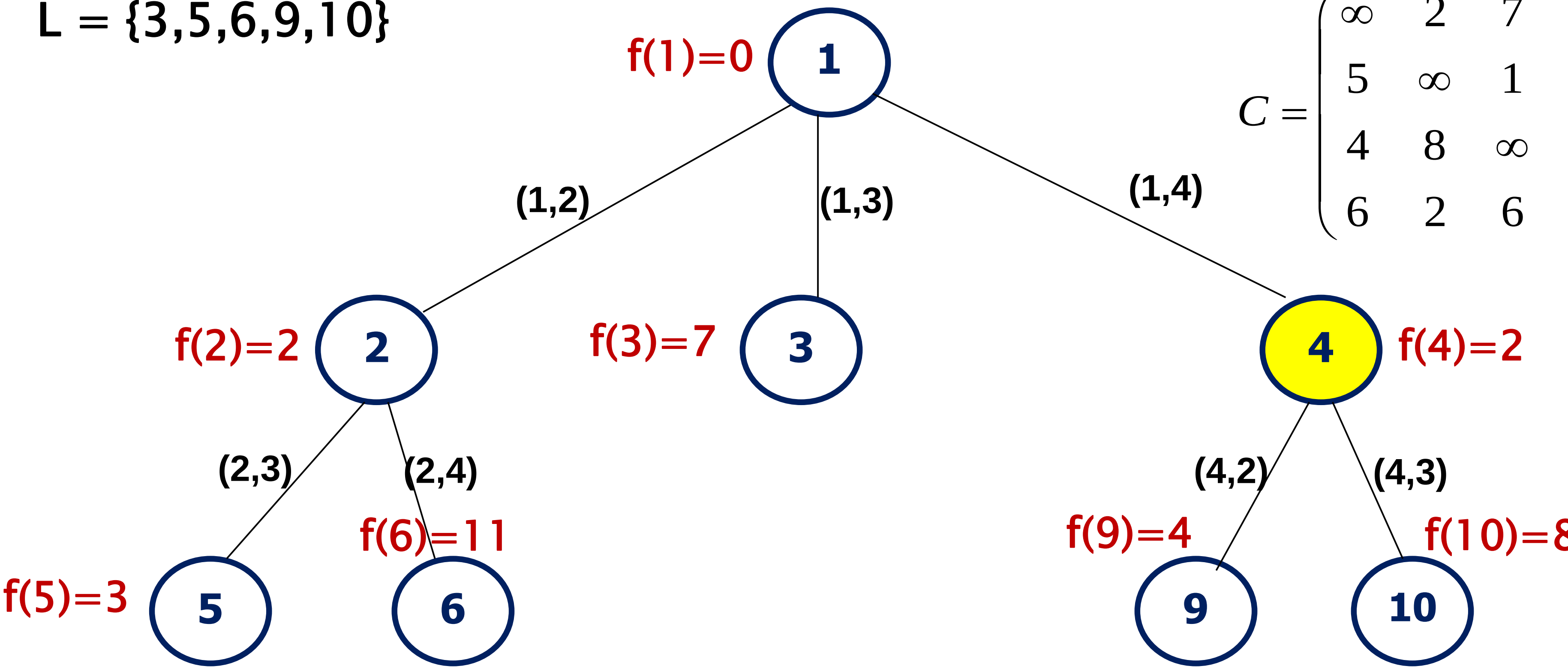


Extragem din L vârful cu f minim şi îi generăm fii

min= ∞

$L = \{3,5,6,9,10\}$

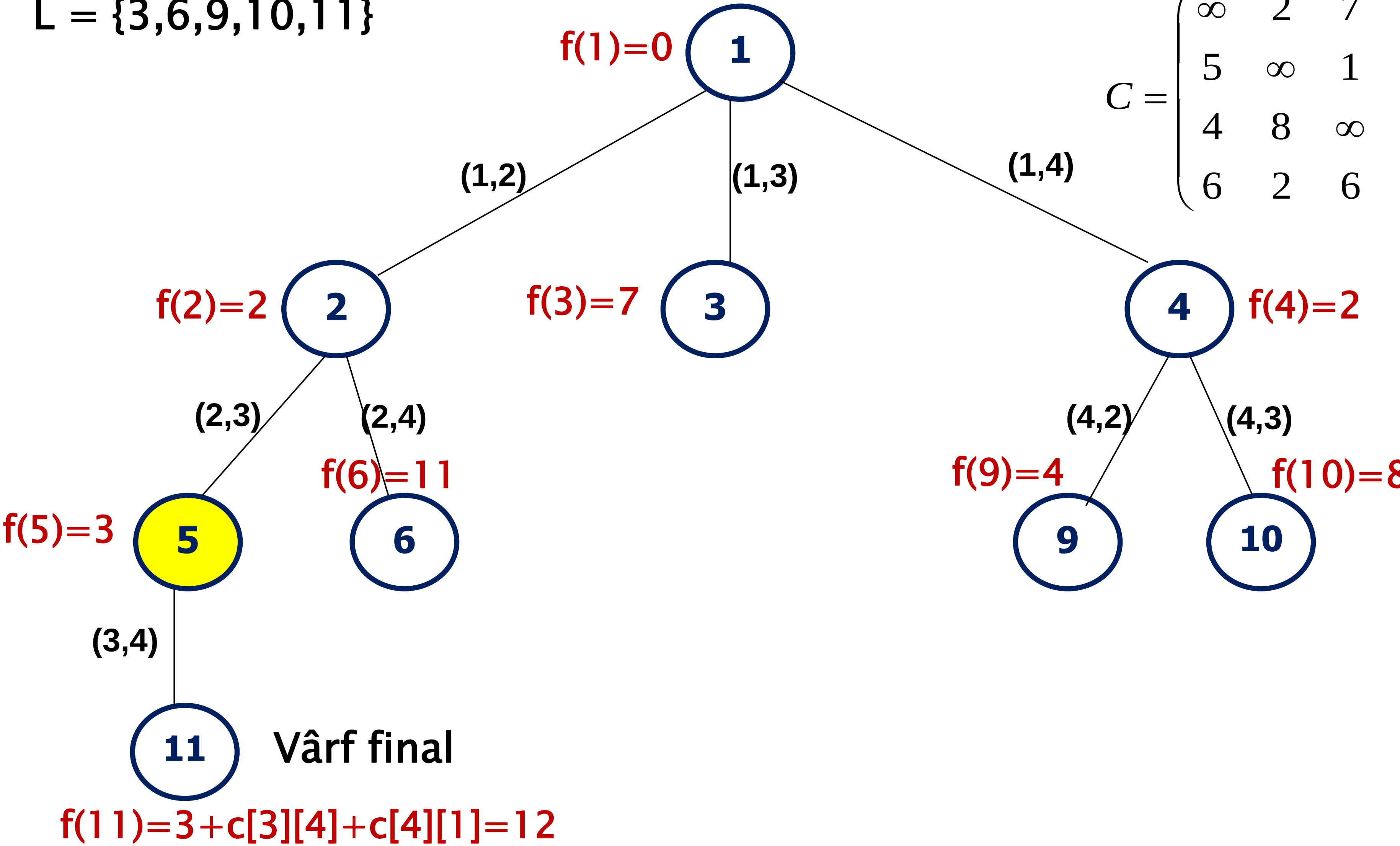
$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$



min= ∞

L = {3,6,9,10,11}

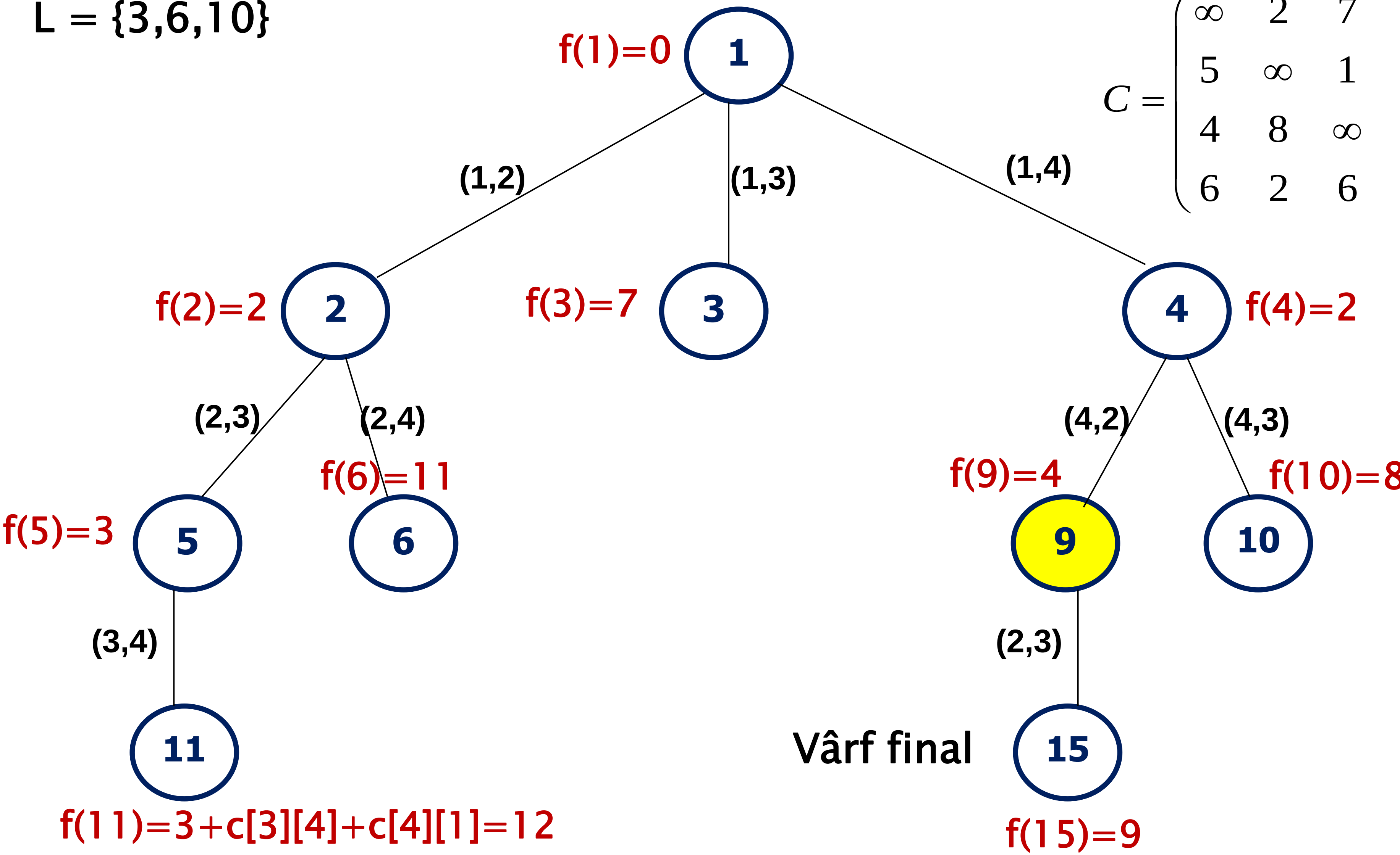
$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$



min=12 – > eliminăm vârfurile cu f mai mare sau egal cu12

$L = \{3,6,10\}$

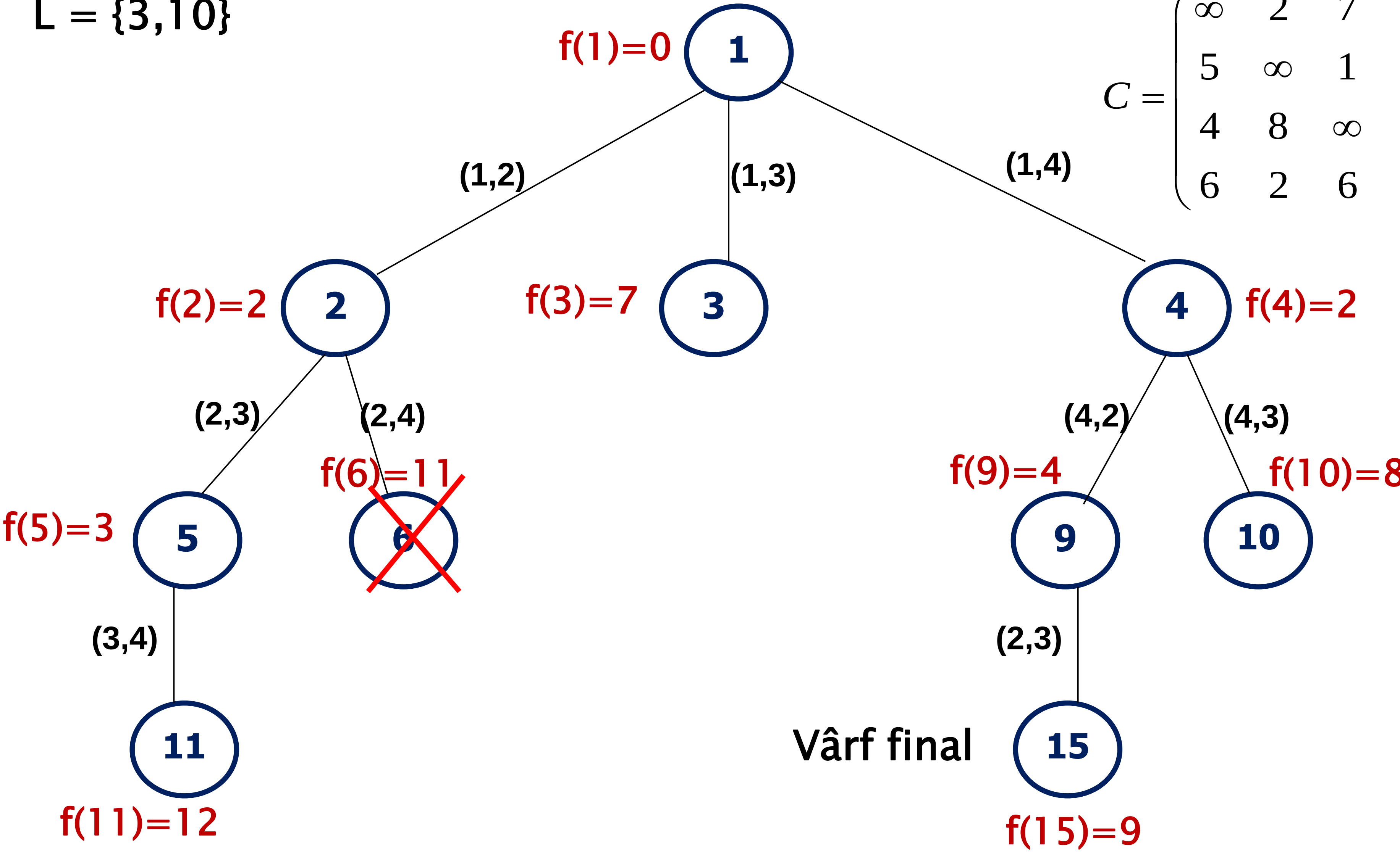
$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$



min=min{12, 9} = 9 -> eliminăm vârfurile cu f mai mare sau egal cu 9

L = {3,10}

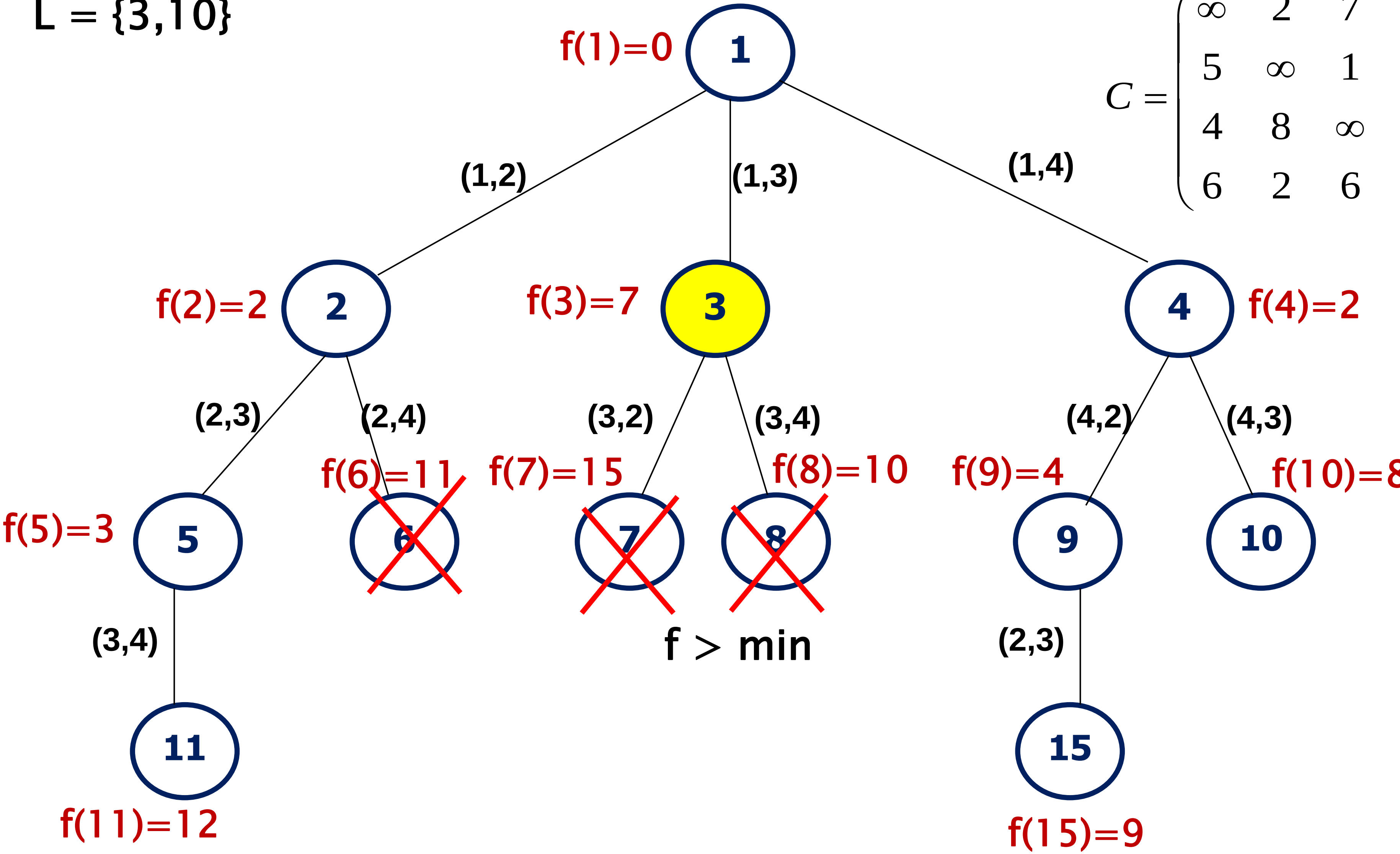
$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$



min= 9

$L = \{3,10\}$

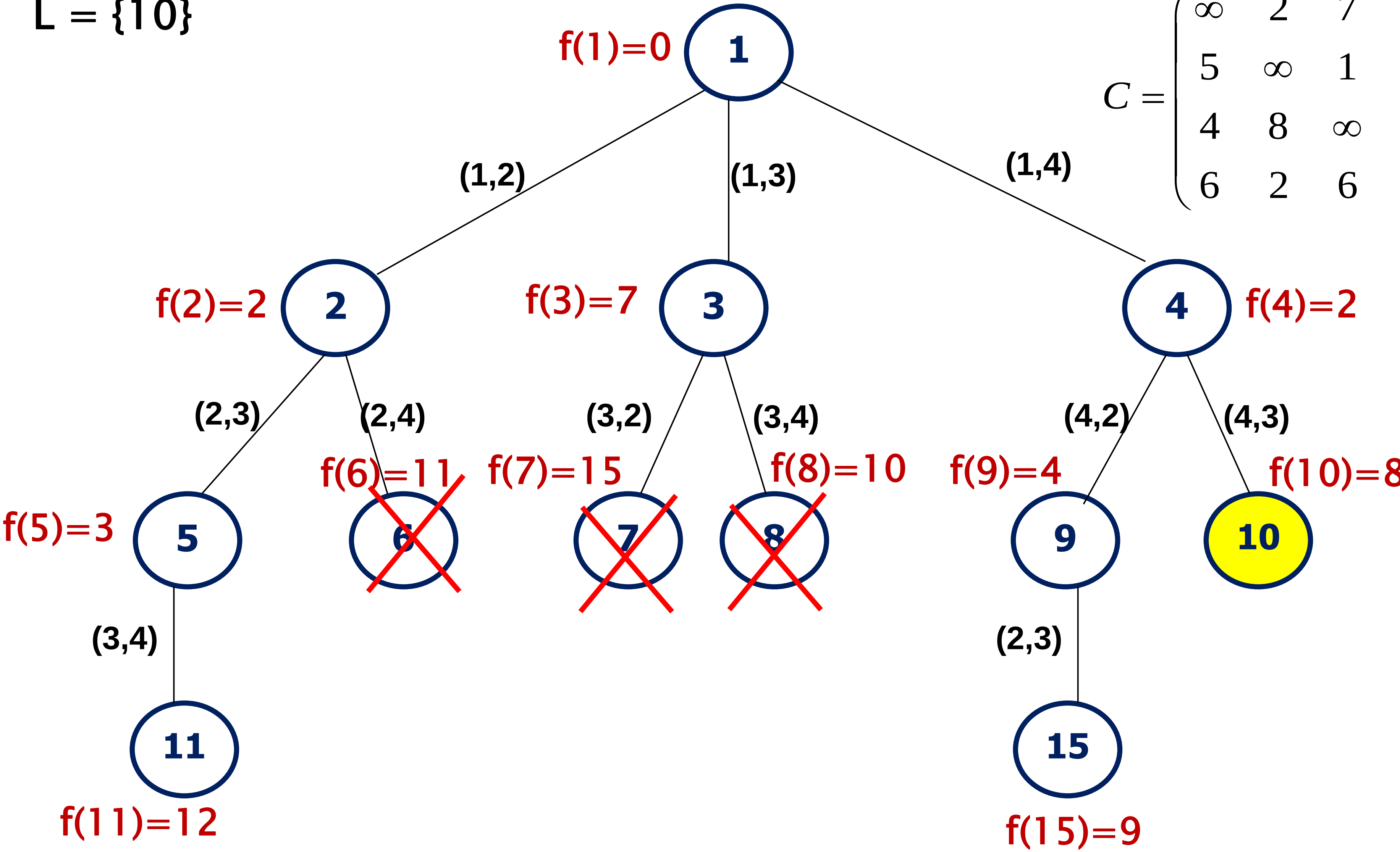
$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$



min= 9

L = {10}

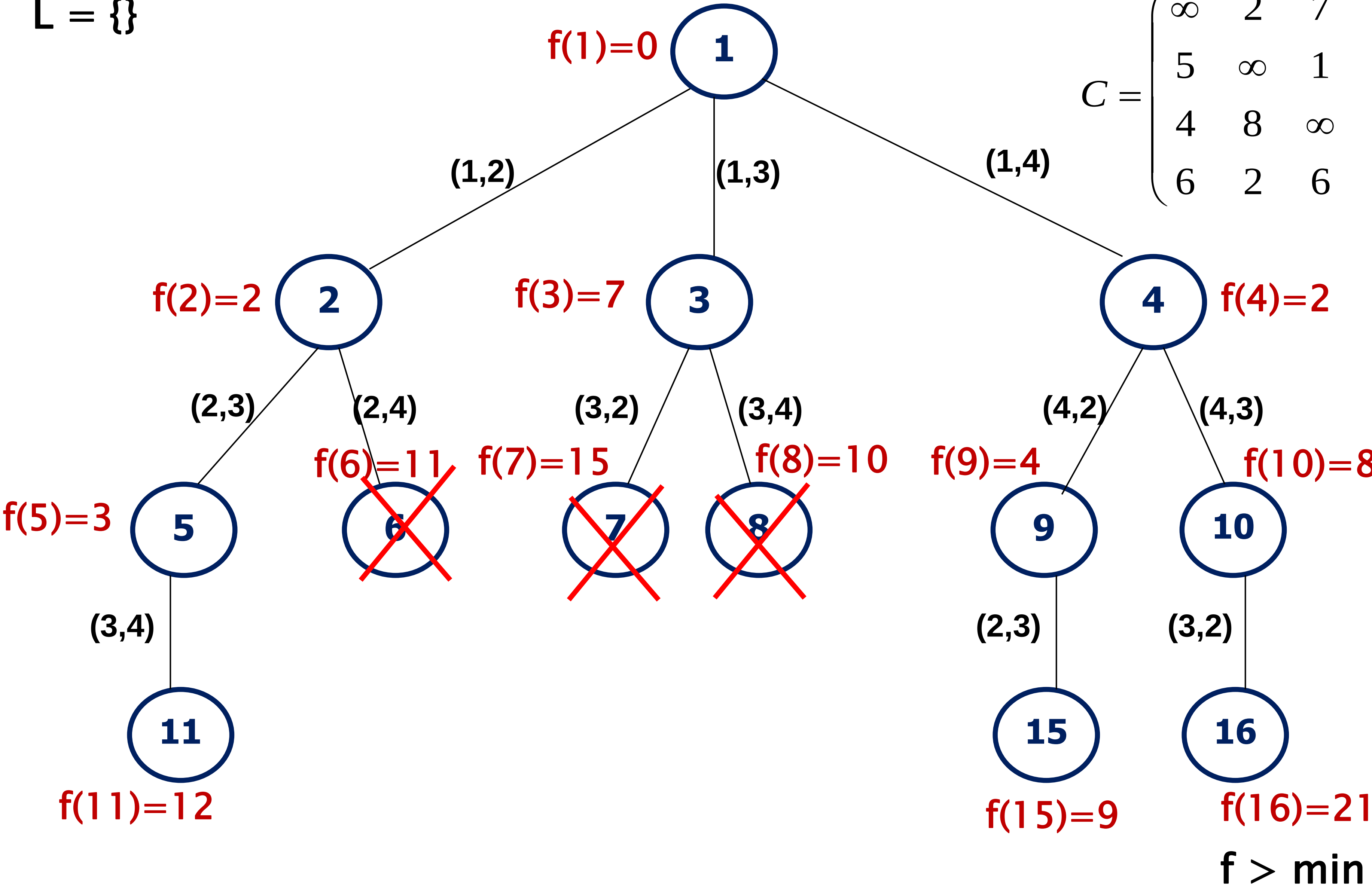
$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$



min= 9

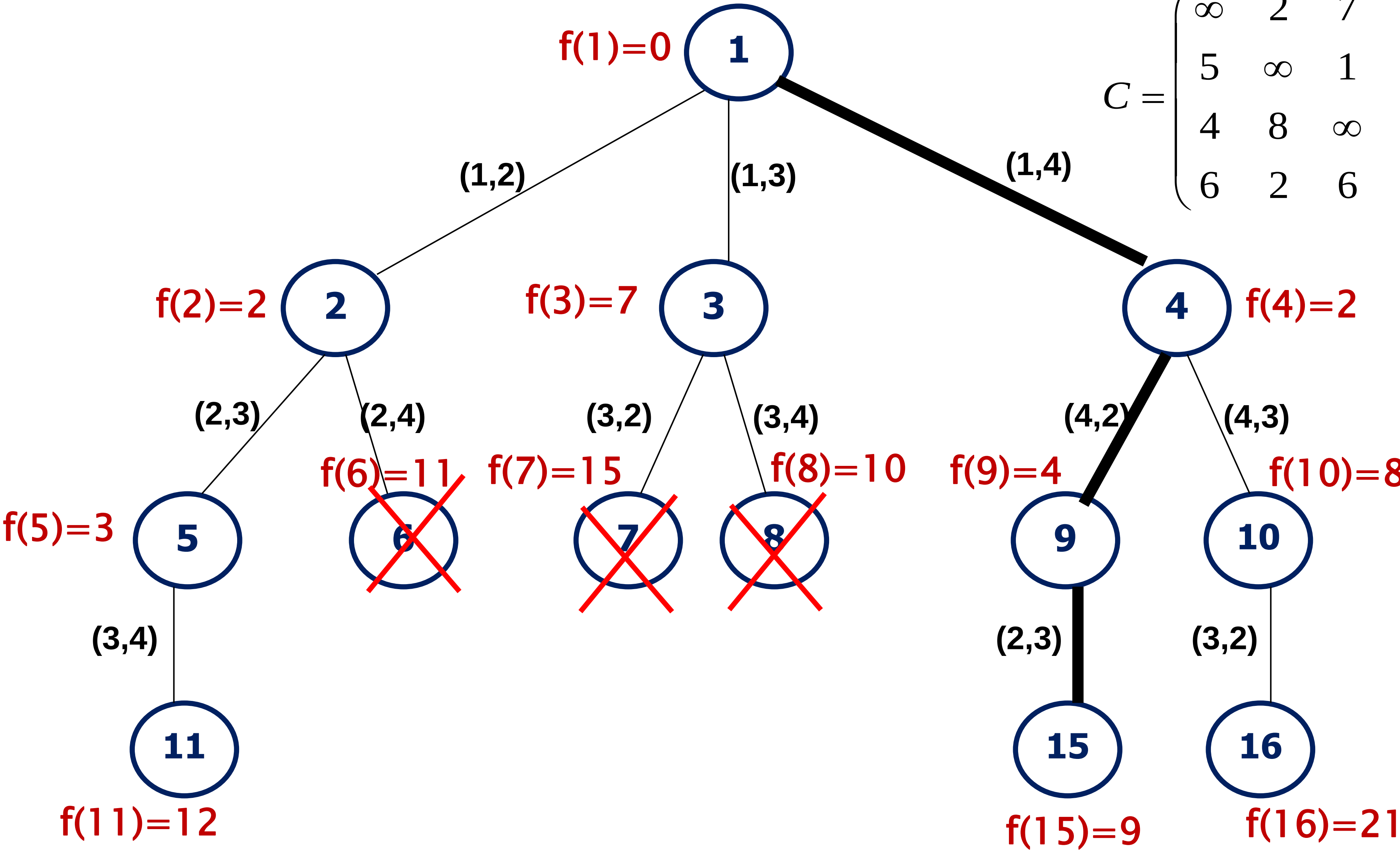
L = {}

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

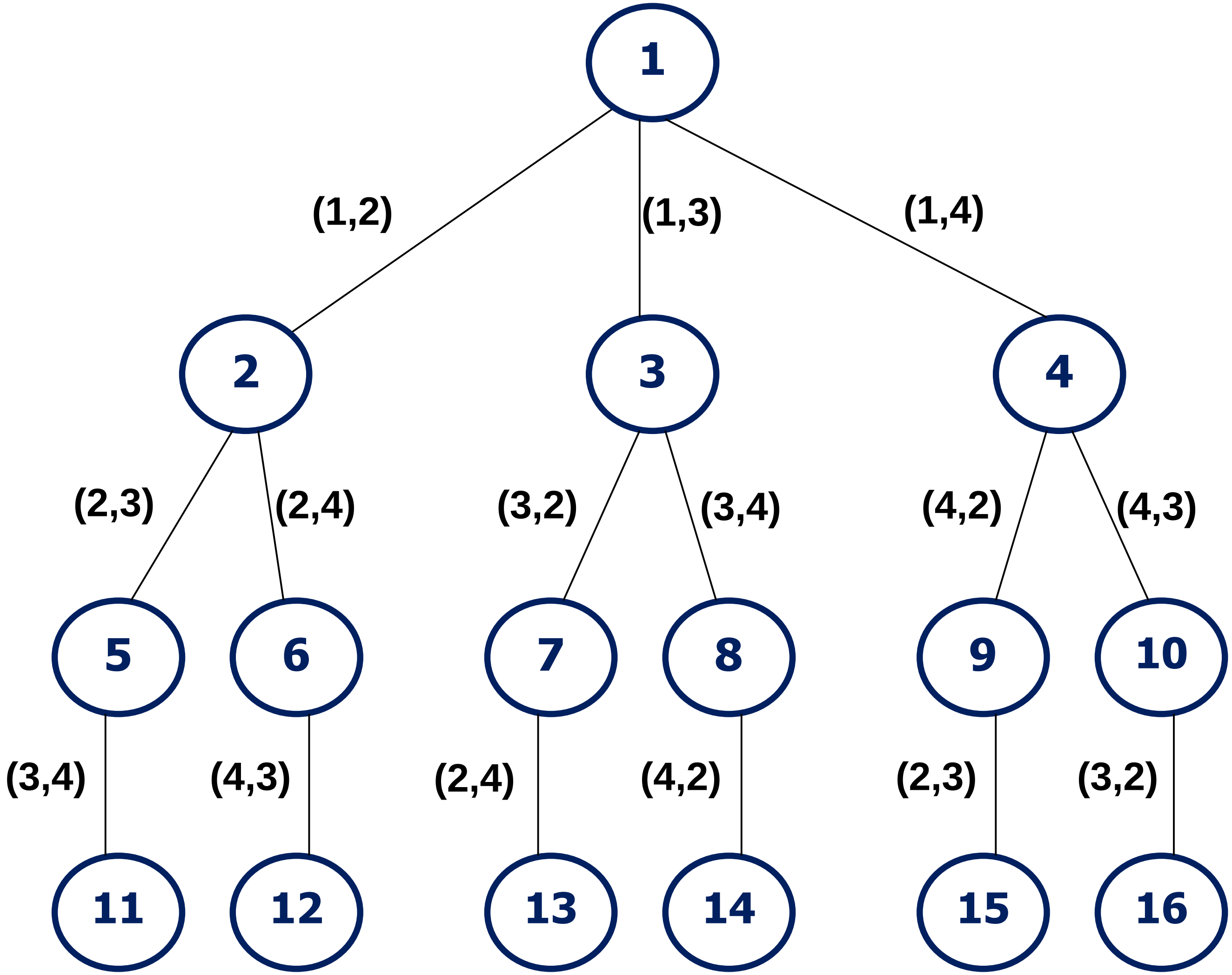


min= 9

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

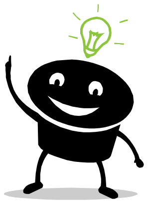


min= 9



Au fost excluse puţine vârfuri de la expandare

Exemplu – circuit hamiltonian minim TSP



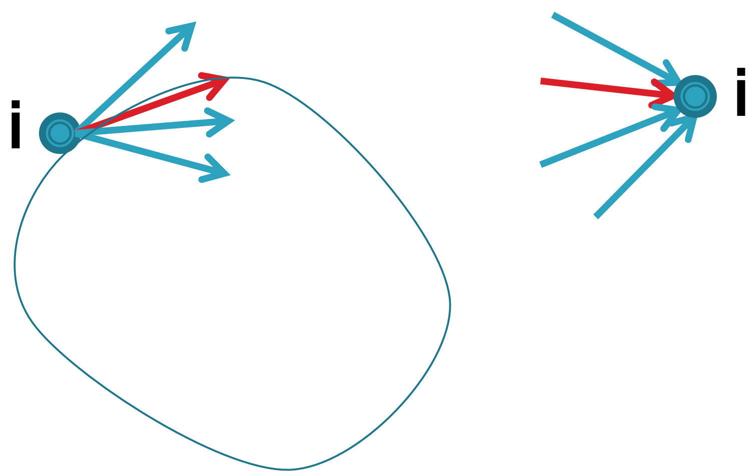
Euristici h mai precise:

- Un circuit hamiltonian conţine pentru fiecare vârf i un **unic arc care intră în i şi un unic arc care iese din i**

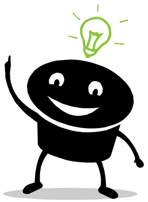
Circuit hamiltonian minim TSP

- **Observaţie.** Dacă micşorăm toate elementele unei linii i din matricea de costuri C cu α, orice circuit hamiltonian va avea costul micşorat cu α
  - **vom considera α = costul minim al unui arc care iese din i = minimul de pe linia i**

- Similar pentru coloane



Exemplu – circuit hamiltonian minim TSP



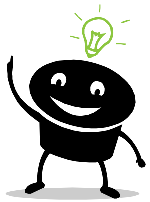
Euristici h mai precise:

- Construim **matricea costurilor reduse** astfel:

  – pentru fiecare linie considerăm α **minimul** de pe acea linie şi scădem α din toate elementele de pe linie, dacă $\alpha < \infty$

  - **astfel, linia i va avea cel puţin un 0, corespunzător arcului minim care iese din i**

Exemplu – circuit hamiltonian minim TSP



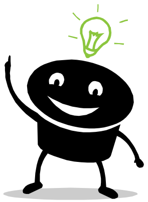
Euristici h mai precise:

- Construim **matricea costurilor reduse** astfel:

  – pentru fiecare linie considerăm α **minimul** de pe acea linie şi scădem α din toate elementele de pe linie, dacă $\alpha < \infty \Rightarrow$ pe fiecare linie care nu are toate elementele ∞ va fi cel puţin un 0

  – similar pentru fiecare coloană (pe care nu apare încă un 0)

Exemplu – circuit hamiltonian minim TSP



Euristici h mai precise:

- Construim **matricea costurilor reduse** astfel:

  – pentru fiecare linie considerăm α **minimul** de pe acea linie şi scădem α din toate elementele de pe linie, dacă $\alpha < \infty \Rightarrow$ pe fiecare linie care nu are toate elementele ∞ va fi cel puţin un 0

  – similar pentru fiecare coloană (pe care nu apare încă un 0)

- Suma numerelor cu care am redus matricea este limita inferioară pentru CHM

Circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Reducem linia 1 cu 2

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Reducem linia 2 cu 1

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 4 & \infty & 0 & 8 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Reducem linia 3 cu 3

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 4 & \infty & 0 & 8 \\ 1 & 5 & \infty & 0 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Reducem linia 4 cu 2

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 4 & \infty & 0 & 8 \\ 1 & 5 & \infty & 0 \\ 4 & 0 & 4 & \infty \end{pmatrix}$$

Circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Reducem linia 1 cu 2

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Reducem linia 2 cu 1

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 4 & \infty & 0 & 8 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Reducem linia 3 cu 3

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 4 & \infty & 0 & 8 \\ 1 & 5 & \infty & 0 \\ 6 & 2 & 6 & \infty \end{pmatrix}$$

Reducem linia 4 cu 2

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 4 & \infty & 0 & 8 \\ 1 & 5 & \infty & 0 \\ 4 & 0 & 4 & \infty \end{pmatrix}$$

Reducem coloana 1 cu 1

$$\begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

Circuit hamiltonian minim TSP

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix} \qquad \textbf{matricea redusă} = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

Cantitatea totală cu care s-a redus matricea =
$2 + 1 + 3 + 2 + 1 = 9$

Algoritmul Branch and Bound pentru TSP

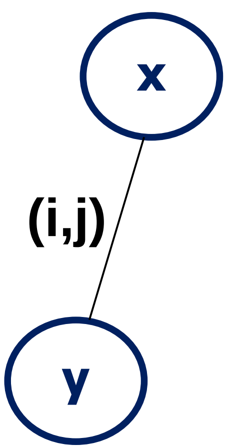
- f(rad) = cantitatea totală cu care s-a redus matricea C = limită inferioară pentru CHM

- Unui vârf x îi asociem
  - o matrice de costuri redusă M_x (dacă nu este frunză)
  - o valoare f(x) calculată după cum urmează

Algoritmul Branch and Bound pentru TSP

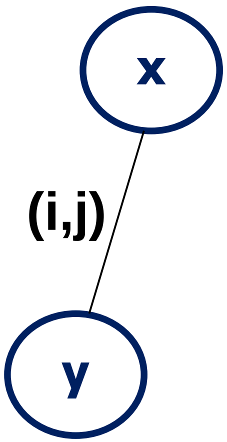
- Când parcurgem ramura corespunzătoare cazului când (i,j) aparţine traseului hamiltonian trebuie să ne asigurăm că
  - **Nu mai folosim arce de care ies din vârful i sau intră în j**
  - **Nu ajungem din j în 1 înainte de a parcurge toate vârfurile**
  - **Actualizăm limita inferioară a lungimii circuitului hamiltonian**

Astfel:



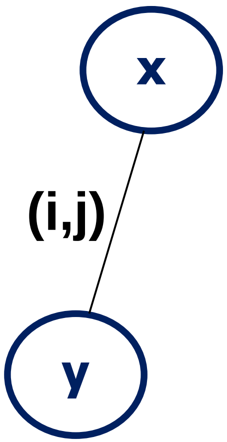
Algoritmul Branch and Bound pentru TSP

Pentru un vârf y din arborele BB al cărui tată este x şi muchia (x,y) este etichetată cu (i,j) construim M_y pornind de la M_x astfel:



Algoritmul Branch and Bound pentru TSP

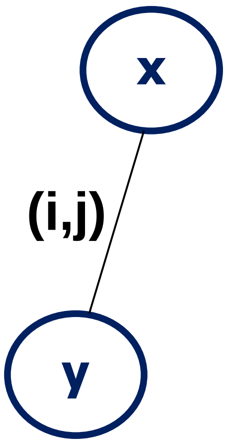
Pentru un vârf y din arborele BB al cărui tată este x şi muchia (x,y) este etichetată cu (i,j) construim M_y pornind de la M_x astfel:



- elementele liniei i devin ∞
  - **mergem sigur către j din i, nu mai folosim în estimare alte arce care ies din i**

Algoritmul Branch and Bound pentru TSP

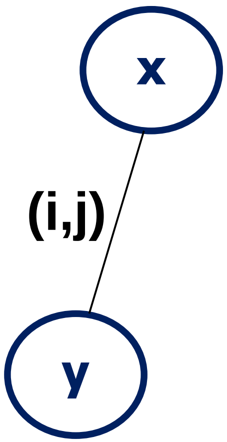
Pentru un vârf y din arborele BB al cărui tată este x şi muchia (x,y) este etichetată cu (i,j) construim M_y pornind de la M_x astfel:



- elementele liniei i devin ∞
  - **mergem sigur către j din i, nu mai folosim în estimare alte arce care ies din i**
- elementele coloanei j devin ∞
  - **am ajuns sigur în j**

Algoritmul Branch and Bound pentru TSP

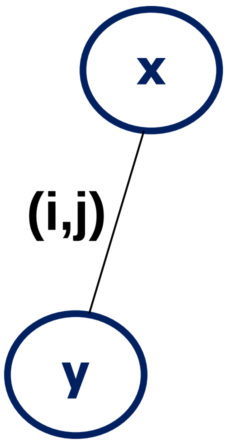
Pentru un vârf y din arborele BB al cărui tată este x şi muchia (x,y) este etichetată cu (i,j) construim M_y pornind de la M_x astfel:



- elementele liniei i devin ∞
  - **mergem sigur către j din i**
- elementele coloanei j devin ∞
  - **am ajuns sigur în j**
- $M_y(j,1) \leftarrow \infty$ **pentru a nu reveni prematur în rădăcina 1**

Algoritmul Branch and Bound pentru TSP

Pentru un vârf y din arborele BB al cărui tată este x şi muchia (x,y) este etichetată cu (i,j) construim M_y pornind de la M_x astfel:



- elementele liniei i devin ∞
  - **mergem sigur către j din i**
- elementele coloanei j devin ∞
  - **am ajuns sigur în j**
- $M_y(j,1) \leftarrow \infty$ **pentru a nu reveni prematur în rădăcina 1**
- reducem matricea
  - **fie r cantitatea cu care s-a redus matricea**
- $f(y) \leftarrow f(x) + M_x(i,j) + r$

Algoritmul Branch and Bound pentru TSP

- Avem

  f(x) $\leq$ cost CHM corespunzător lui x

- Dacă x este frunză avem

  f(x) = c(x) = cost CHM

$$C = \begin{pmatrix} \infty & 2 & 7 & 2 \\ 5 & \infty & 1 & 9 \\ 4 & 8 & \infty & 3 \\ 6 & 2 & 6 & \infty \end{pmatrix} \qquad M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

$$f(1) = 2 + 1 + 3 + 2 + 1 = 9$$

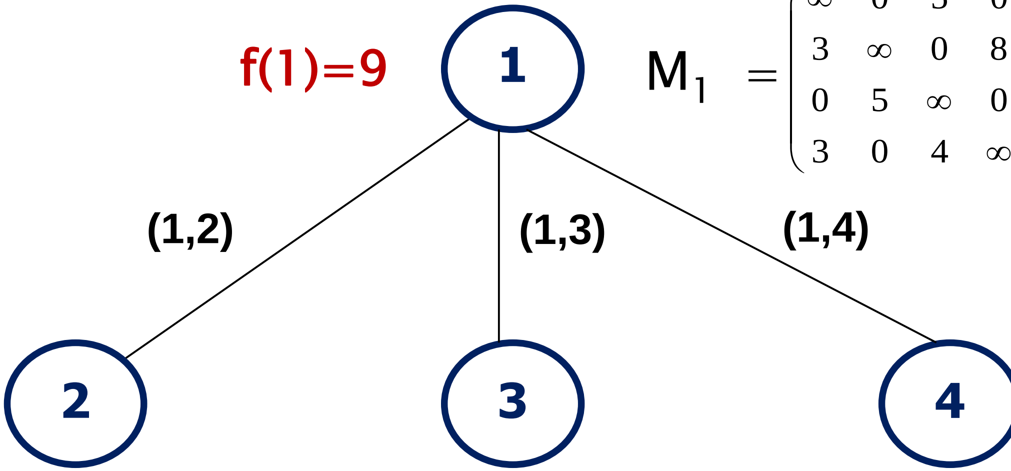
f(1)=9

1

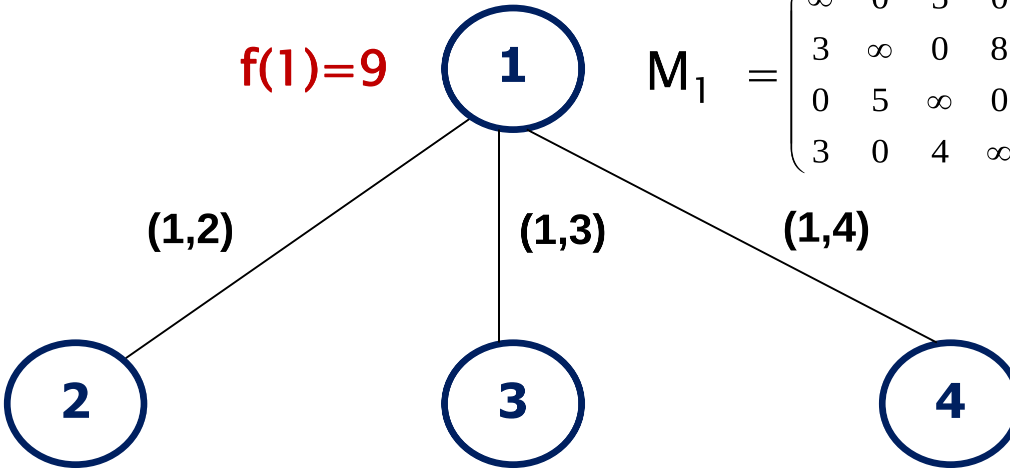
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

f(1)=9

$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$



L = {2, 3, 4}



$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

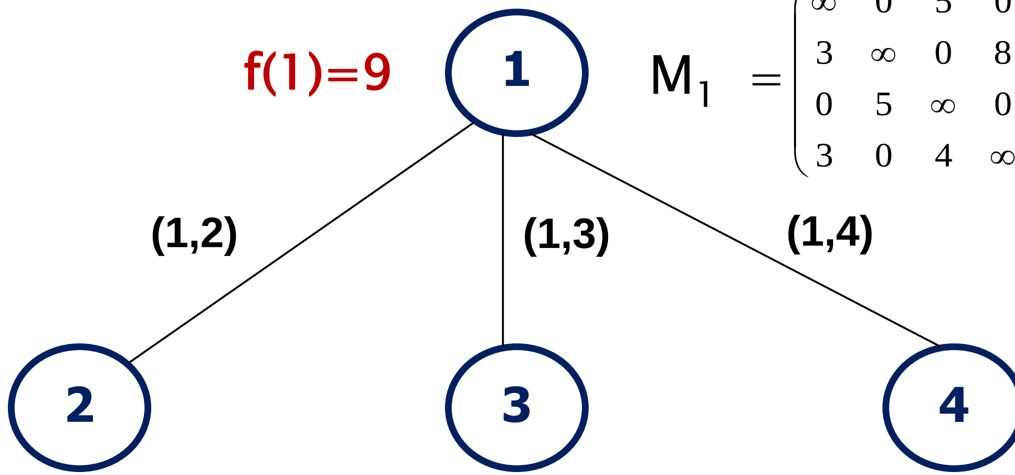
L = {2, 3, 4}

Calculăm f(2)

$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

$f(1)=9$

$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$



$L = \{2, 3, 4\}$

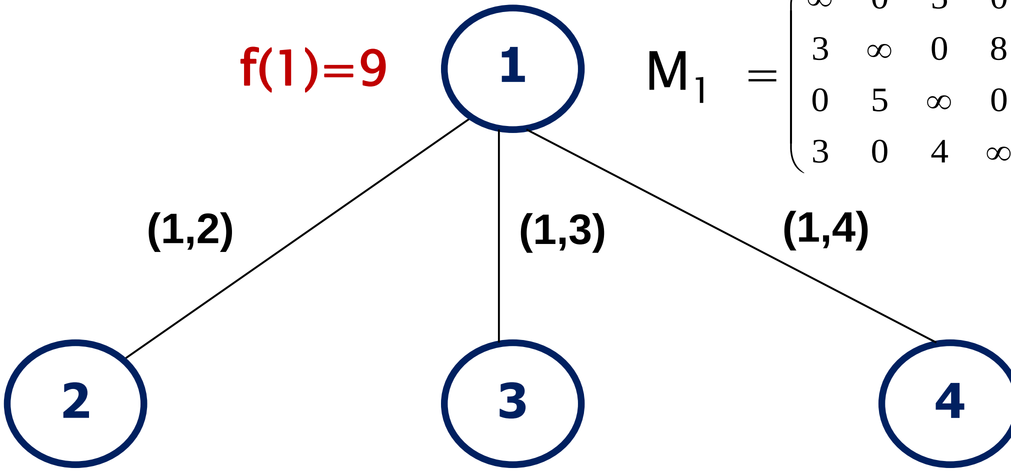
Calculăm f(2)

$$\begin{pmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 8 \\ 0 & \infty & \infty & 0 \\ 3 & \infty & 4 & \infty \end{pmatrix}$$

- Linia 1 și coloana 2 devin ∞
- Elementul (2, 1) devine ∞
- Reducem linia 4 cu 3

$f(1)=9$

$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

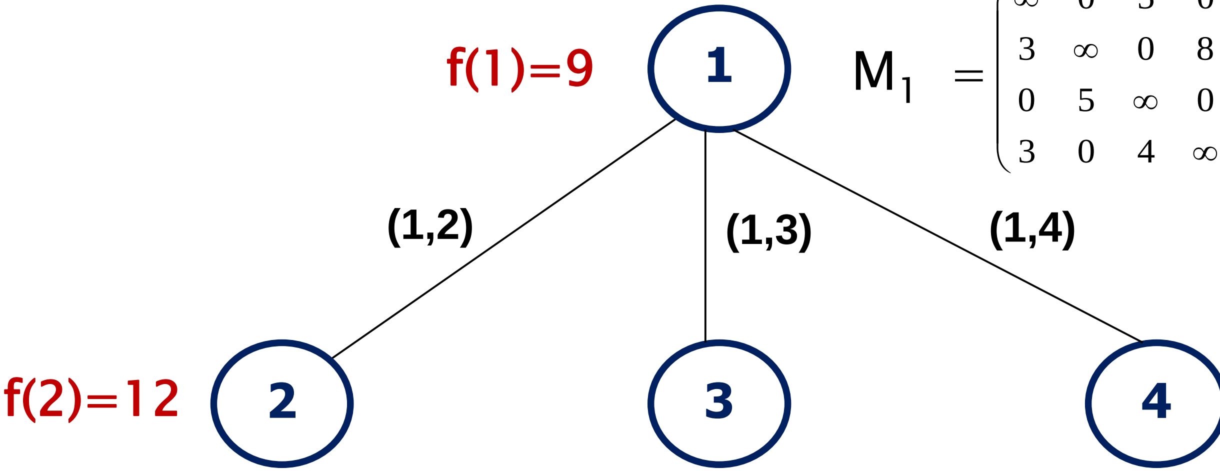


L = {2, 3, 4}

Calculăm f(2)

$$\begin{pmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 8 \\ 0 & \infty & \infty & 0 \\ 0 & \infty & 1 & \infty \end{pmatrix}$$

- Linia 1 și coloana 2 devin ∞
- Elementul (2, 1) devine ∞
- Reducem linia 4 cu 3



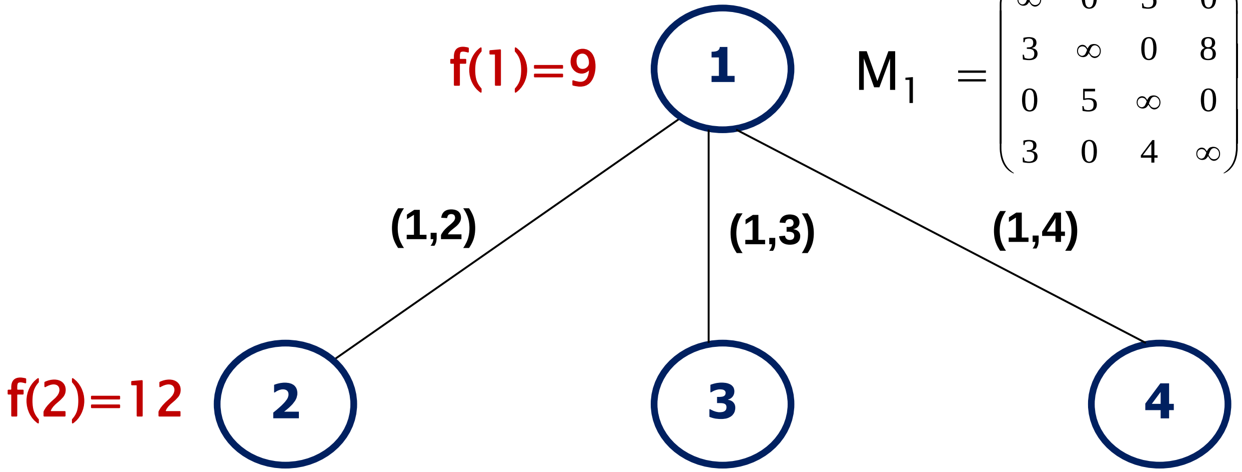
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}

Calculăm f(2)

$$M_2 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 8 \\ 0 & \infty & \infty & 0 \\ 0 & \infty & 1 & \infty \end{pmatrix}$$

- Linia 1 și coloana 2 devin ∞
- Elementul (2, 1) devine ∞
- Reducem linia 4 cu 3
- Obținem $f(2) = f(1)+r + M_1(1,2) = 9+3+0=12$



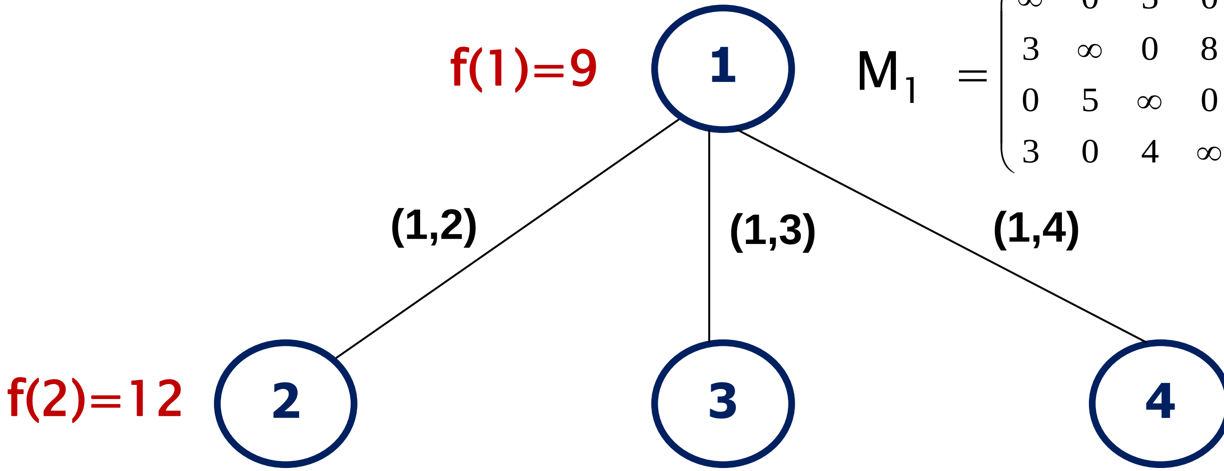
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}

Calculăm f(3)

$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

- Linia 1 și coloana 3 devin ∞
- Elementul (3, 1) devine ∞



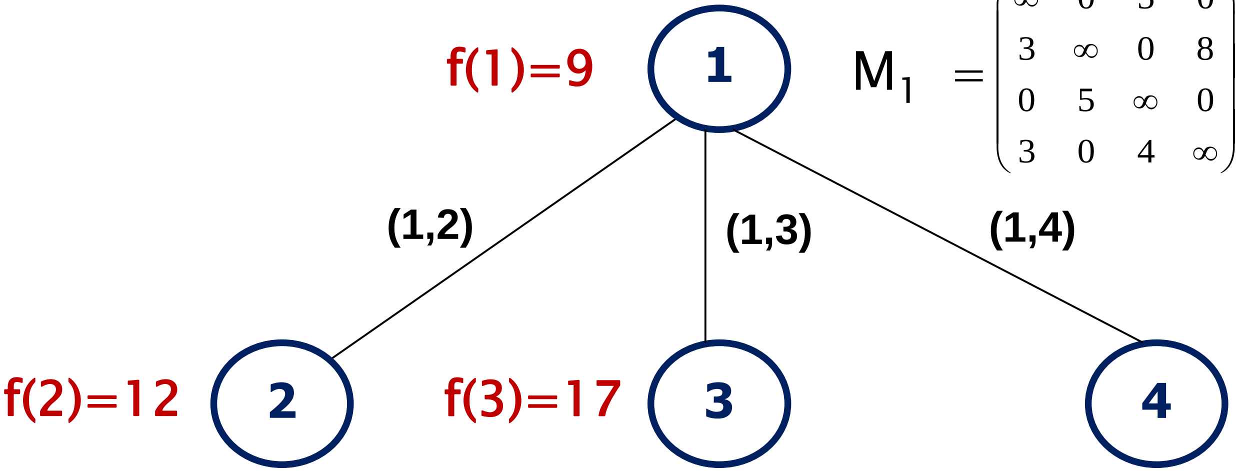
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}

Calculăm f(3)

$$\begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & 8 \\ \infty & 5 & \infty & 0 \\ 3 & 0 & \infty & \infty \end{pmatrix}$$

- Linia 1 și coloana 3 devin ∞
- Elementul (3, 1) devine ∞
- Reducem linia 2 cu 3



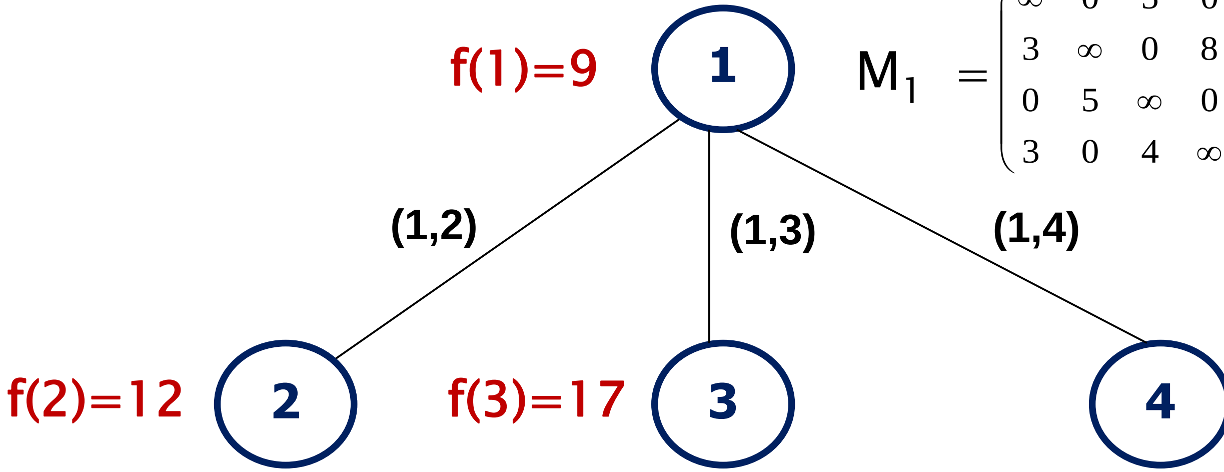
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}

Calculăm f(3)

$$\begin{pmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 5 \\ \infty & 5 & \infty & 0 \\ 3 & 0 & \infty & \infty \end{pmatrix}$$

- Linia 1 și coloana 3 devin ∞
- Elementul (3, 1) devine ∞
- Reducem linia 2 cu 3
- Obținem $f(3) = f(1)+r + M_1(1,3) = 9+3+5=17$



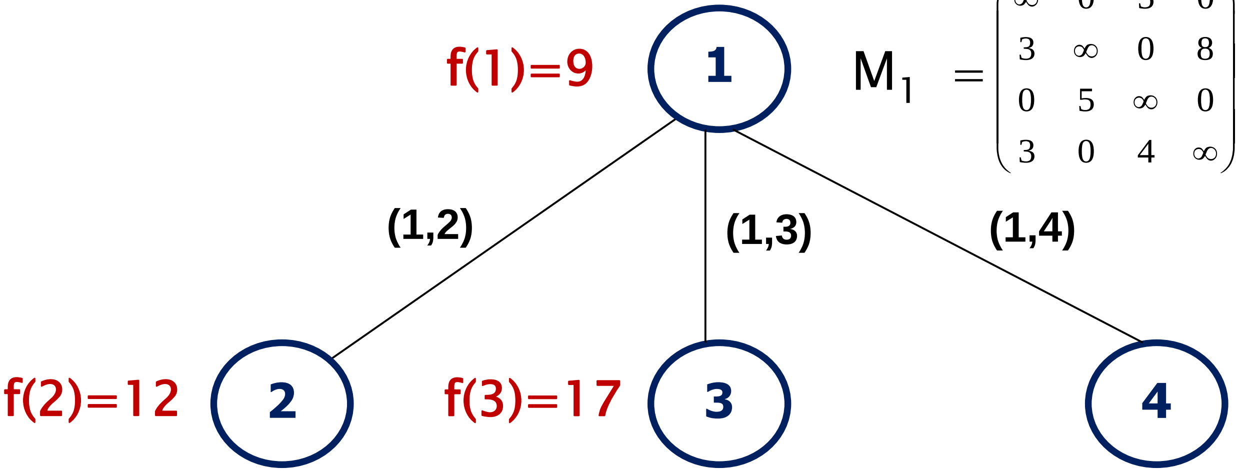
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}

Calculăm f(4)

$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

- Linia 1 și coloana 4 devin ∞
- Elementul (4, 1) devine ∞



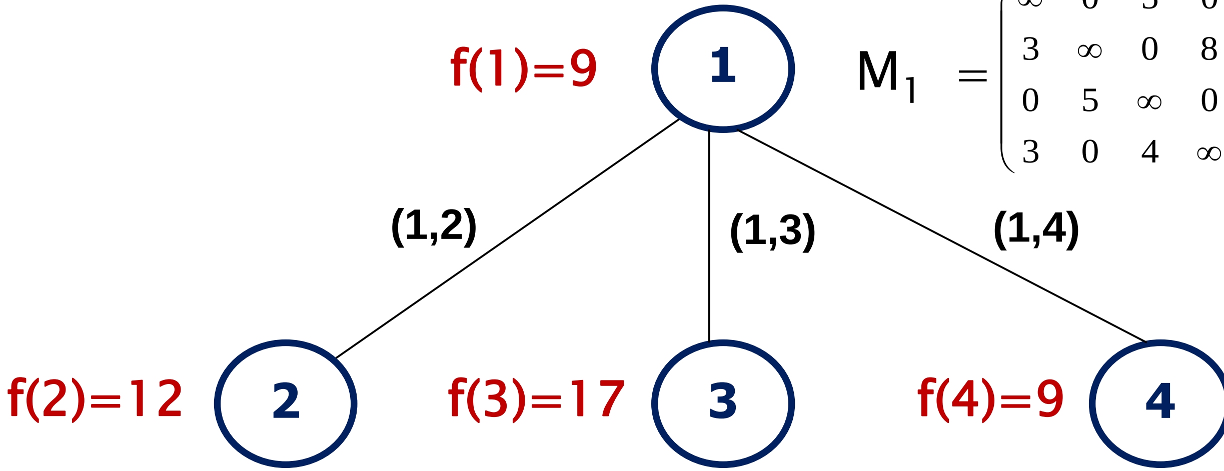
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}

Calculăm f(4)

$$\begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ \infty & 0 & 4 & \infty \end{pmatrix}$$

- Linia 1 și coloana 4 devin ∞
- Elementul (4, 1) devine ∞



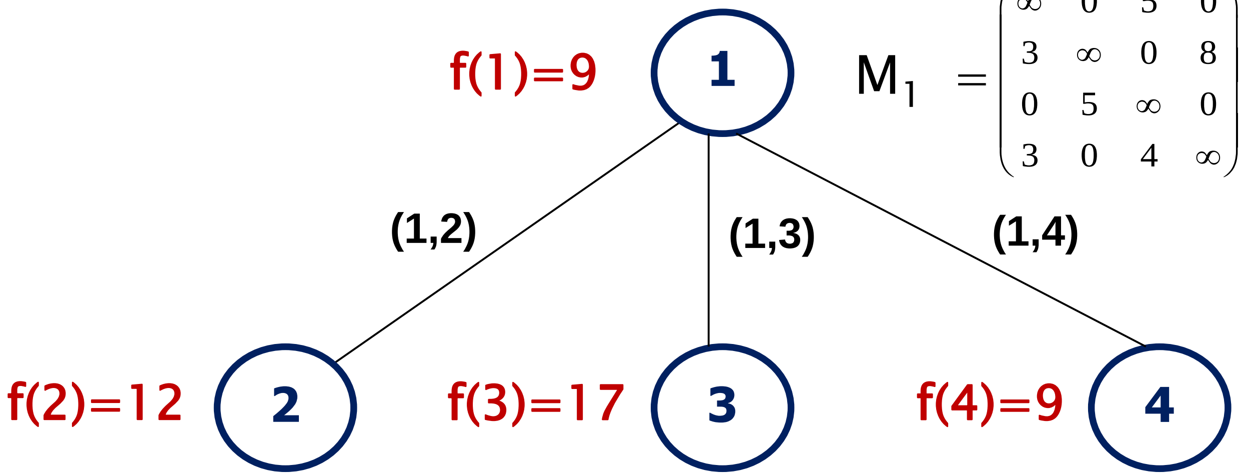
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}

Calculăm f(4)

$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ \infty & 0 & 4 & \infty \end{pmatrix}$$

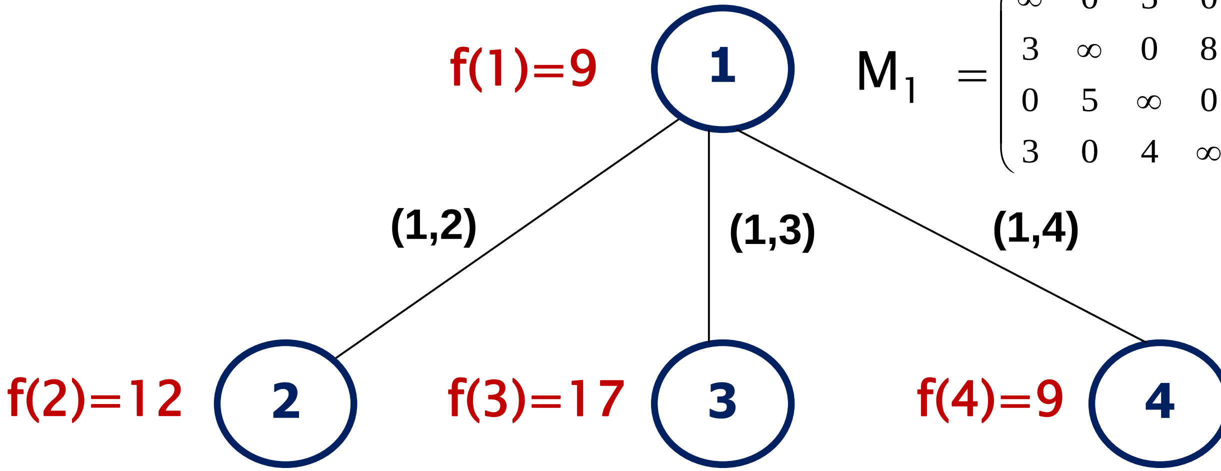
- Linia 1 și coloana 4 devin ∞
- Elementul (4, 1) devine ∞
- Nu sunt necesare reduceri
- Obținem $f(4) = f(1) + M_1(1,4) = 9 + 0 = 9$



$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ \infty & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}



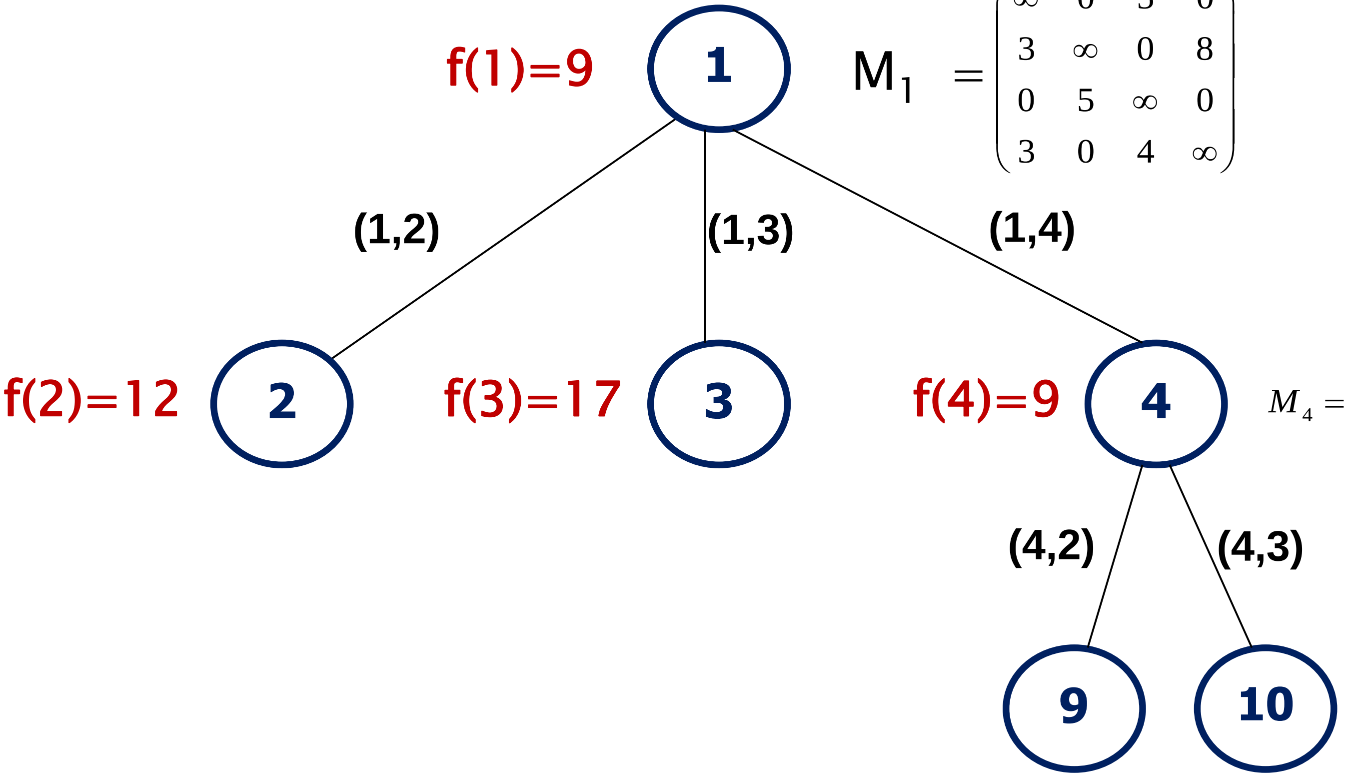
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ \infty & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 4}

Extragem din L vârful cu f minim -> 4

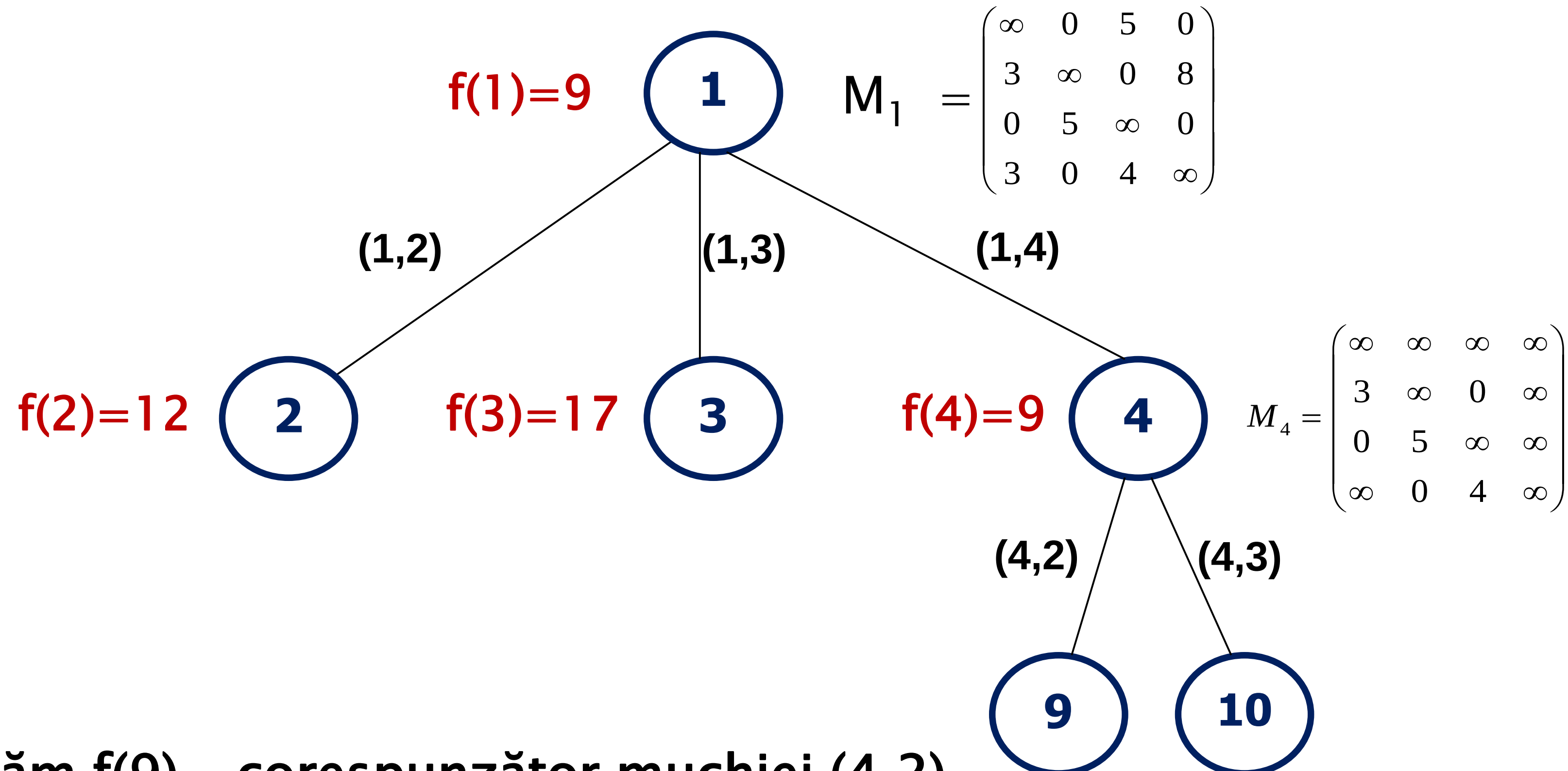
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$



$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ \infty & 0 & 4 & \infty \end{pmatrix}$$

$L = \{2, 3, 9, 10\}$

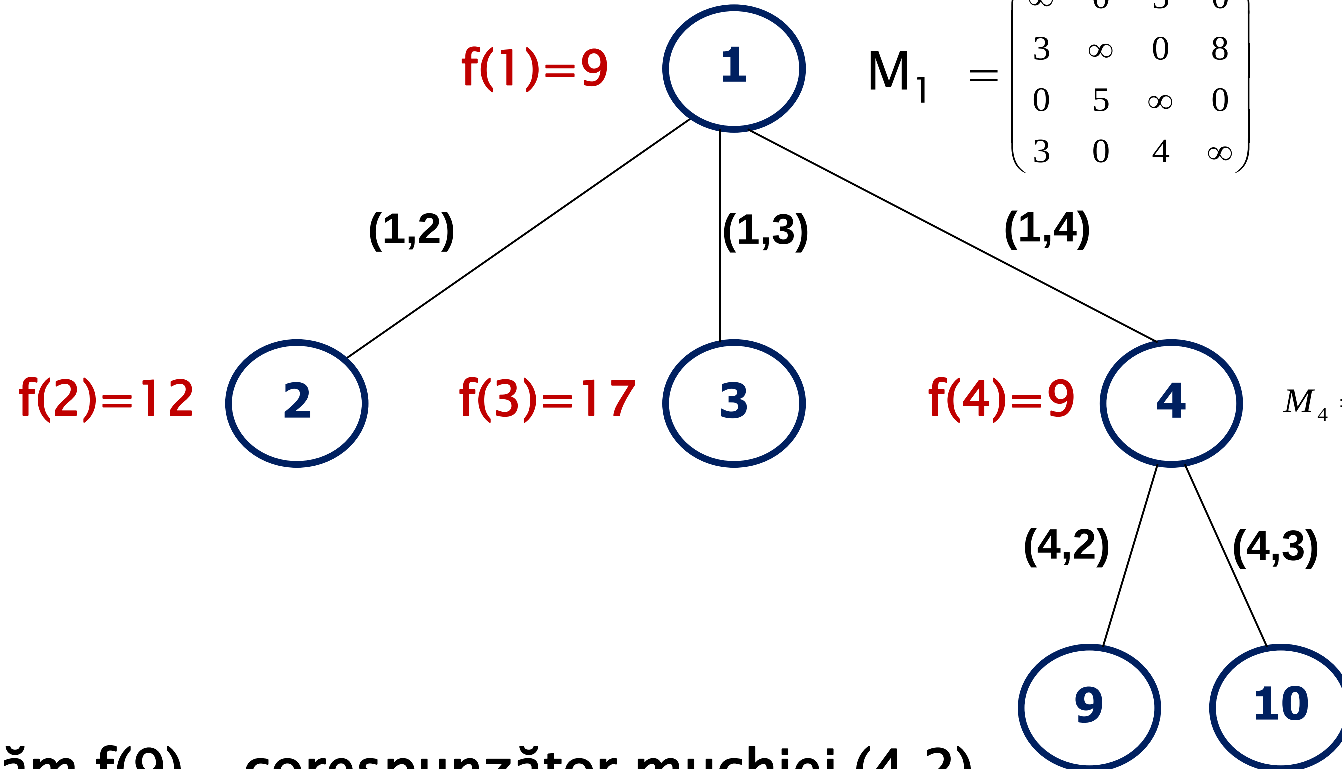
Calculăm f(9) și f(10) analog (pornind de la M_4)



Calculăm f(9) – corespunzător muchiei (4,2)

$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

- Linia 4 și coloana 2 devin ∞
- Elementul (2, 1) devine ∞



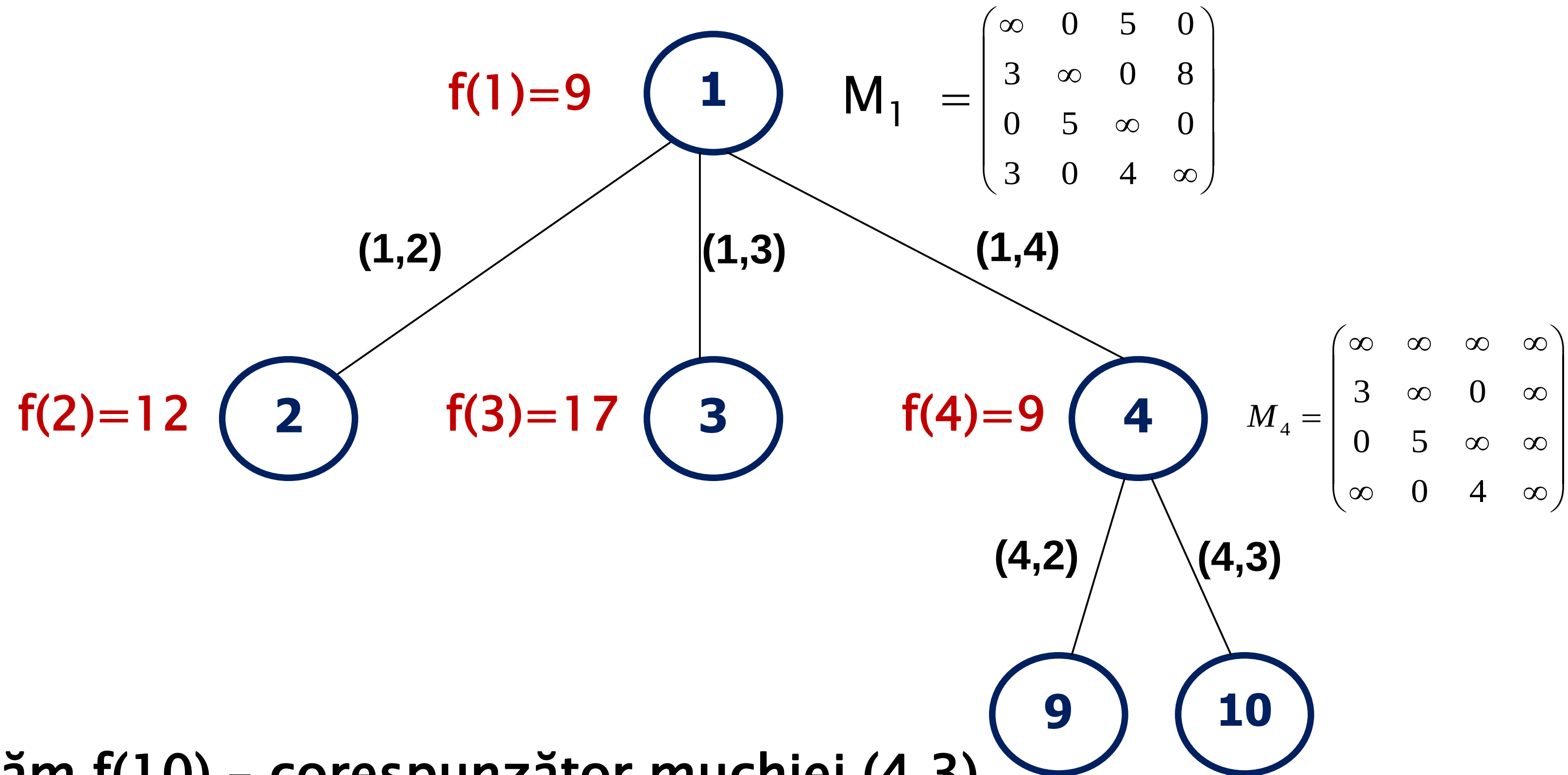
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ \infty & 0 & 4 & \infty \end{pmatrix}$$

Calculăm f(9) – corespunzător muchiei (4,2)

$$M_9 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty \\ 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \end{pmatrix}$$

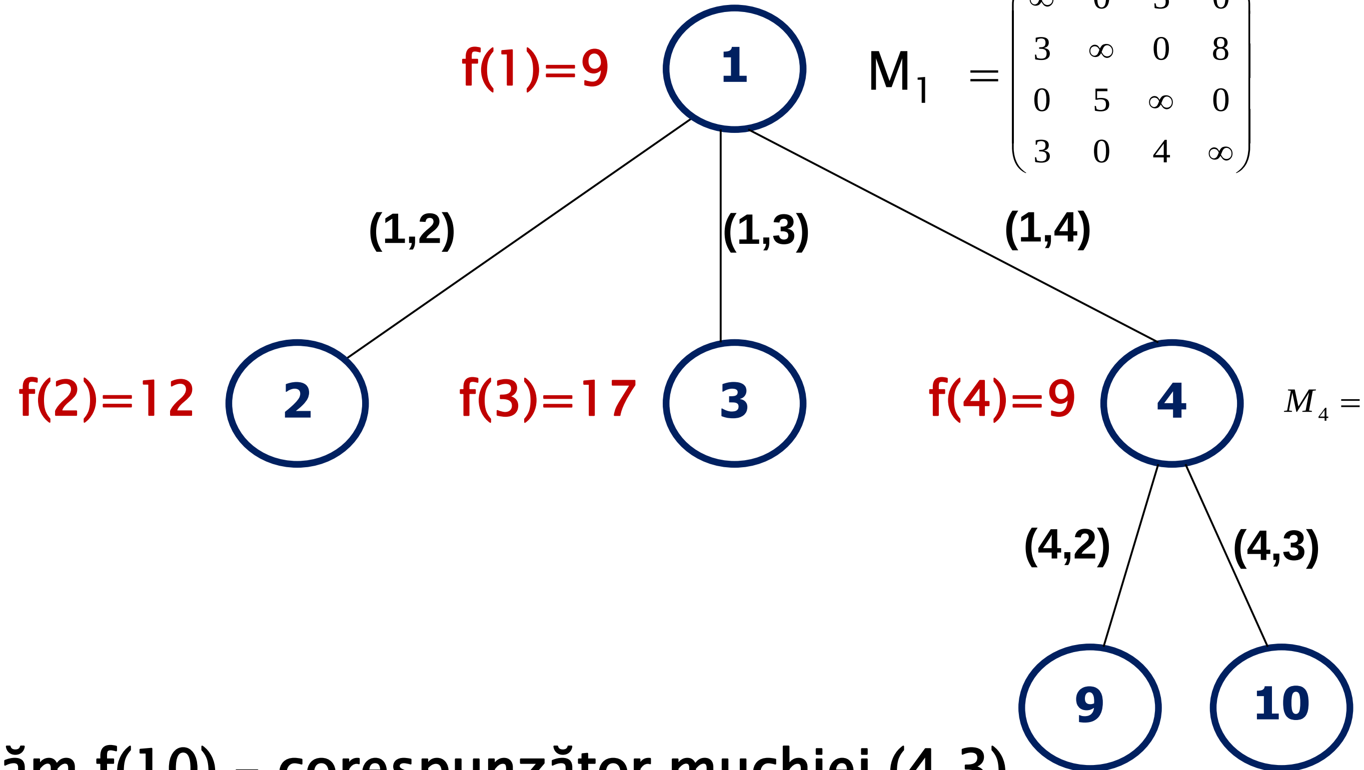
- Linia 4 și coloana 2 devin ∞
- Elementul (2, 1) devine ∞
- Nu sunt necesare reduceri
- $f(9) = f(4) + M_4(4,2) = 9 + 0 = 9$



Calculăm f(10) – corespunzător muchiei (4,3)

$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

- Linia 4 și coloana 3 devin ∞
- Elementul (3, 1) devine ∞



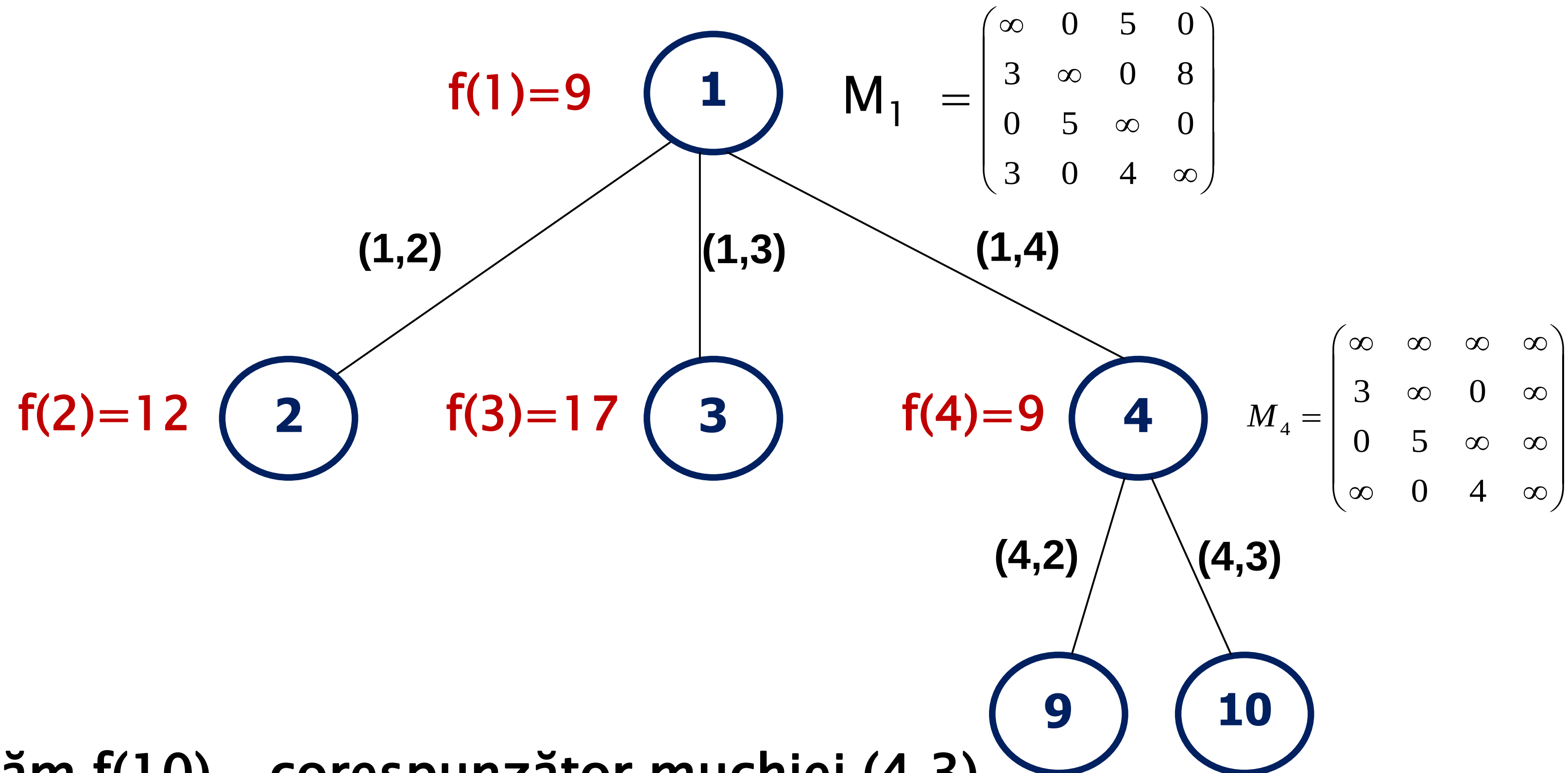
$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ \infty & 0 & 4 & \infty \end{pmatrix}$$

Calculăm f(10) – corespunzător muchiei (4,3)

$$\begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & \infty \\ \infty & 5 & \infty & \infty \\ \infty & \infty & \infty & \infty \end{pmatrix}$$

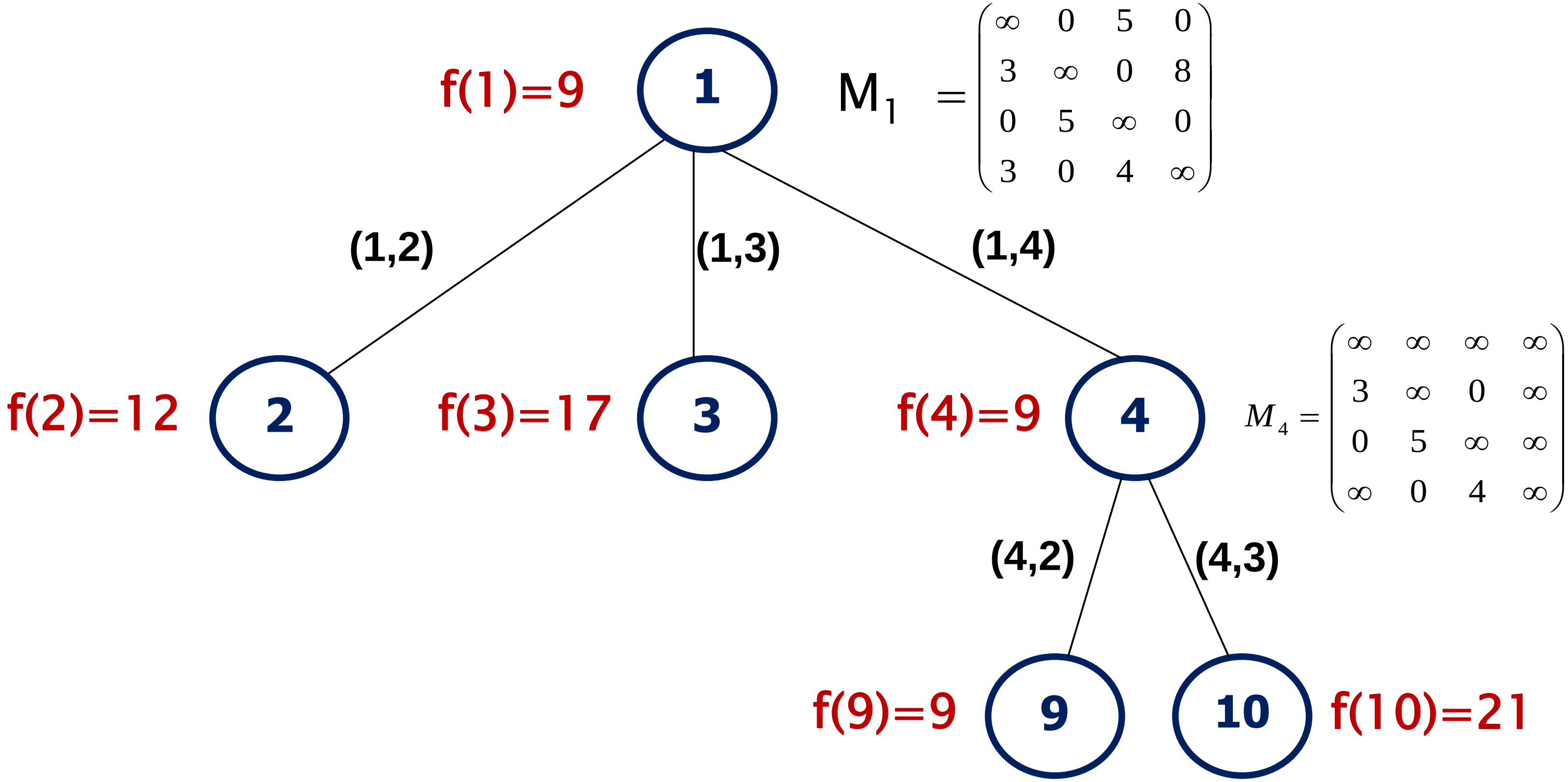
- Linia 4 și coloana 3 devin ∞
- Elementul (3, 1) devine ∞
- Reducem linia 2 cu 3 și linia 3 cu 5



Calculăm f(10) – corespunzător muchiei (4,3)

$$M_{10} = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty \end{pmatrix}$$

- Linia 4 și coloana 3 devin ∞
- Elementul (3, 1) devine ∞
- Reducem linia 2 cu 3 și linia 3 cu 5
- $f(10)=f(4)+r+M_4(4,3)=9+(3+5)+4=21$

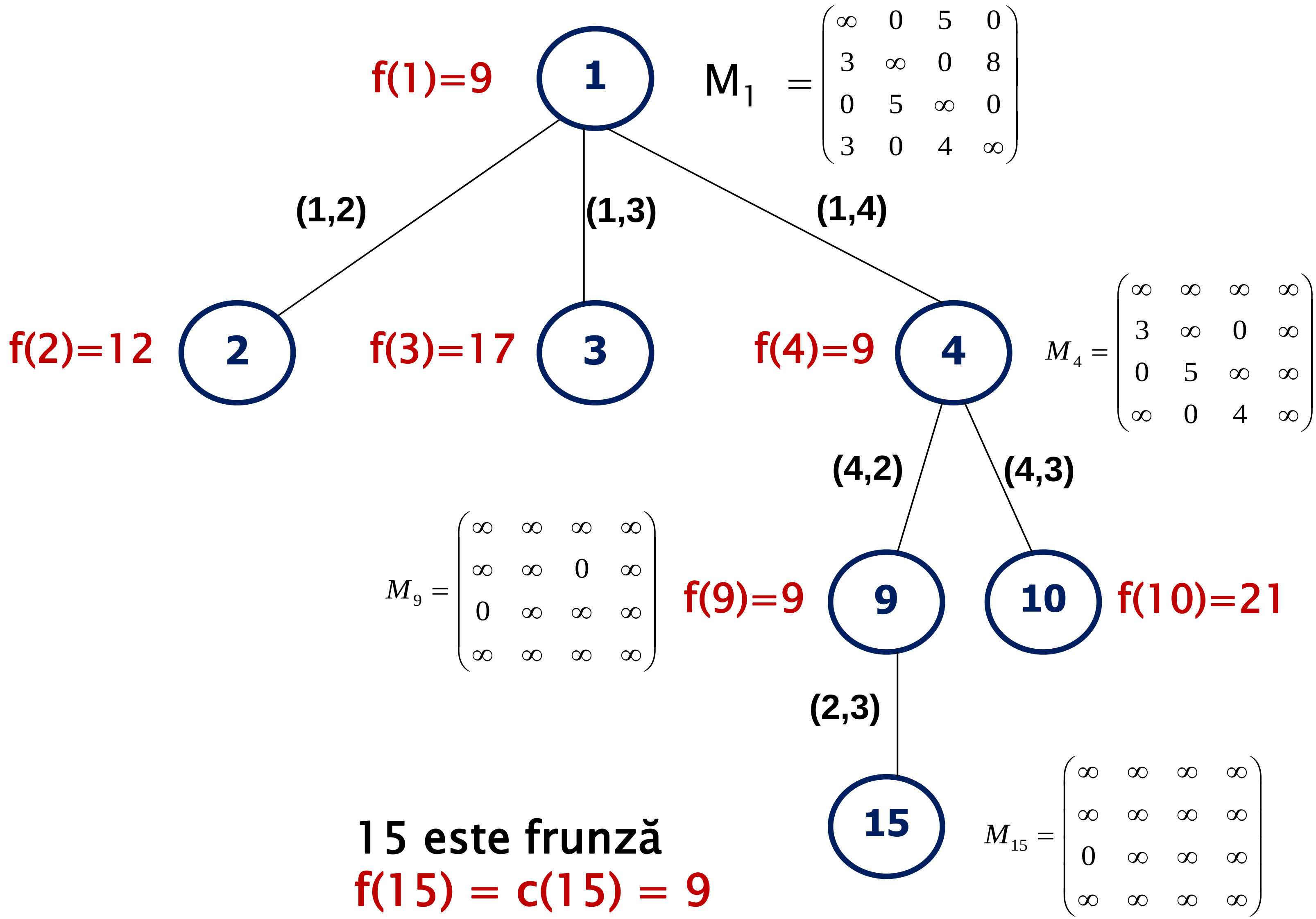


$$M_1 = \begin{pmatrix} \infty & 0 & 5 & 0 \\ 3 & \infty & 0 & 8 \\ 0 & 5 & \infty & 0 \\ 3 & 0 & 4 & \infty \end{pmatrix}$$

$$M_4 = \begin{pmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & 0 & \infty \\ 0 & 5 & \infty & \infty \\ \infty & 0 & 4 & \infty \end{pmatrix}$$

L = {2, 3, 9, 10}

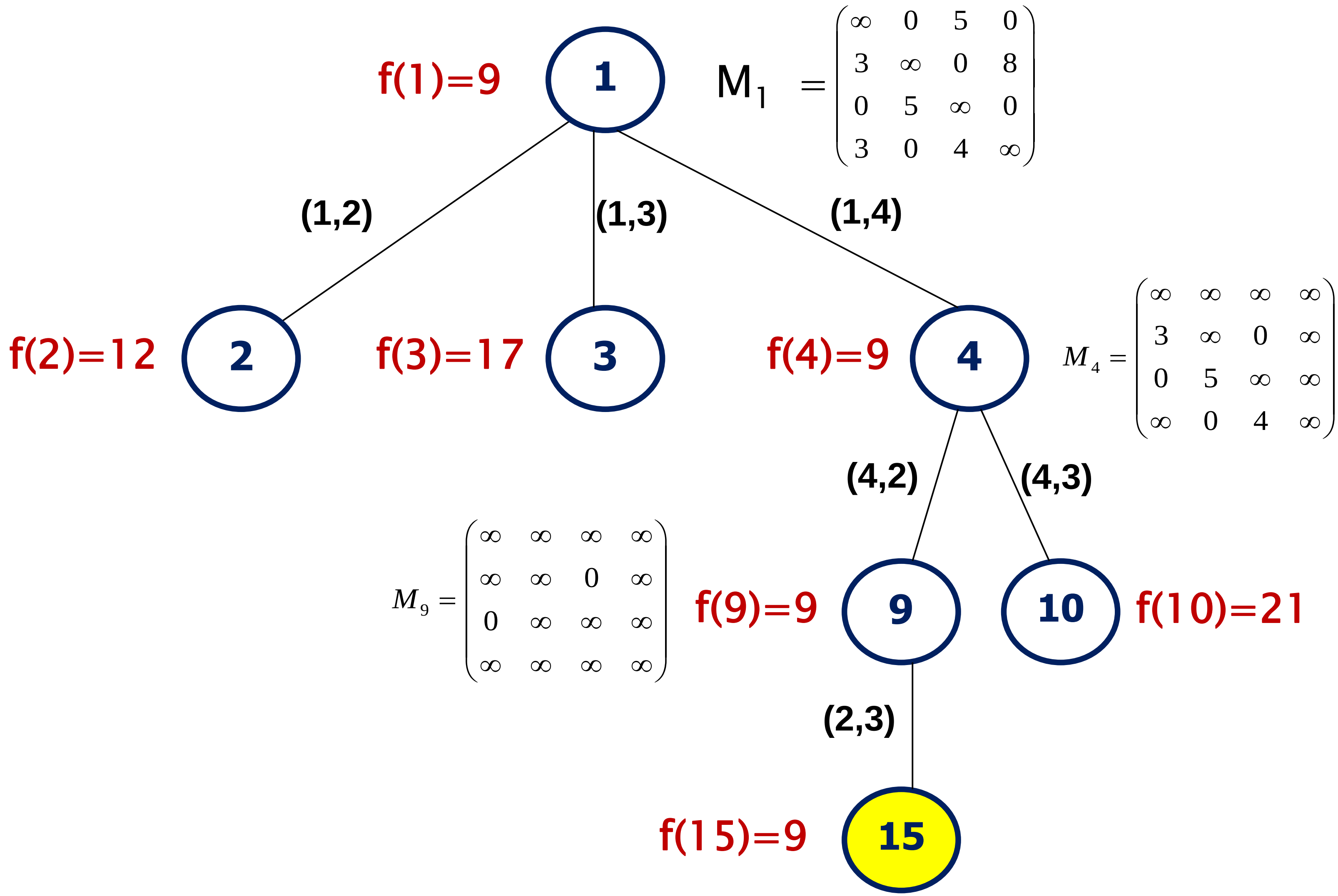
Extragem din L vârful cu f minim -> 9



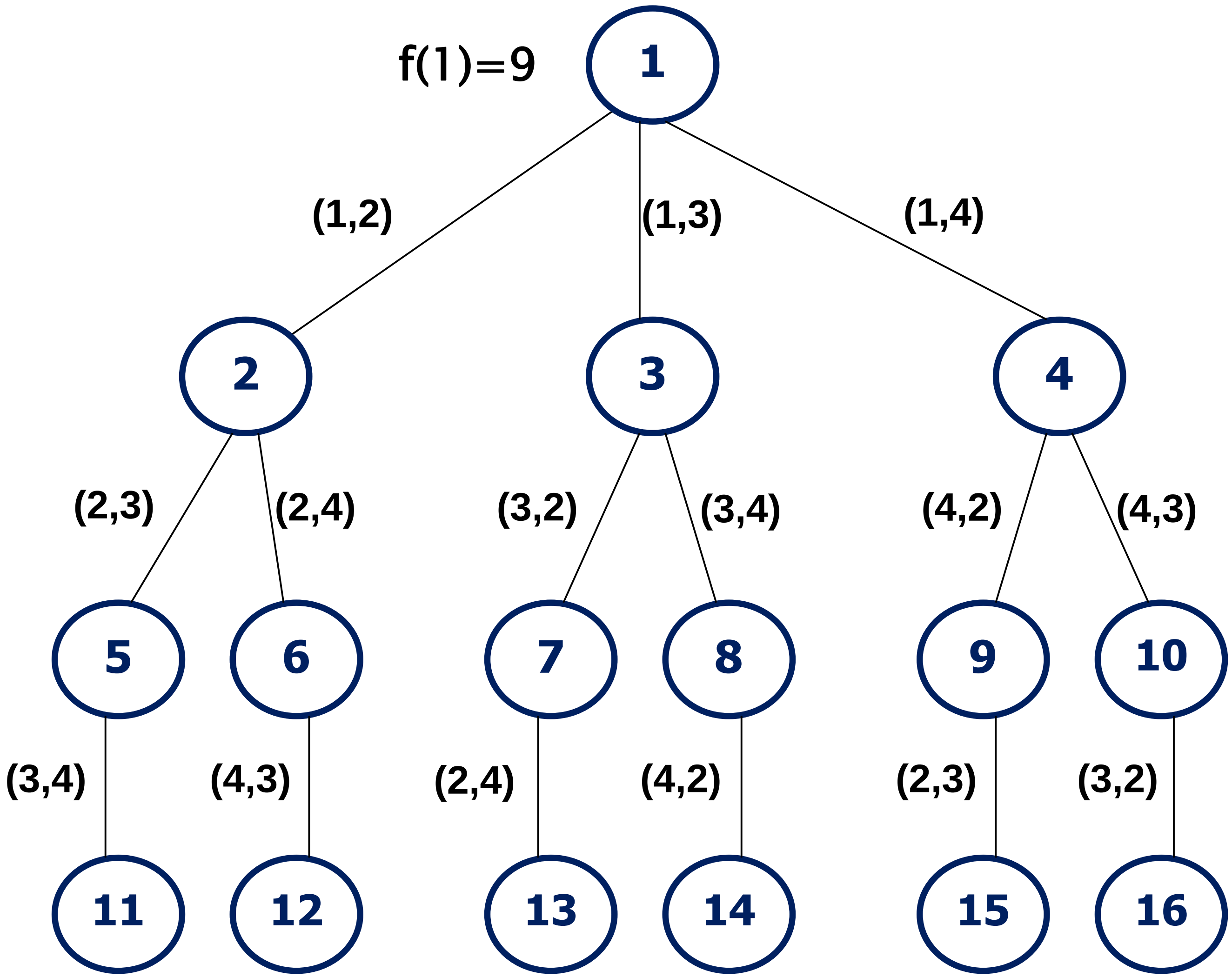
15 este frunză

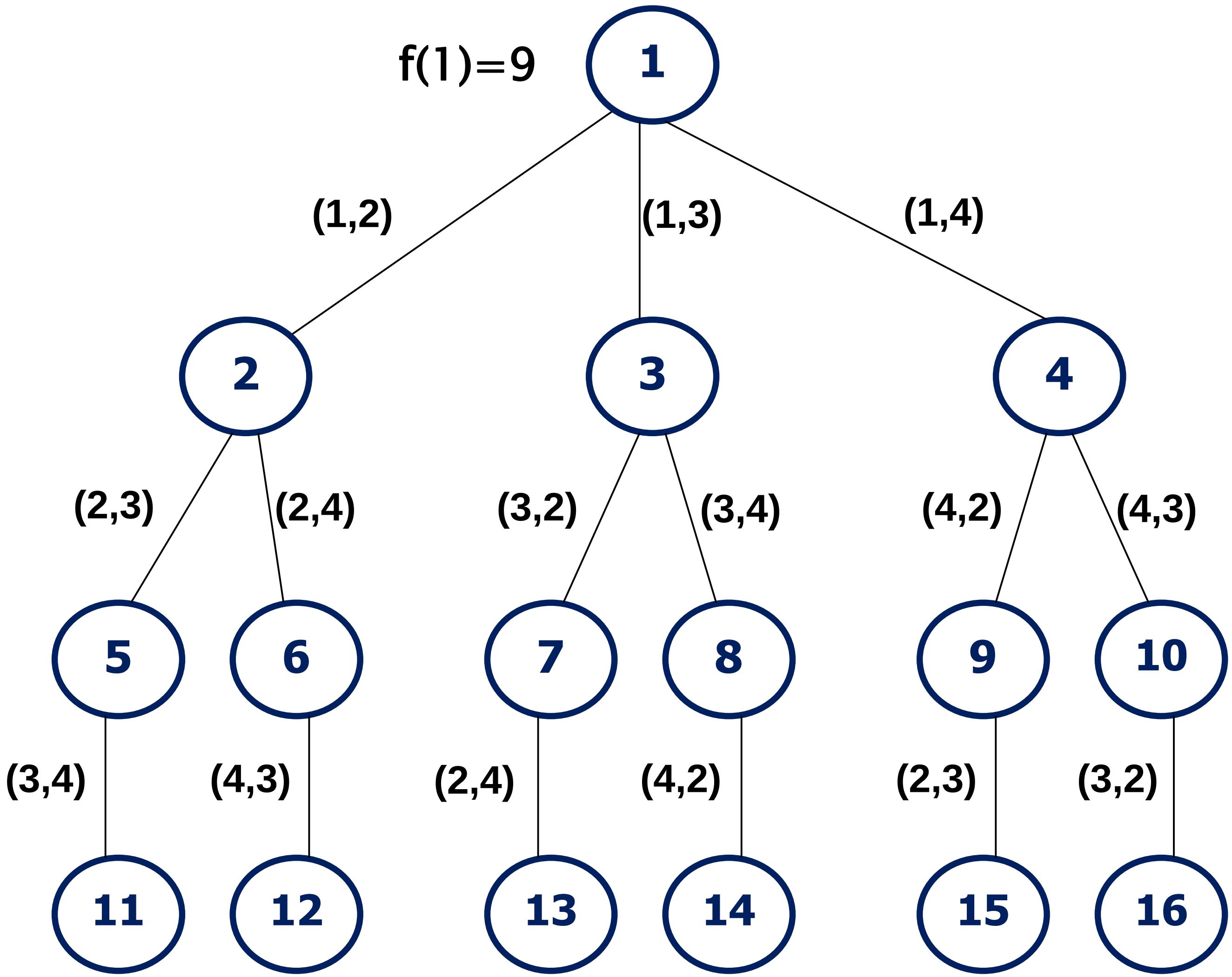
f(15) = c(15) = 9

(este chiar costul circuitului Hamiltonian, nu doar o aproximare)



- min devine 9
- se elimină din L toate vârfurile cu f mai mare decât 9
- L devine vidă -> STOP


f(1)=9
1
(1,2)
(1,3)
(1,4)
2
3
4
(2,3)
(2,4)
(3,2)
(3,4)
(4,2)
(4,3)
5
6
7
8
9
10
(3,4)
(4,3)
(2,4)
(4,2)
(2,3)
(3,2)
11
12
13
14
15
16


f(1)=9
1
(1,2)
(1,3)
(1,4)
2
3
4
(2,3)
(2,4)
(3,2)
(3,4)
(4,2)
(4,3)
5
6
7
8
9
10
(3,4)
(4,3)
(2,4)
(4,2)
(2,3)
(3,2)
11
12
13
14
15
16

Exemplu – circuit hamiltonian minim TSP

Alte euristici h :

- Pentru cazul neorientat:

  `cost(TSP)≥ cost arbore parţial de cost minim`

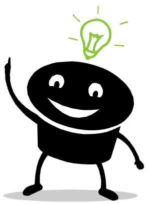
În general – **euristici h se pot obţine cu algoritmi greedy de aproximare a soluţiei optime (relaxând cerinţele).**

Exemplu – Problema rucsacului

Cum estimăm o limită superioară a profitului pentru greutatea G şi obiectele {1,…,n} dacă nu este permisă fracţionarea obiectelor?

Exemplu – Problema rucsacului

Cum estimăm o limită superioară a profitului pentru greutatea G şi obiectele {1,…,n} dacă nu este permisă fracţionarea obiectelor?



Rezolvăm problema în ipoteza că obiectele pot fi fracţionate, folosind algoritm greedy – câştigul astfel obţinut este mai mare sau egal decât cel care se poate obţine în cazul discret

Exemplu – Problema rucsacului

valoarea maximă problema discretă a rucsacului ≤

valoarea maximă problema fracționară a rucsacului corespunzătoare

Soluția problemei rucsacului – un vector $x = (x_1, \ldots, x_n)$ cu fracțiunile de obiecte

problema discretă		problema fracționară
$\max\left(\sum_{i=1}^{n} x_i c_i\right)$	$\leq$	$\max\left(\sum_{i=1}^{n} x_i c_i\right)$
$\sum_{i=1}^{n} x_i g_i \leq G$		$\sum_{i=1}^{n} x_i g_i \leq G$
$x_i \in \{0,1\}, i = \overline{1,n}$		$x_i \in [0,1], i = \overline{1,n}$
problemă de programare întreagă		problemă de programare liniară asociată (relaxarea ipotezei că variabilele sunt întregi) – rezolvabilă polinomial

Exemplu – Problema rucsacului

Rezolvând problema fracționară asociată putem estima şi o **limită superioară a câştigului (funcția euristică h)** pe parcursul algoritmului când mai avem de ales între obiectele {i+1,…,n} (variabilele x_1, x_2, ..., x_i **au deja valori fixate**)

Exemplu – Problema rucsacului

Rezolvând problema fracționară asociată putem estima şi o **limită superioară a câştigului (funcția euristică h)** pe parcursul algoritmului când mai avem de ales între obiectele {i+1,…,n} (variabilele x_1, x_2, ..., x_i **au deja valori fixate**)

De exemplu, pentru configurația parțială $x_1 = 1$, $x_2 = 0$ valoarea funcției f va fi soluția următoarei probleme fracționare a rucsacului (în care primul obiect este deja încărcat în rucsac, iar al doilea nu se ia):

$$\max\left(\sum_{i=1}^{n} x_i c_i\right)$$

$$\sum_{i=1}^{n} x_i g_i \le G$$

$$x_1 = 1, x_2 = 0$$

$$x_i \in [0,1], i = \overline{3,n}$$

Exemplu – Problema rucsacului

Posibilitate de parcurgere arbore de stări:

- **Obiectele se consideră în ordinea descrescătoare a câştigului pe unitatea de greutate**
- **La nivelul i întâi se consideră cazul în care luam obiectul i este luat**
- **Parcurgere arbore – DF**

Exemplu - Problema rucsacului

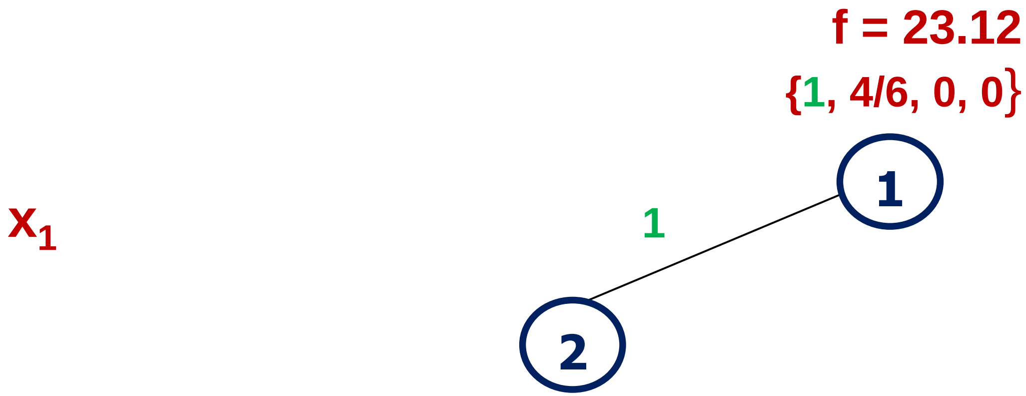
Temă - implementarea

```
g:    7    6     3   4
c:   15   12.2   6   4
r:  2.12 2.03    2   1
G = 11
```

1

f(1) = 23.12

soluţia greedy {1, 4/6, 0, 0}



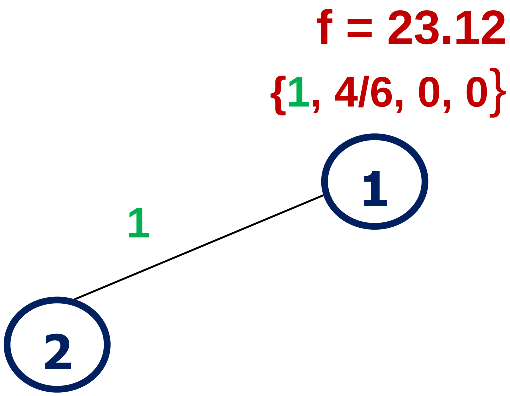
g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11

x_1



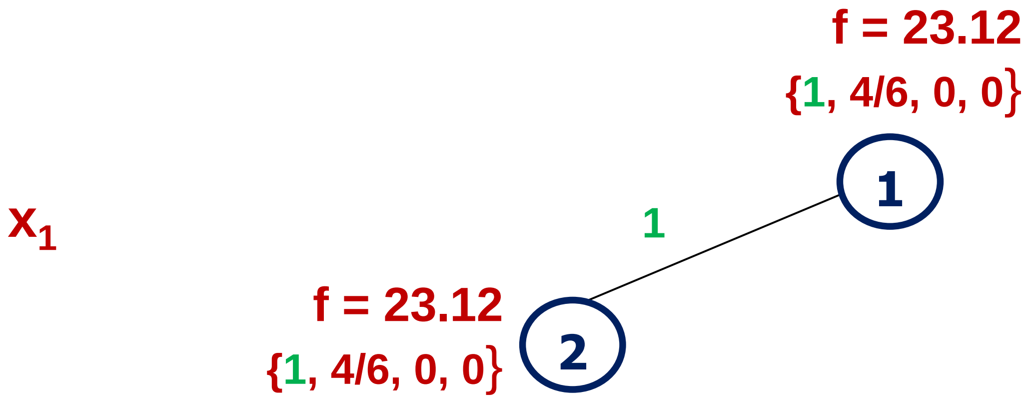
$x_1 = 1$ fixat $\Rightarrow$ rămâne greutatea $g = G - g_1 = 11 - 7 = 4$

Rucsac fracţionar pentru greutatea 4 şi obiectele {2,3,4} =>

câştig $h = 4/6 * 12.2 = 8.12$ pentru soluţia {4/6, 0, 0}

$\Rightarrow$ $f = c_1 + h = 15 + 8.12 = 23.12$

(avem f(2)=f(1))

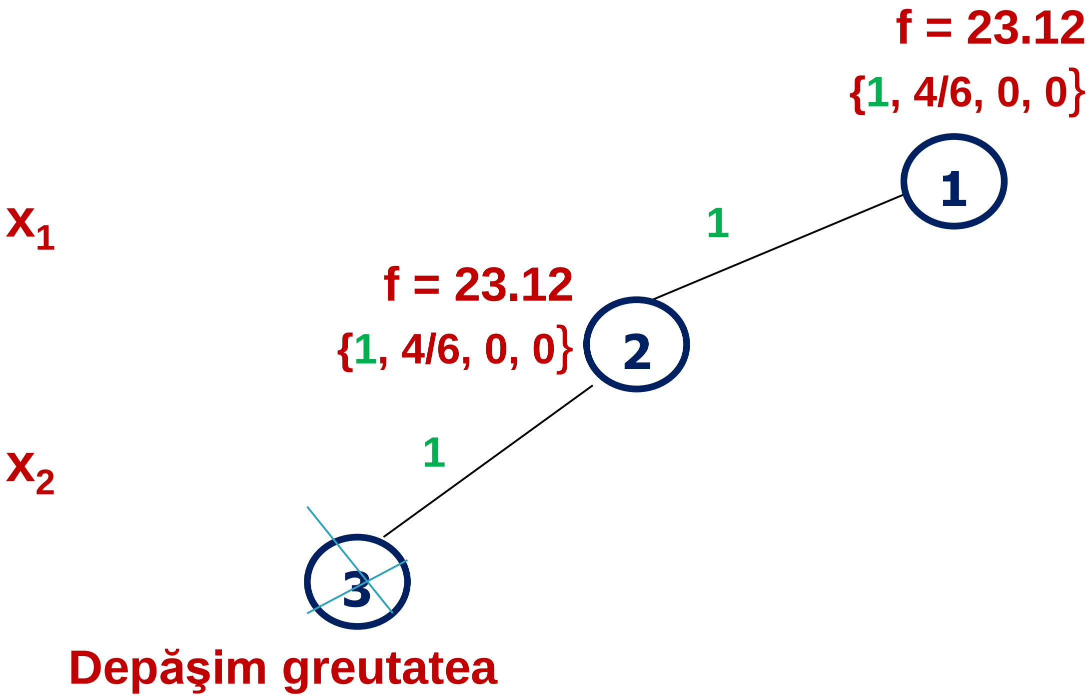


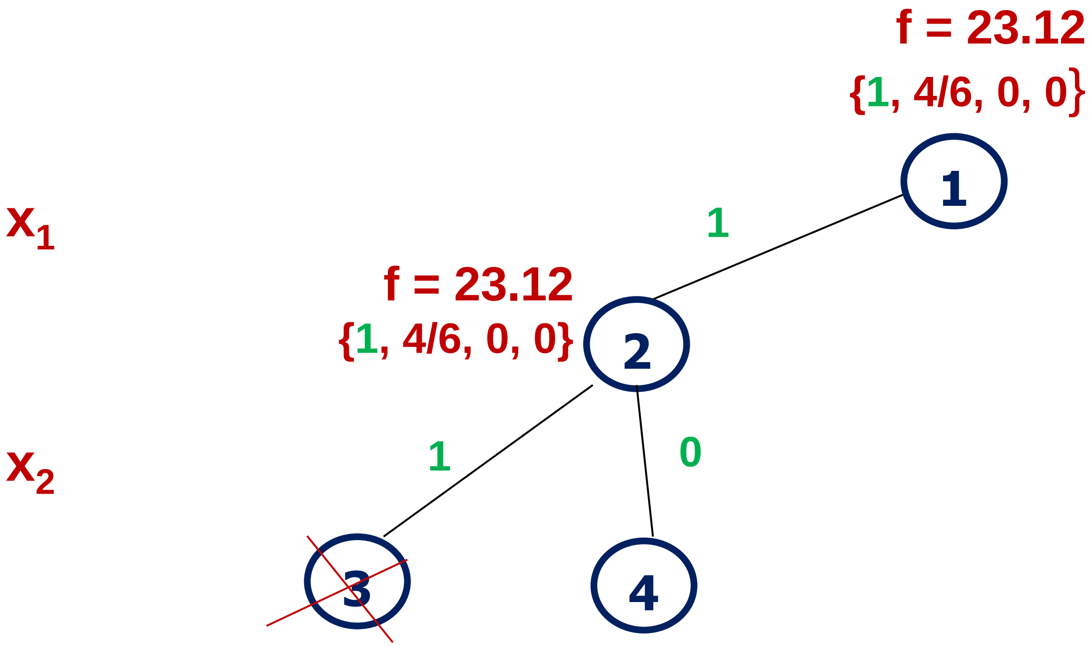
g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11

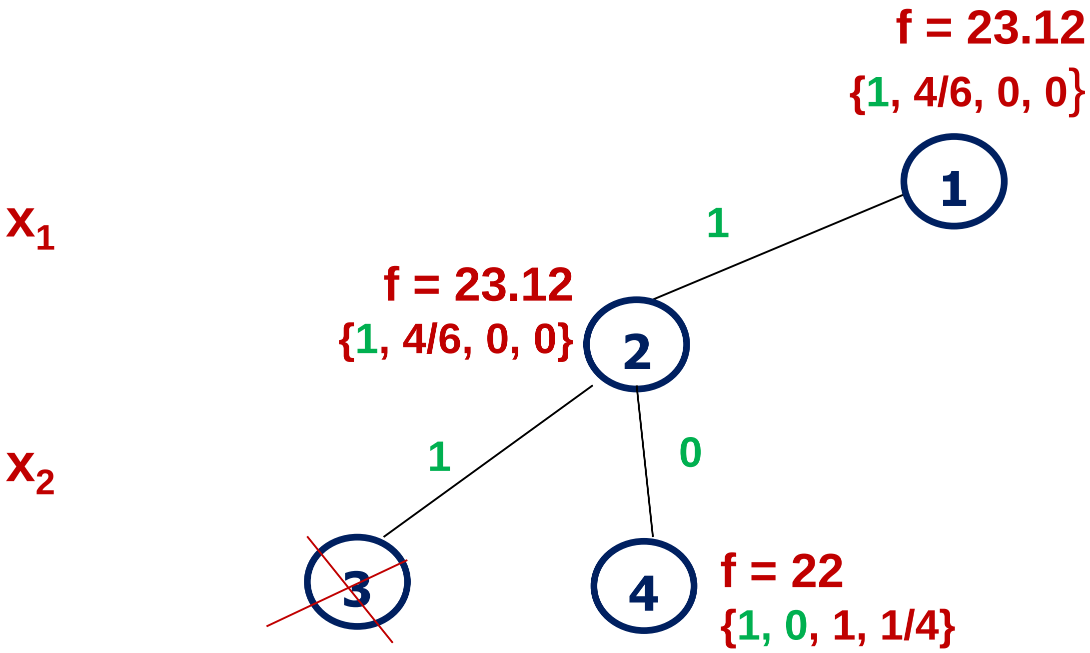


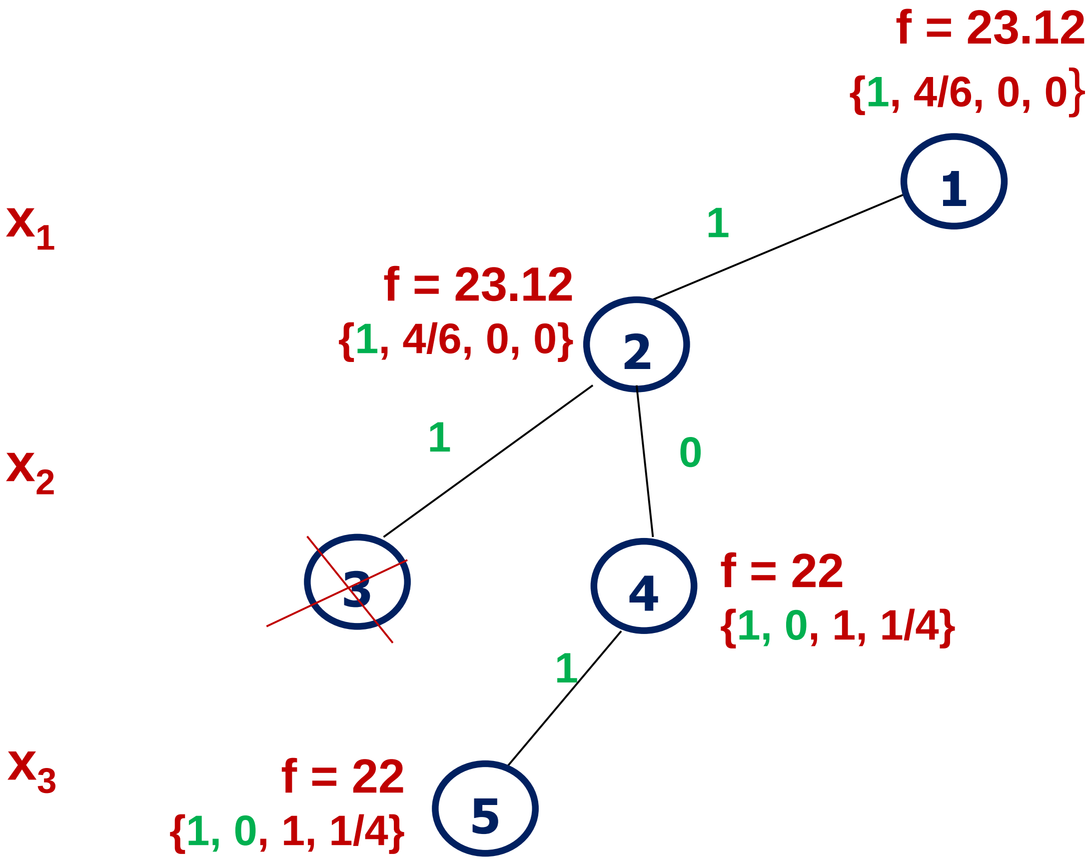


g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11

```
g:    7    6      3   4
c:   15   12.2    6   4
r:  2.12  2.03    2   1
G = 11
```



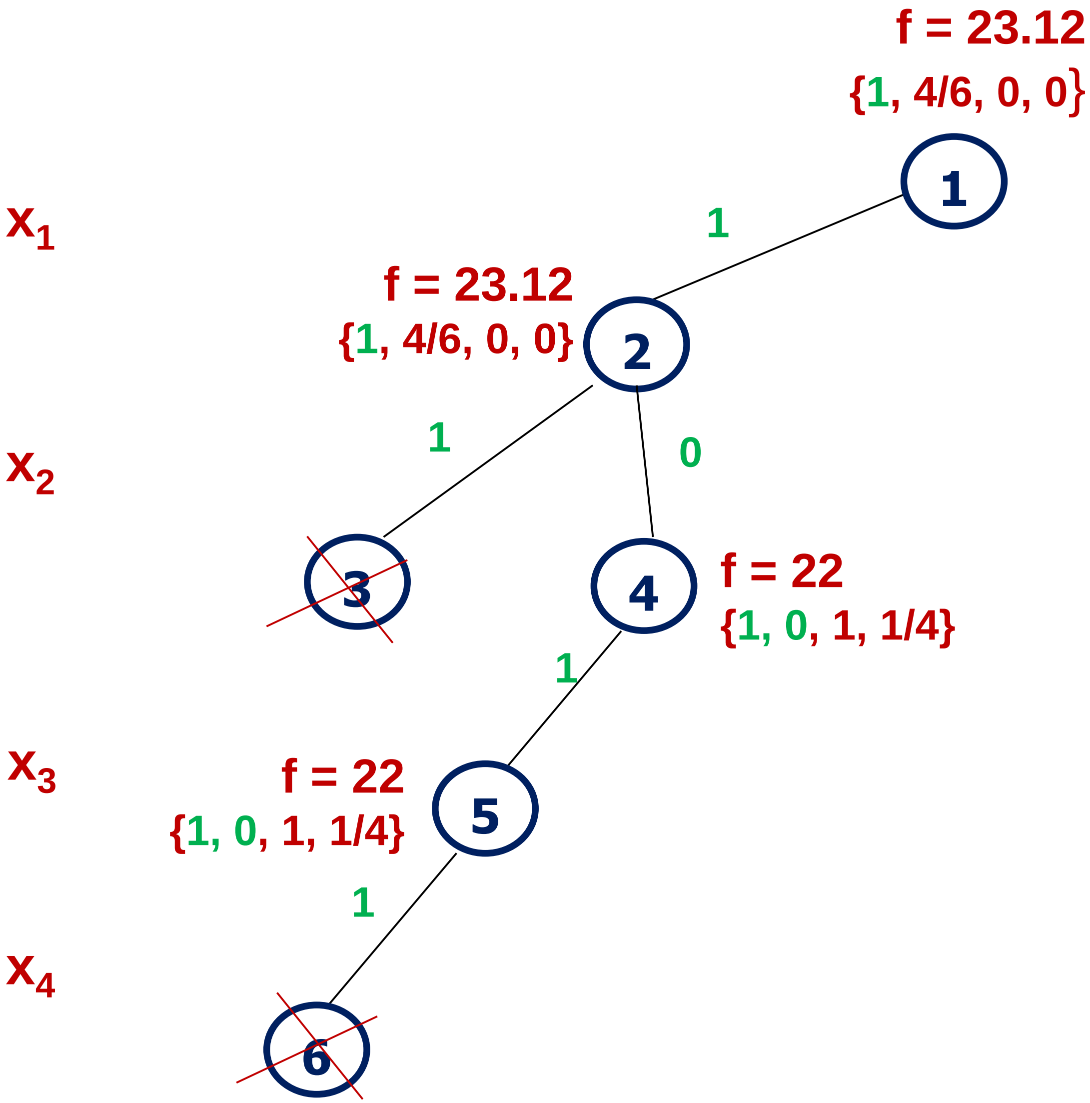


g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11

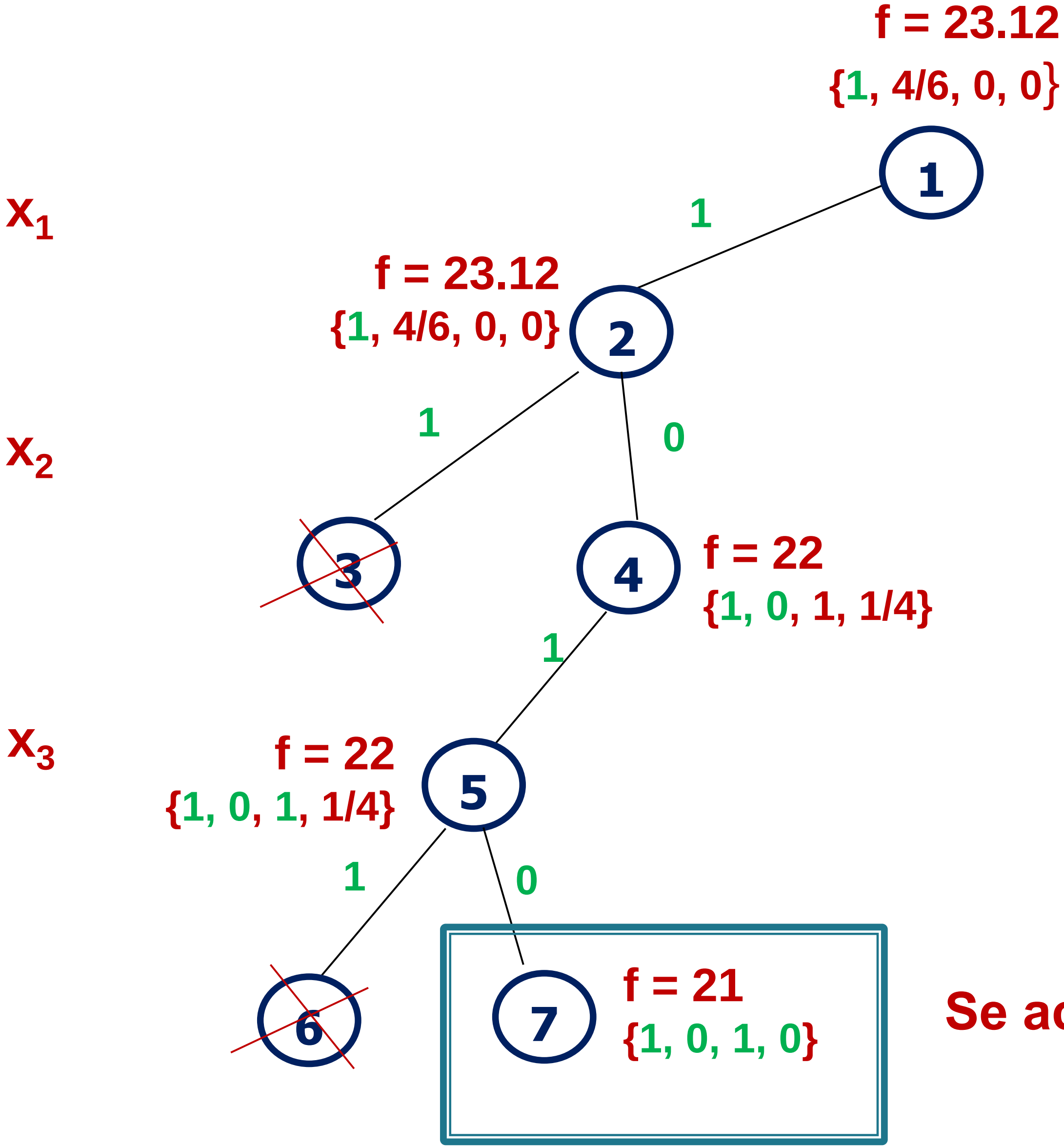
g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11



g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

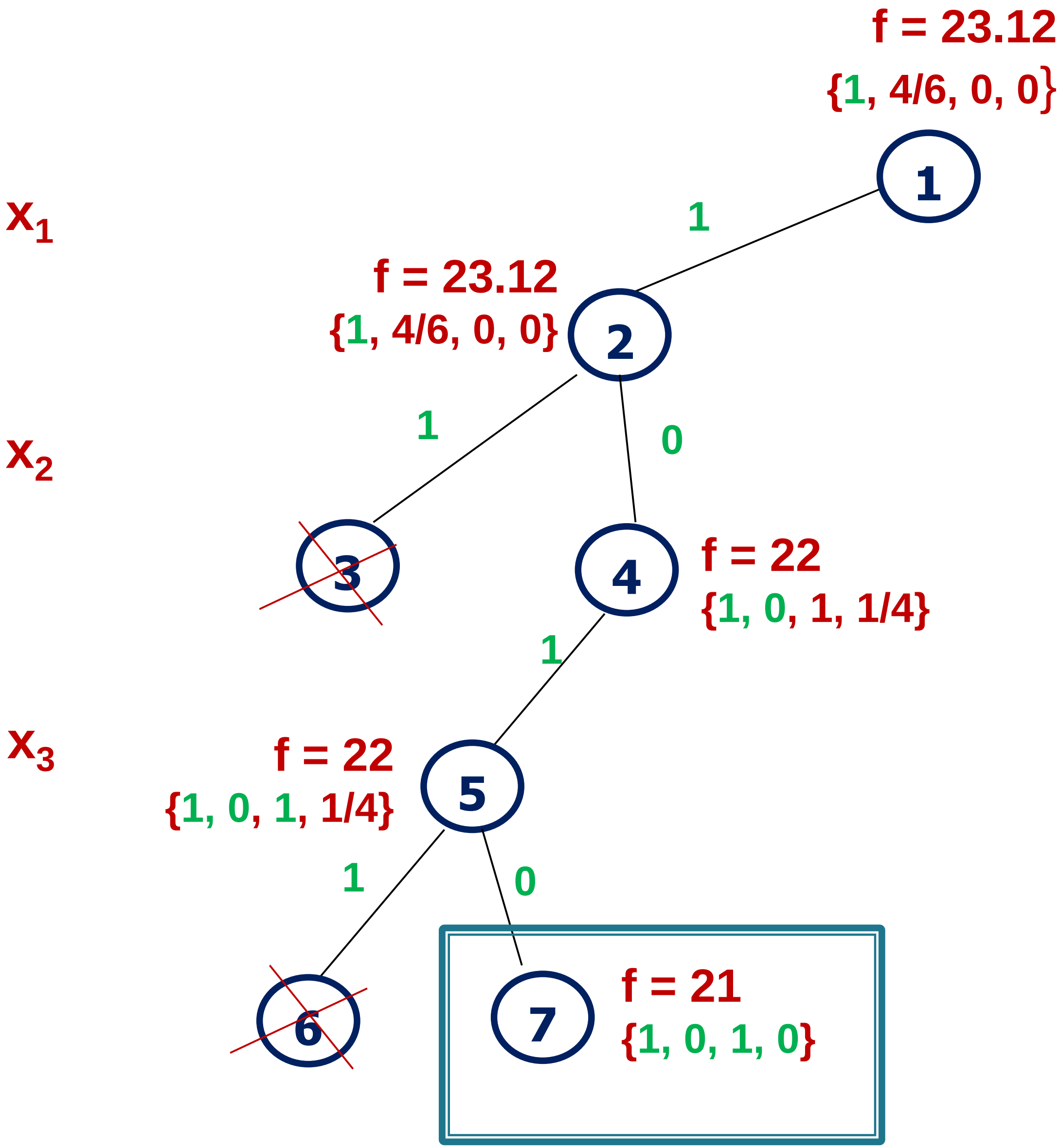
G = 11



Se actualizează max = 21

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

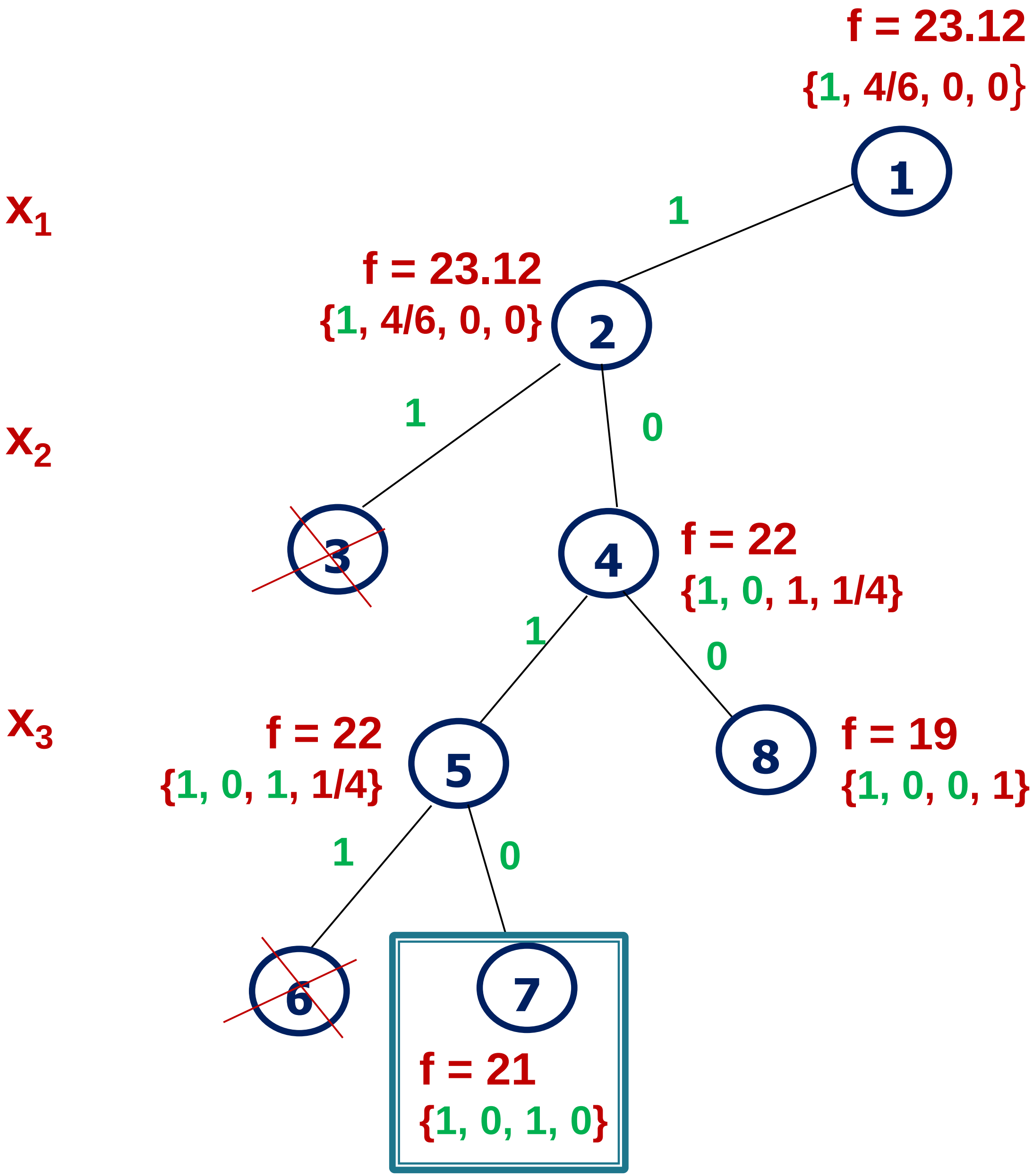
G = 11



max = 21

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

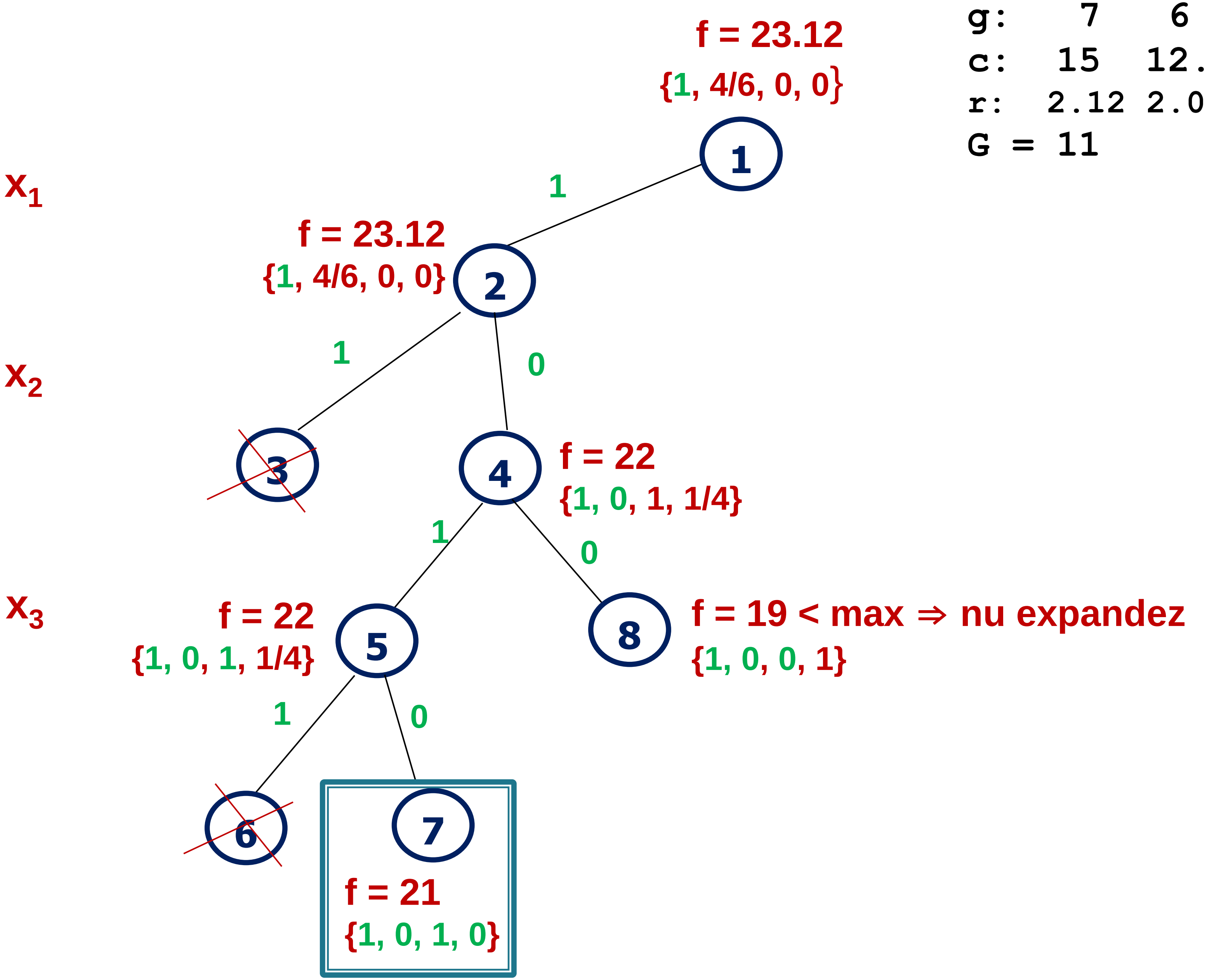
G = 11



max = 21

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

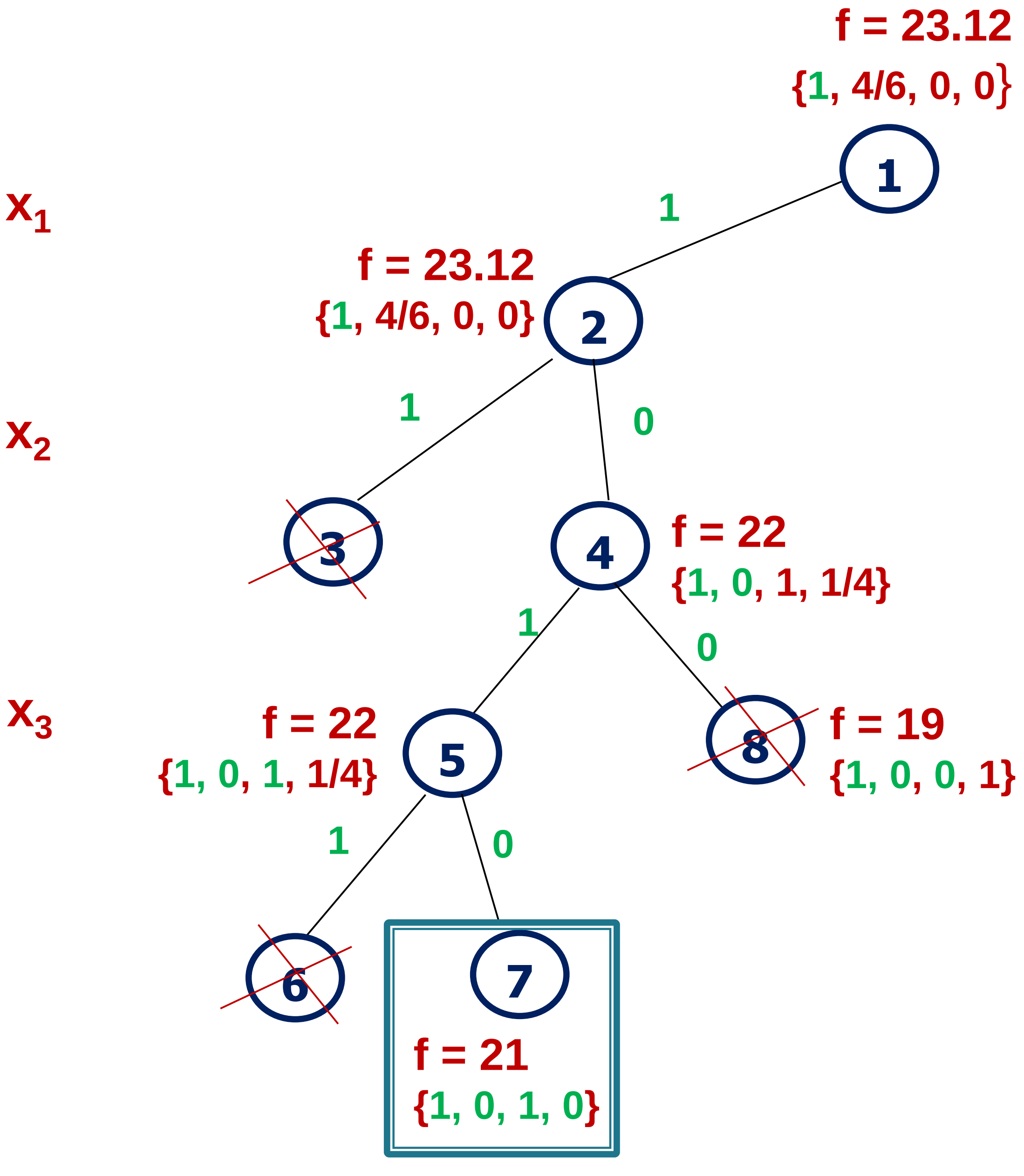
G = 11



max = 21

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

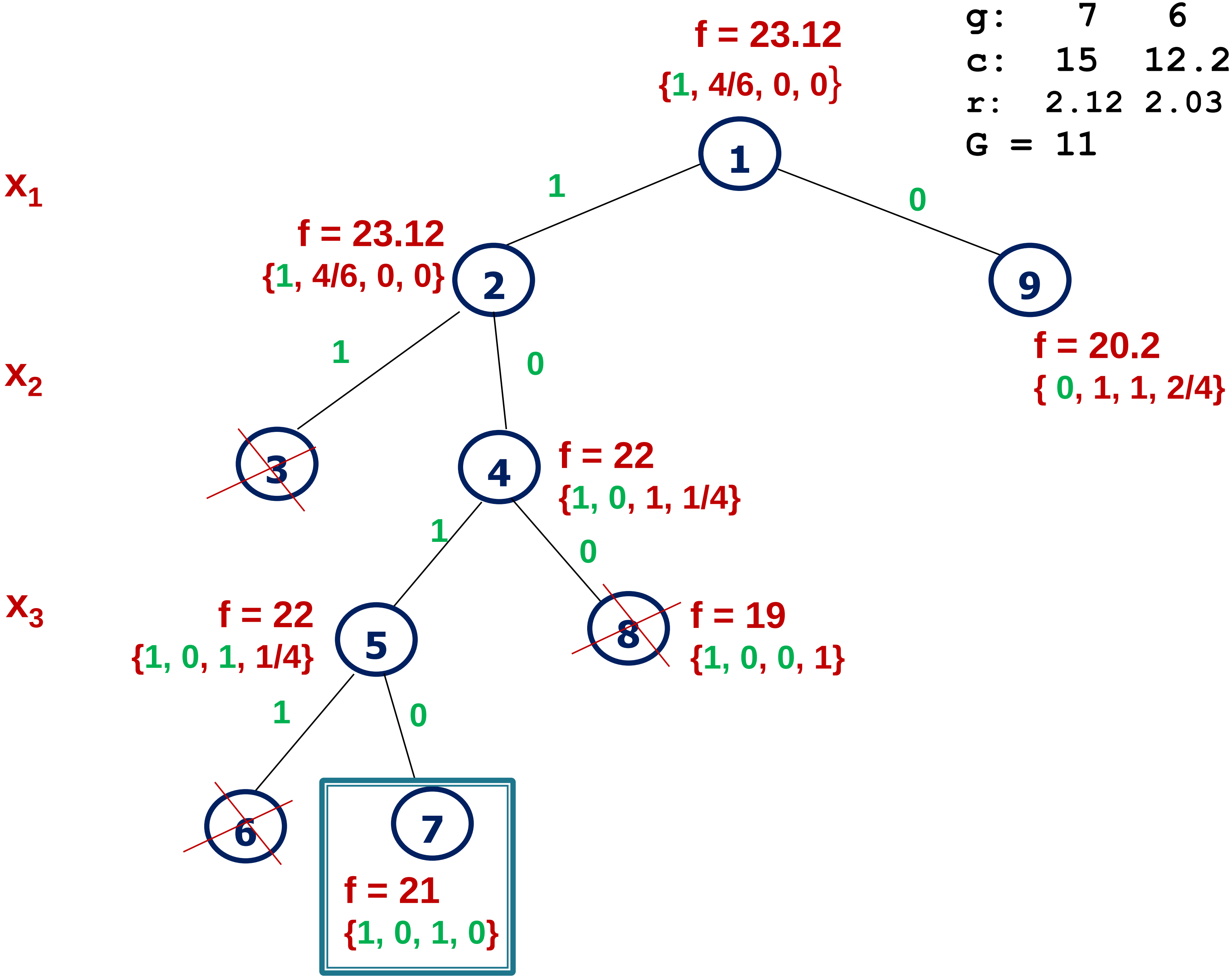
G = 11



max = 21

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

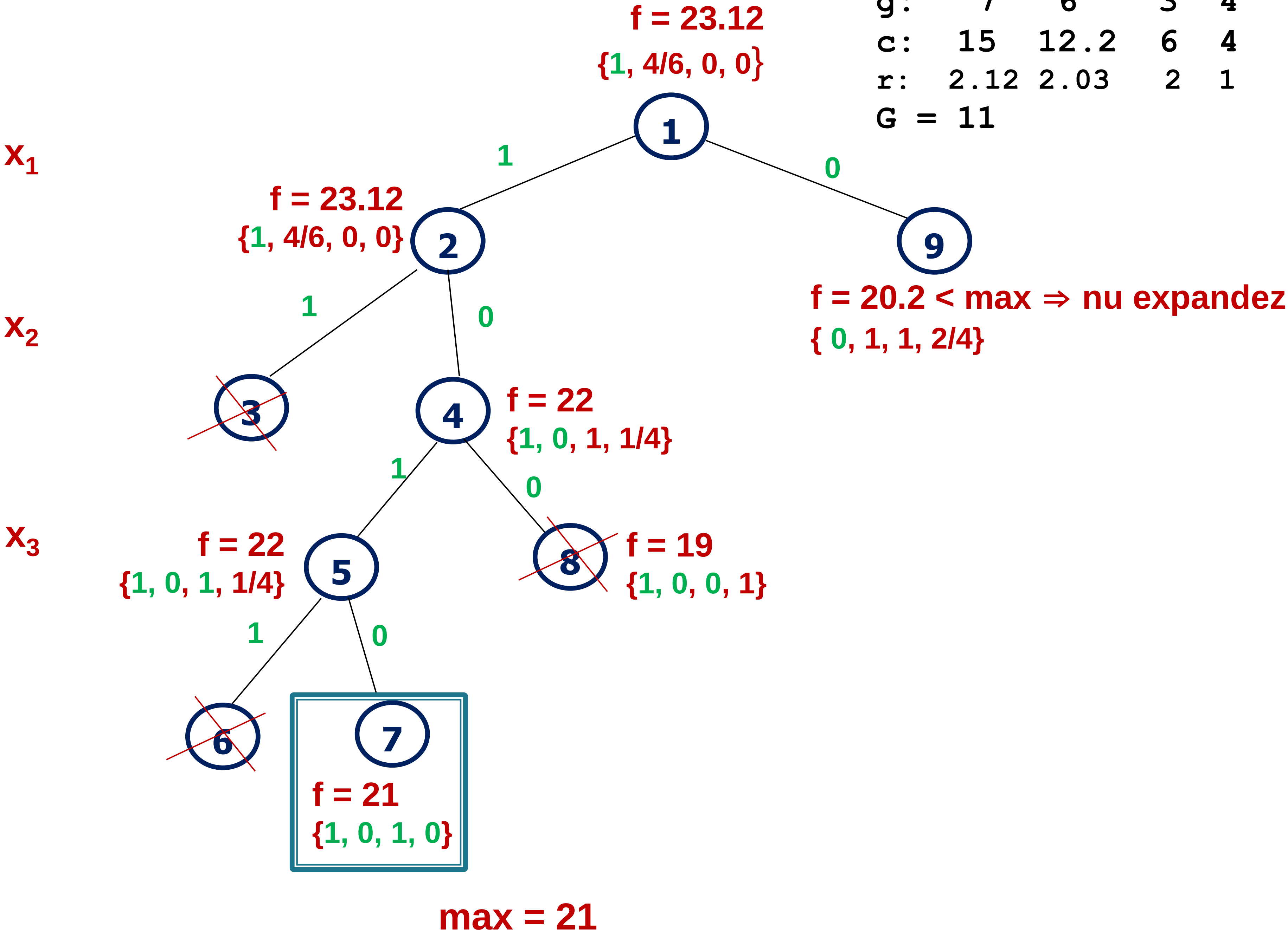
G = 11

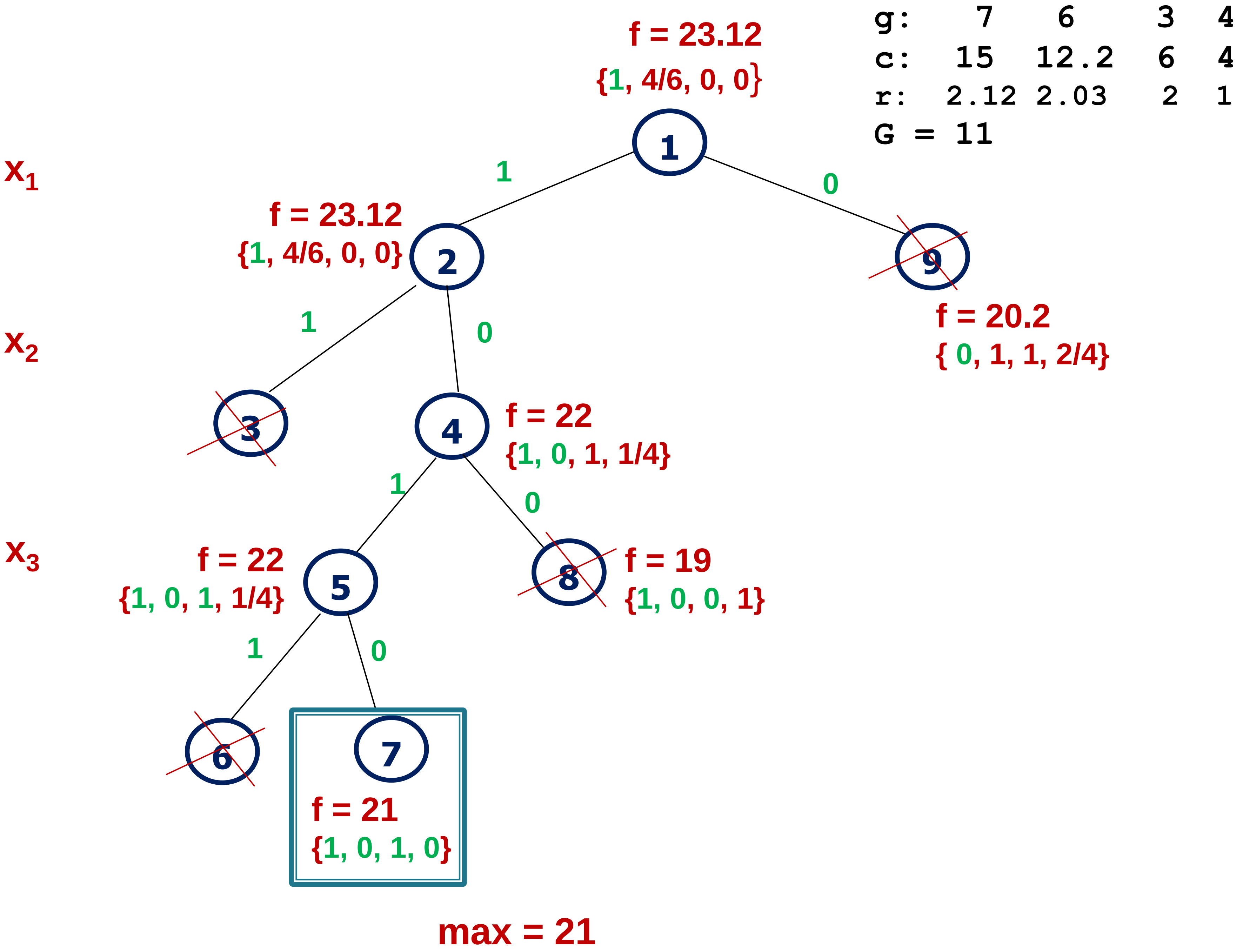


max = 21

g:	7	6	3	4
c:	15	12.2	6	4
r:	2.12	2.03	2	1

G = 11




g: 7 6 3 4
c: 15 12.2 6 4
r: 2.12 2.03 2 1
G = 11
x1
x2
x3
f = 23.12
{1, 4/6, 0, 0}
1
1
0
f = 23.12
{1, 4/6, 0, 0}
2
9
f = 20.2
{ 0, 1, 1, 2/4}
1
0
3
4
f = 22
{1, 0, 1, 1/4}
1
0
f = 22
{1, 0, 1, 1/4}
5
8
f = 19
{1, 0, 0, 1}
1
0
6
7
f = 21
{1, 0, 1, 0}
max = 21