

# Arbori Roșu Negru

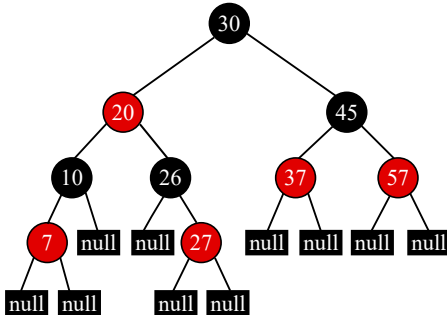
Universitatea "Transilvania" din Brașov

6 aprilie 2022

# Arbore roșu - negru

**Definiție:** Un arbore roșu-negru (ARN) este un arbore binar de căutare în care:

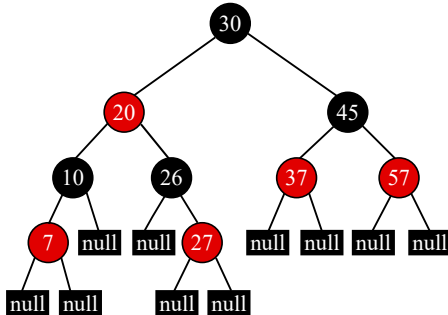
- 1 Fiecare nod este roșu sau negru - are deci un câmp suplimentar *color*



# Arbore roșu - negru

**Definiție:** Un arbore roșu-negru (ARN) este un arbore binar de căutare în care:

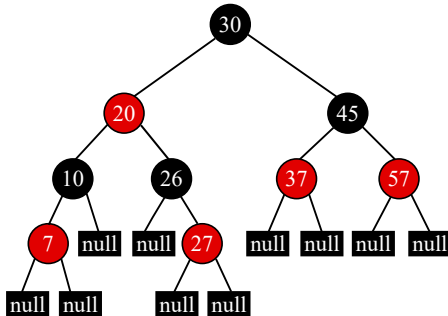
- 1 Fiecare nod este roșu sau negru - are deci un câmp suplimentar *color*
- 2 Rădăcina este neagră



# Arbore roșu - negru

**Definiție:** Un arbore roșu-negru (ARN) este un arbore binar de căutare în care:

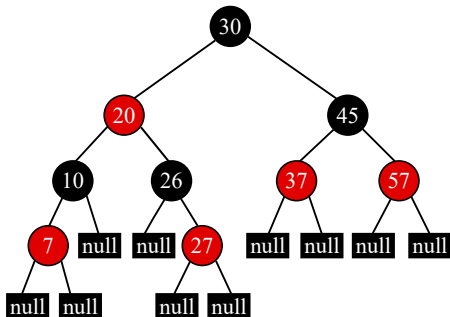
- 1 Fiecare nod este roșu sau negru - are deci un câmp suplimentar *color*
- 2 Rădăcina este neagră
- 3 Fiecare frunză este neagră și NIL



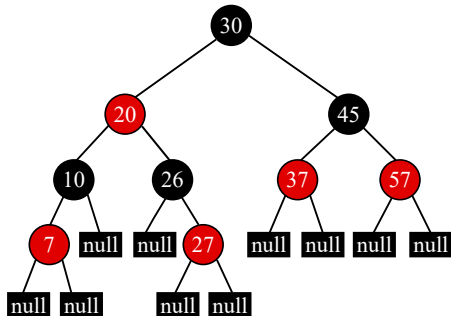
# Arbore roșu - negru

**Definiție:** Un arbore roșu-negru (ARN) este un arbore binar de căutare în care:

- 1 Fiecare nod este roșu sau negru - are deci un câmp suplimentar *color*
- 2 Rădăcina este neagră
- 3 Fiecare frunză este neagră și NIL
- 4 Dacă un nod este roșu, ambii fii sunt negri  $\Rightarrow$  părintele unui nod roșu este negru.



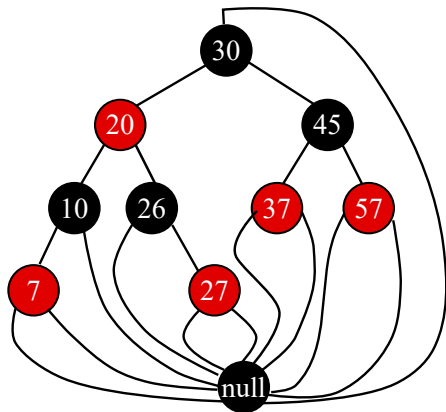
# Arbore roșu - negru



**Definiție:** Un arbore roșu-negru (ARN) este un arbore binar de căutare în care:

- 1 Fiecare nod este roșu sau negru - are deci un câmp suplimentar *color*
- 2 Rădăcina este neagră
- 3 Fiecare frunză este neagră și NIL
- 4 Dacă un nod este roșu, ambii fii sunt negri  $\Rightarrow$  părintele unui nod roșu este negru.
- 5 Pentru fiecare nod  $x$ , oricare drum de la nod la o frunză NIL se întâlnește același număr de noduri negre (inclusiv frunza NIL și exclusiv nodul de la care se pornește). Acest număr se numește înălțimea neagră a subarborului de rădăcină  $x$ . Notăm înălțimea neagră cu *bh* - *black height*.

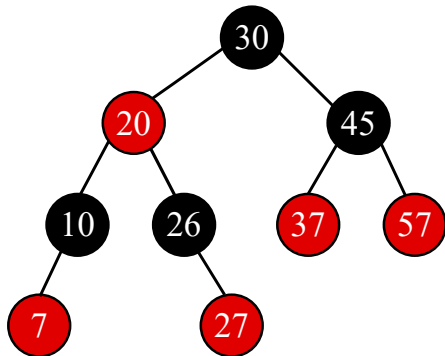
# Arbore roșu - negru



## Observații

- Definirea unui nod NIL pentru fiecare frunză presupune un consum inutil de memorie. Se poate considera în locul acestor frunze un singur nod santinelă T.nil.

# Arbore roșu - negru

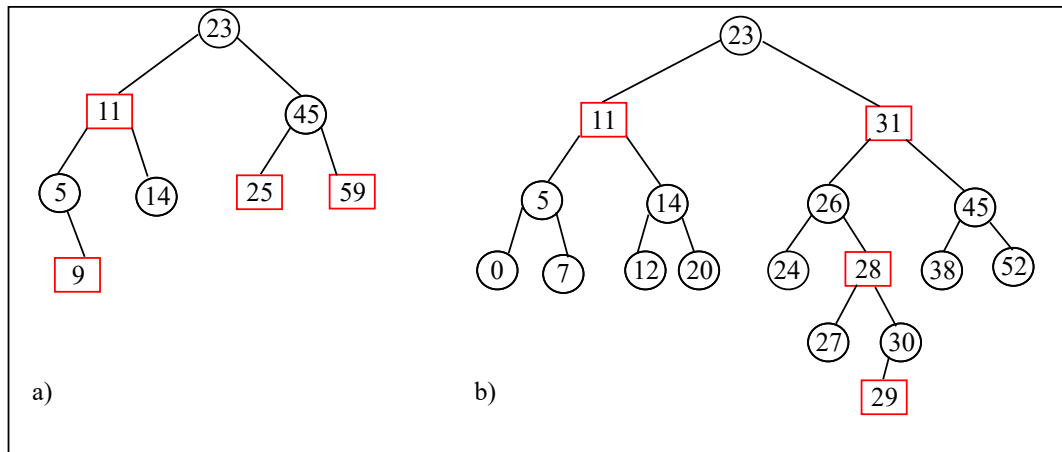


## Observații

- Definirea unui nod NIL pentru fiecare frunză presupune un consum inutil de memorie. Se poate considera în locul acestor frunze un singur nod santinelă T.nil.
- Pentru simplitate în continuare vom ignora în desene nodurile NIL.

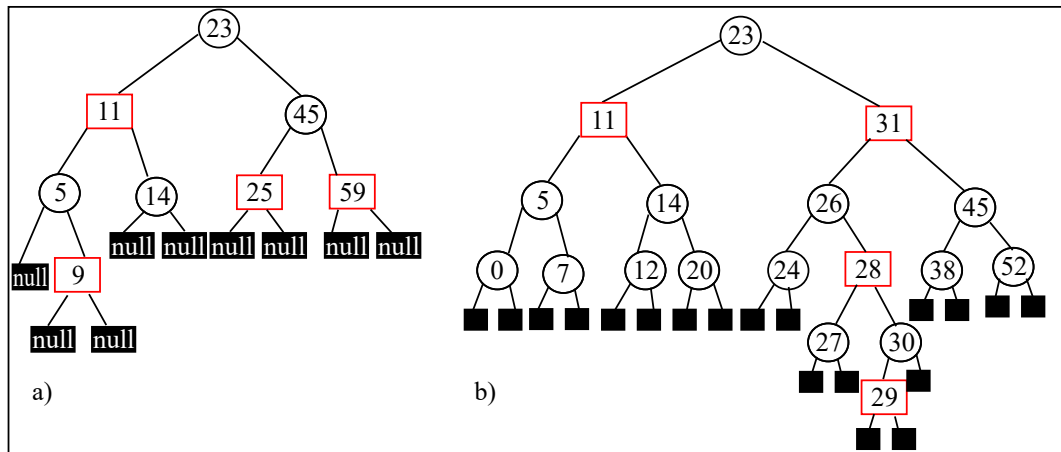


# Înălțimea neagră



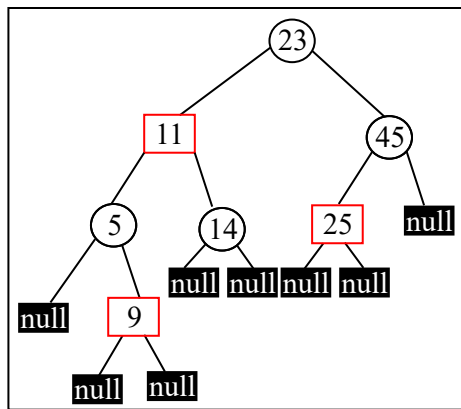
Care este înălțimea neagră a fiecărui arbore? a)  $bh = ?$  b)  $bh = ?$

# Înălțimea neagră



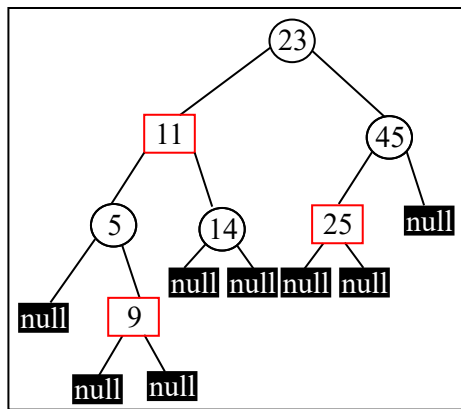
Care este înălțimea neagră a fiecărui arbore? a)  $bh = 2$  b)  $bh = 3$

# Înălțimea unui ARN



- Într-un ARN niciun drum de la rădăcină la o frunză nu poate fi mai lung decât dublul unui alt drum la altă frunză.

# Înălțimea unui ARN



- Într-un ARN niciun drum de la rădăcină la o frunză nu poate fi mai lung decât dublul unui alt drum la altă frunză.
- Înălțimea maximă a unui ARN este  $2 \log_2(n + 1)$ .

Se demonstrează prin inducție că orice subarbore de rădăcină  $x$  conține cel puțin  $2^{bh(x)} - 1$  noduri interne. Notând cu  $h$  înălțimea arborelui și cu  $r$  rădăcina, din proprietatea 4 se obține  $bh(r) \geq h/2$ . Dar  $n \geq 2^{bh(r)} - 1 \geq 2^{h/2} - 1$  de unde rezultă  $h \leq 2 \log_2(n + 1)$

# ARN - Inserție

## Etape

- 1 Inserția noului nod ca la orice arbore binar de căutare
- 2 Refacerea proprietăților roșu - negru.

**Observație:** Nodul care se inserează are culoarea roșie

# ARN - Inserție

## Proprietățile ARN: prin inserție

- Proprietățile 1, 3 și 5 se păstrează, datorită faptului că se inserează un nod roșu, care are ca fii două frunze NIL.

# ARN - Inserție

## Proprietățile ARN: prin inserție

- Proprietățile 1, 3 și 5 se păstrează, datorită faptului că se inserează un nod roșu, care are ca fii două frunze NIL.
- Proprietatea 2 poate fi contrazisă (rădăcina are culoarea neagră). Pentru rezolvarea acestei situații este suficientă colorarea rădăcinii cu negru.

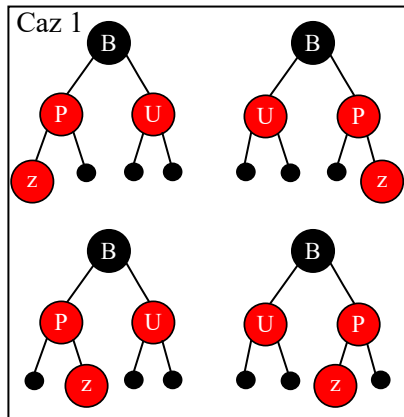
# ARN - Inserție

## Proprietățile ARN: prin inserție

- Proprietățile 1, 3 și 5 se păstrează, datorită faptului că se inserează un nod roșu, care are ca fii două frunze NIL.
- Proprietatea 2 poate fi contrazisă (rădăcina are culoarea neagră). Pentru rezolvarea acestei situații este suficientă colorarea rădăcinii cu negru.
- Proprietatea 4 poate fi contrazisă, dacă părintele de care s-a legat noul nod are culoarea roșie. În acest caz trebuie refăcută proprietatea. Există în această situație 3 cazuri care vor fi discutate în continuare.



# ARN - Inserție - Refacerea proprietăților RN

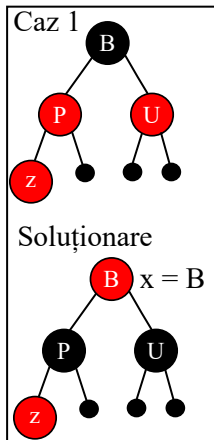


Proprietățile RN trebuie refăcute dacă nodul inserat  $z$  roșu se leagă de un părinte  $P$  roșu.

**Observație:** s- a realizat inserție într-un ARN valid  
 $\Rightarrow P$  nu e rădăcină  $\Rightarrow P$  are un părinte negru  $B$ .

**Cazul I:** Unchiul  $U$  al lui  $x$  (fratele lui  $P$ ) este roșu.

## ARN - Inserție - Refacerea proprietăților RN



Proprietățile RN trebuie refăcute dacă nodul inserat  $z$  roșu se leagă de un părinte  $P$  roșu.

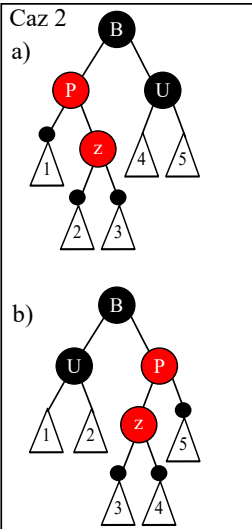
**Observație:** s- a realizat inserție într-un ARN valid  
 $\Rightarrow P$  nu e rădăcină  $\Rightarrow P$  are un părinte negru  $B$ .

**Cazul I:** Unchiul  $U$  al lui  $x$  (fratele lui  $P$ ) este roșu.

**Soluție:**

- Se colorează cu roșu  $B$ , se colorează cu negru  $P$ ,  $U$
- Se reia procesul de refacere a proprietăților de la  $x = B$  (se urcă în arbore)

# ARN - Inserție - Refacerea proprietăților RN



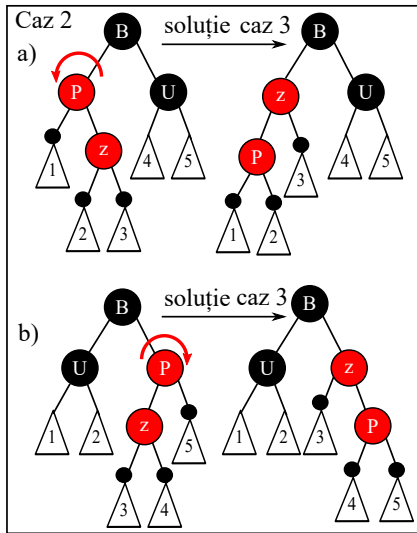
Proprietățile RN trebuie refăcute dacă nodul inserat  $z$  roșu se leagă de un părinte  $P$  roșu.

**Observație:** s- a realizat inserție într-un ARN valid  
 $\Rightarrow P$  nu e rădăcină  $\Rightarrow P$  are un părinte negru  $B$ .

**Cazul II:** Unchiul  $U$  al lui  $x$  este negru și

- a)  $z$  la dreapta lui  $P$ , iar  $P$  la stânga lui  $B$
- b)  $z$  la stânga lui  $P$ , iar  $P$  la dreapta lui  $B$

# ARN - Inserție - Refacerea proprietăților RN



Proprietățile RN trebuie refăcute dacă nodul inserat  $z$  roșu se leagă de un părinte  $P$  roșu.

**Observație:** s- a realizat inserție într-un ARN valid  
 $\Rightarrow P$  nu e rădăcină  $\Rightarrow P$  are un părinte negru  $B$ .

**Cazul II:** Unchiul  $U$  al lui  $x$  este negru și

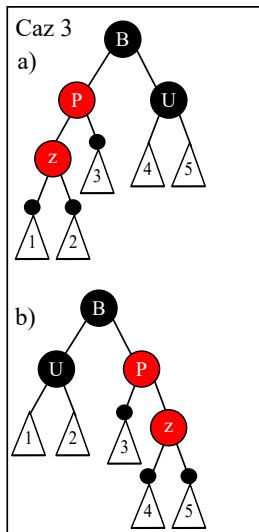
- $z$  la dreapta lui  $P$ , iar  $P$  la stânga lui  $B$
- $z$  la stânga lui  $P$ , iar  $P$  la dreapta lui  $B$

**Soluție:**

- în cazul a) - rotație la stânga după  $P$ .
- în cazul b) - rotație la dreapta după  $P$

Nu se soluționează complet, ci se trece practic în cazul 3, în care refacerea începe de la nodul care prin rotație a coborât, deci de la  $z = P$ .

# ARN - Inserție - Refacerea proprietăților RN



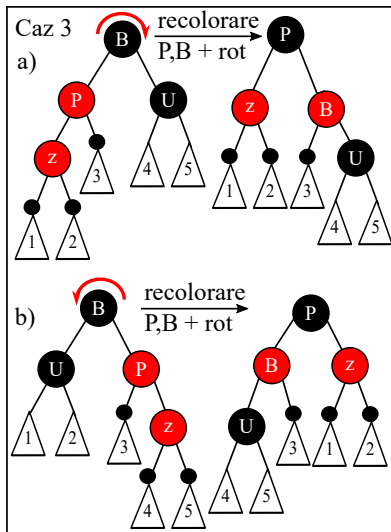
Proprietățile RN trebuie refăcute dacă nodul inserat  $z$  roșu se leagă de un părinte  $P$  roșu.

**Observație:** s- a realizat inserție într-un ARN valid  
 $\Rightarrow P$  nu e rădăcină  $\Rightarrow P$  are un părinte negru  $B$ .

**Cazul III:** Unchiul  $U$  al lui  $x$  este negru și

- a)  $z$  la stânga lui  $P$ , iar  $P$  la stânga lui  $B$
- b)  $z$  la dreapta lui  $P$ , iar  $P$  la dreapta lui  $B$

## ARN - Inserție - Refacerea proprietăților RN



Proprietățile RN trebuie refăcute dacă nodul inserat  $z$  roșu se leagă de un părinte  $P$  roșu.

**Observație:** s- a realizat inserție într-un ARN valid  
 $\Rightarrow P$  nu e rădăcină  $\Rightarrow P$  are un părinte negru  $B$ .

**Cazul III:** Unchiul  $U$  al lui  $x$  este negru și

- $z$  la stânga lui  $P$ , iar  $P$  la stânga lui  $B$
- $z$  la dreapta lui  $P$ , iar  $P$  la dreapta lui  $B$

**Soluție:**

- Colorare  $B$  cu roșu, colorare  $P$  cu negru.
- apoi
  - pentru a) Rotație la dreapta în jurul lui  $B$
  - pentru b) Rotație la stânga în jurul lui  $B$ .

# ARN - Inserție - Refacerea proprietăților RN

---

**Algoritm 1:** ARN-Insert-Repara( $T, z$ )

---

**cat\_timp**  $z.p.color = \text{rosu}$  **executa**

**daca**  $z.p = z.p.p.st$  **atunci**

$U \leftarrow z.p.p.dr$

\_\_\_\_\_ caz 1

**daca**  $U.color = \text{rosu}$  **atunci**

$z.p.color \leftarrow \text{negru}$

$U.color \leftarrow \text{negru}$

$z.p.p.color \leftarrow \text{rosu}$

$z \leftarrow z.p.p$

**sfarsit\_daca**

**altfel**

\_\_\_\_\_ caz 2

**daca**  $z \leftarrow z.p.dr$  **atunci**

$z \leftarrow z.p$

$ROT\_ST(T, z)$

**sfarsit\_daca**

# ARN - Inserție - Refacerea proprietăților RN

caz 3

$z.p.color \leftarrow \text{negru}$

$z.p.p.color \leftarrow \text{rosu}$

$\text{ROT\_DR}(T, z.p.p)$

**altfel (daca z.p e pe dreapta lui z.p.p)**

//similar dar simetric pentru nodul z

//aflat la stânga bunicului

$T.rad.color \leftarrow \text{negru}$

RETURN



# Complexitatea algoritmului de inserție

Inserția presupune:

- Algoritmul de inserarea ca la arborii binari de căutare obișnuiți: complexitate  $O(h)$ ,  $h \leq 2 \log_2(n + 1) \Rightarrow$  complexitatea este  $O(\log_2 n)$ .
- Algoritmul de refacere al proprietăților RN: pornește de jos înspre rădăcină pe o ramură  $\Rightarrow$  complexitatea este  $O(\log_2 n)$ .

Rezultă complexitatea totală:  $O(\log_2 n)$ .

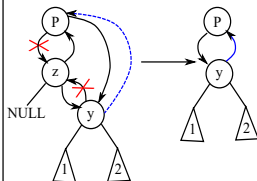
# Reprezentarea unui ARN - cod C++

```
struct ARN{  
    struct nod{  
        int info;  
        char color;  
        nod* p, *st, *dr;  
        nod(int key = 0)  
        {  
            info = key;  
            color = 'r';  
            p = st = dr = nullptr;  
        }  
    };  
};
```

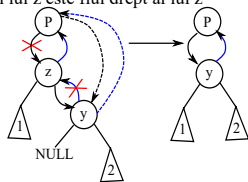
```
nod* rad;  
nod* nil;  
ARN()  
{  
    nil = new nod;  
    nod->color = 'n';  
    nod->p = nil;  
    nod->st = nod->dr = nil;  
    rad = nil;  
}  
//restul functiilor.  
//In program NU mai exista NULL.  
// Se inlocuieste cu nil  
};
```

# Algoritmul pentru ștergerea unui nod - recapitulare

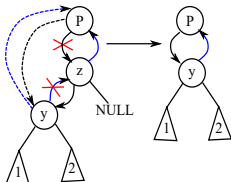
**Cazul 1:** z nu are fiu stâng



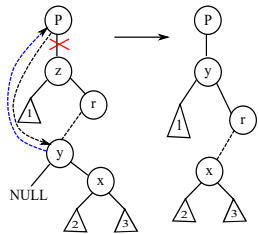
**Cazul 3.1:** z are 2 fii, succesorul y al lui z este fiul drept al lui z



**Cazul 2:** z nu are fiu drept



**Cazul 3.2:** z are 2 fii, succesorul y al lui z nu este fiul al lui z



## Algoritm 2: TRANSPLANT( $T, u, v$ )

daca  $u.p = NIL$  atunci

sfarsit\_daca

altfel

    daca  $u = u.p.st$  atunci

        |  $u.p.st = v$

    sfarsit\_daca

    altfel

        |  $u.p.dr = v$

    sfarsit\_daca

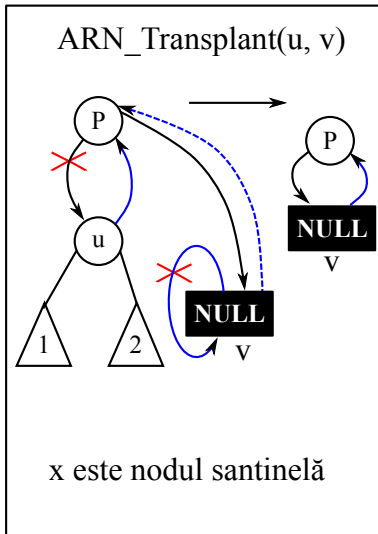
sfarsit\_daca

daca  $v \neq Nil$  atunci

    |  $v.p = u.p$

sfarsit\_daca

## Algoritmul pentru ștergerea unui nod



DAR: la ARN trebuie tratat cazul nodului santinelă!

**Algorithm 3:** ARN\_TRANSPLANT( $T, u, v$ )

**daca  $u.p = T.NIL$  atunci**

**sfarsit\_daca****altfel**

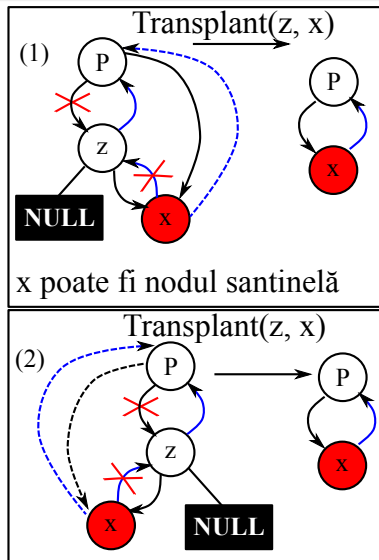
**daca  $u = u.p.st$  atunci**

$$u.p.st = v$$
**sfarsit\_daca**

**altfel**

$$u.p.dr = v$$
**sfarsit\_daca****sfarsit\_daca**
$$v.p = u.p$$

# Algoritmul pentru ștergerea unui nod



$\text{ARN\_DELETE}(T, z)$

$\text{color} = z.\text{color}$

daca  $z.\text{st} = T.\text{NIL}$  atunci

$x = z.\text{dr}$

$\text{ARN\_TRANSPLANT}(T, z, x)$

altfel

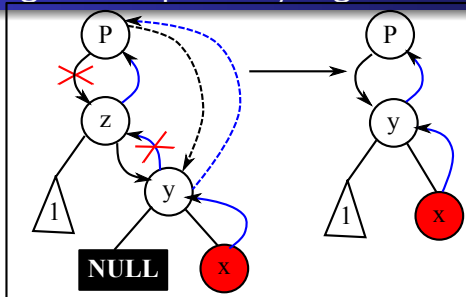
daca  $z.\text{dr} = T.\text{NIL}$  atunci

$x = z.\text{st}$

$\text{ARN\_TRANSPLANT}(T, z, x)$

altfel

# Algoritmul pentru ștergerea unui nod



[...] altfel

$y = \text{SUCCESOR}(T, z)$

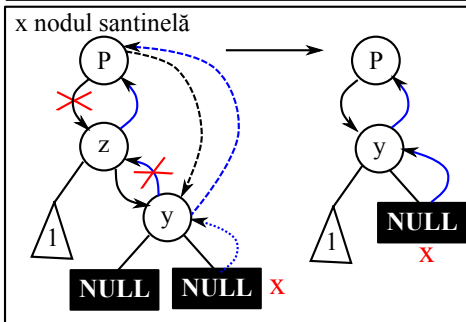
$color = y.color$

$x = y.dr$

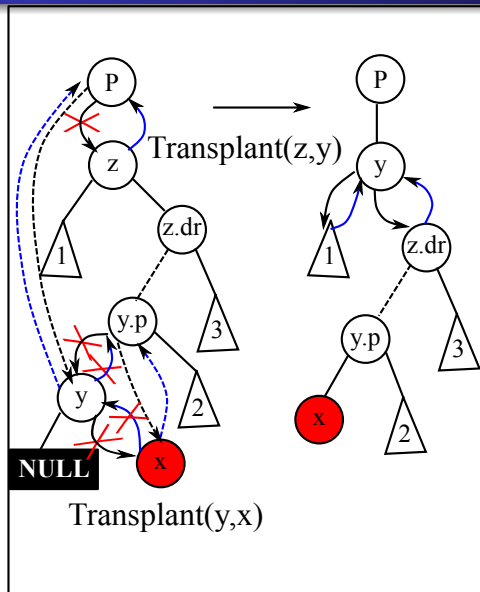
daca  $y.p = z$  atunci

$x.p = y$

altfel



# Algoritmul pentru ștergerea unui nod



[...] altfel

RB\_TRANSPLANT( $T, y, x$ )

$y.dr = z.dr$

$z.dr.p = y$

sfarsit daca

RB\_TRANSPLANT( $T, z, y$ )

$y.st = z.st$

$z.st.p = y$

$y.color = z.color$

sfarsit daca

sfarsit daca

daca  $color = negru$  atunci

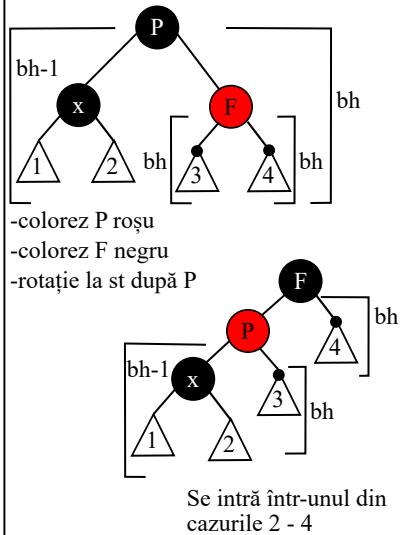
RB\_DELETE\_REPARA( $T, x$ )

sfarsit daca

RETURN

# Algoritmul pentru refacerea proprietăților RN

Caz 1:



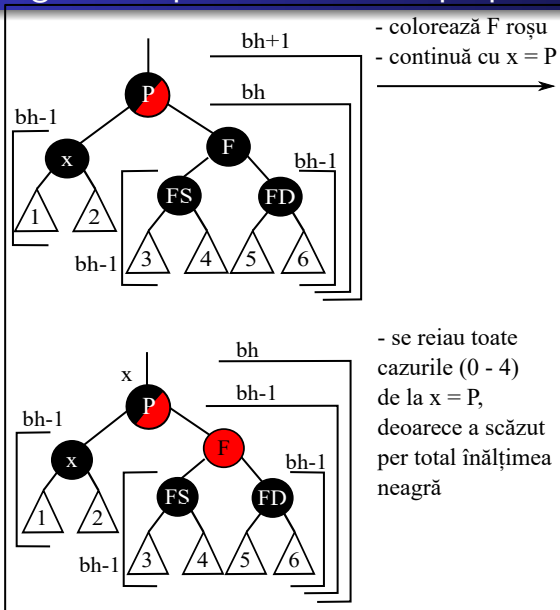
**Caz 0:** dacă  $x.color = \text{roșu}$  atunci colorez  $x$  cu negru și STOP.

Acest caz e rezolvat la sfârșitul algoritmului!

```
RB_DELETE_REPAIR( $T, x$ )
    cat timp  $x \neq T.rad$  și  $x.color = \text{negru}$ 
        dacă  $x = x.p.st$  atunci
             $F = x.p.dr$ 
            // caz 1: fratele e roșu
        dacă  $F.color = \text{roșu}$ 
             $F.color = \text{negru}$ 
             $x.p.color = \text{roșu}$ 
            ROT_ST( $T, x.p$ )
             $F = x.p.dr$ 
    sfarsit dacă
```



# Algoritmul pentru refacerea proprietăților RN



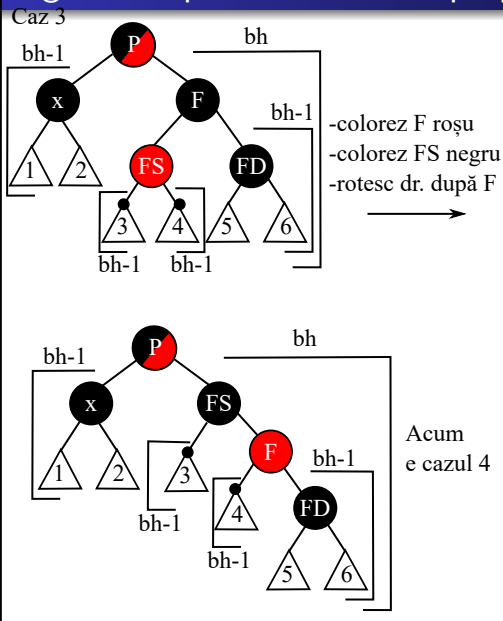
// caz 2: fratele si  
// cei 2 fii ai sai sunt negri  
[...]daca (  $F.st.color = negru$  si  
 $F.dr.color = negru$ ) atunci

$F.color = rosu$

$x = x.p$

altfel

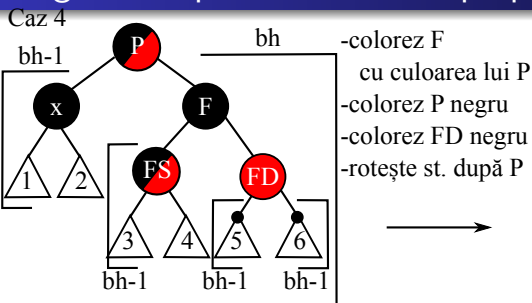
## Algoritmul pentru refacerea proprietăților RN



```
// caz 3: fratele F negru
// F.st = roșu și F.dr = negru
[...] altfel
```

```
dacă F.dr.color = negru atunci
    F.st.color = negru
    F.color = roșu
    ROT_DR(T, F)
    F = x.p.dr
sfarsit dacă
```

# Algoritmul pentru refacerea proprietăților RN



// caz 4: fratele F negru

// F.dr = rosu

[...]

$F.color = x.p.color$

$x.p.color = negru$

$F.dr.color = negru$

$ROT\_ST(T, x.p)$

$x = T.rad$

sfarsit daca

altfel //daca x pe dreapta parintelui

//se reia algoritmul simetric,

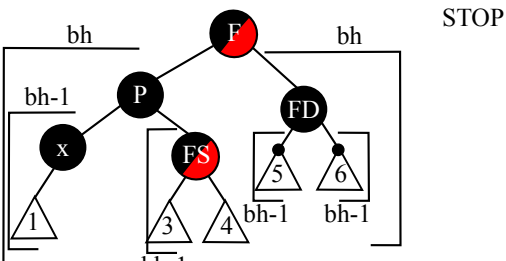
// inlocuind peste tot dreapta cu

// stanga si invers



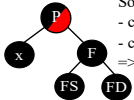
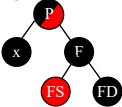
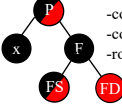
sfarsit cat timp

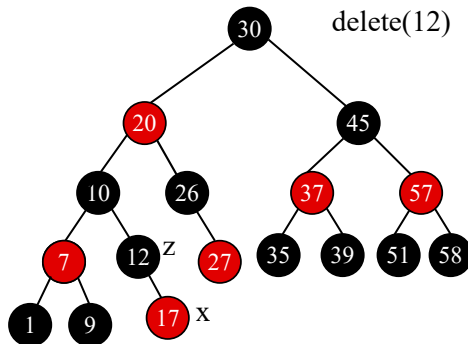
$x.color = negru$  //rezolva caz 0

RETURN



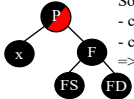
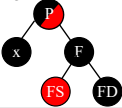
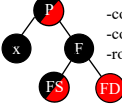


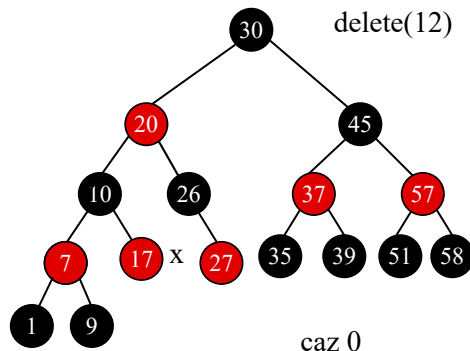
# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P





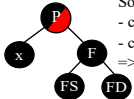
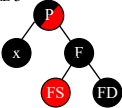
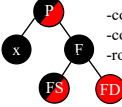
# ARN ștergere - Exemple

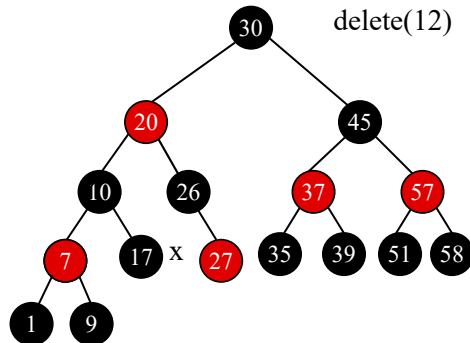
Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P





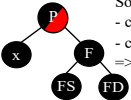
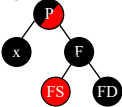
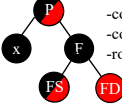
Sunrem în cazul 0 ( $x.color = \text{rosu}$ )  $\Rightarrow$  colorăm  $x$  cu negru.

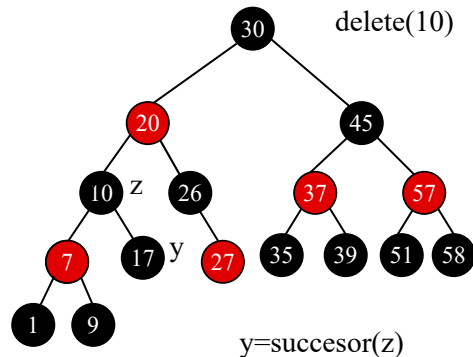
# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P



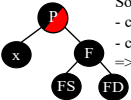
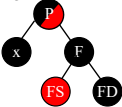
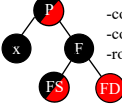


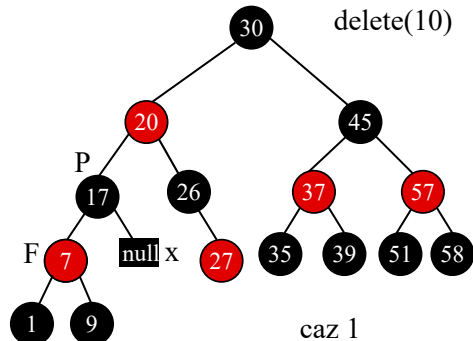
# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P



# ARN ștergere - Exemple



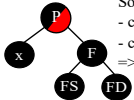
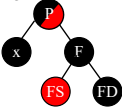
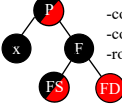
Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P

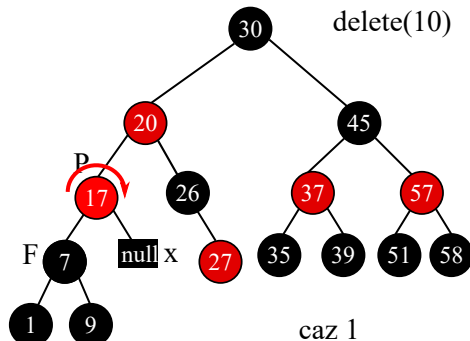


Suntem în cazul 1.



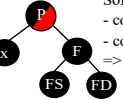
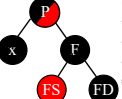
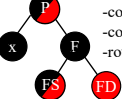


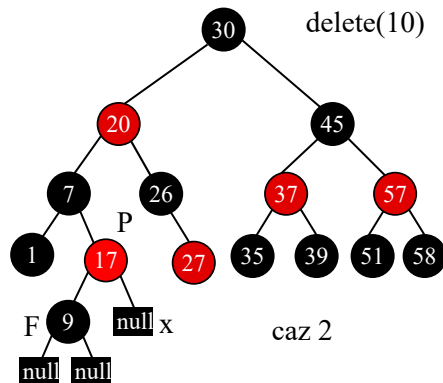
# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P




# ARN ștergere - Exemple


Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P

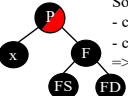


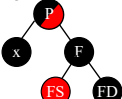
Suntem în cazul 2  $\Rightarrow$  colorăm  $F$  roșu și urcăm  $x = x.p$ . Reluăm toate cazurile.

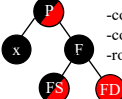
## ARN ștergere - Exemple

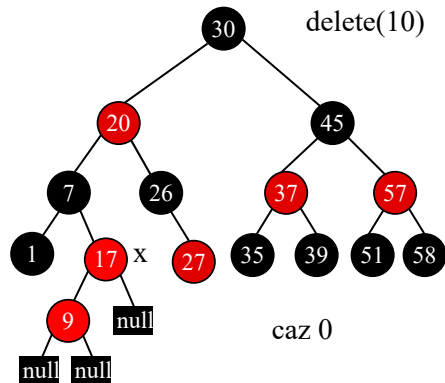
Caz 0:  Sol: colorez x negru

Caz 1: Sol:  
  
 -colorez P roșu  
 -colorez F negru  
 -rotație la st după P  
 => caz 2, 3 sau 4

Caz 2: Sol:  
  
 - colorează F roșu  
 - continuă cu x = P  
 => reiau toate cazurile



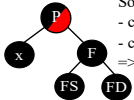
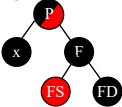
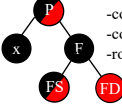
Caz 3 Sol:  
  
 -colorez F roșu  
 -colorez FS negru  
 -rotesc dr. după F  
 => caz 4

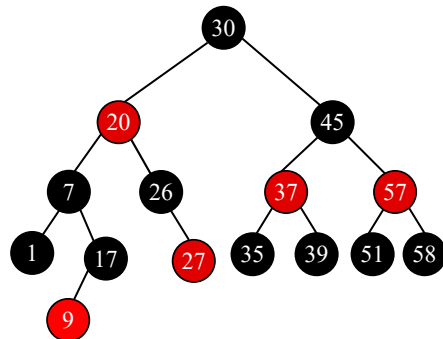
Caz 4 Sol:  
  
 -colorez F cu P.color  
 -colorez P, FD negru  
 -rotesc st. după P





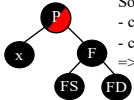
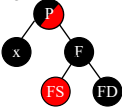
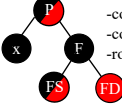
x roșu => caz 0 => coloram x negru. STOP

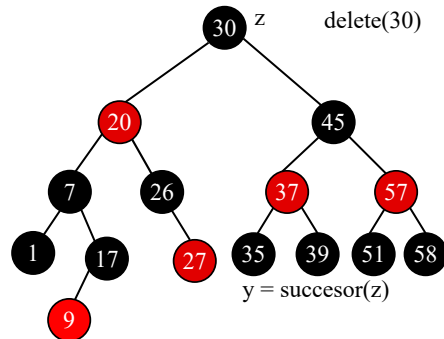
# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P



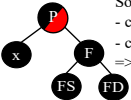
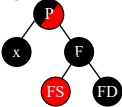
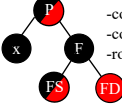


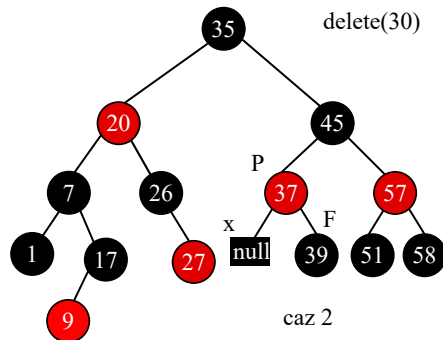
# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P



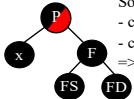
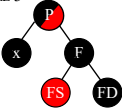
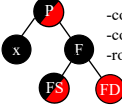


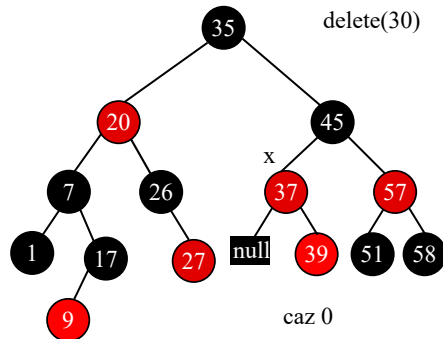
# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3:		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4:		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P



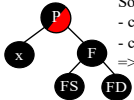
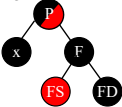
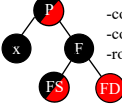


# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P



# ARN ștergere - Exemple

Caz 0:		Sol: colorez x negru
Caz 1:		Sol: -colorez P roșu -colorez F negru -rotație la st după P => caz 2, 3 sau 4
Caz 2:		Sol: - colorează F roșu - continuă cu x = P => reiau toate cazurile
Caz 3		Sol: -colorez F roșu -colorez FS negru -rotesc dr. după F => caz 4
Caz 4		Sol: -colorez F cu P.color -colorez P, FD negru -rotesc st. după P

