

# Îmbogățirea structurilor de date

Universitatea "Transilvania" din Brașov

April 25, 2018

## Etape

- ① Alegerea unei structuri de date
- ② Alegerea informației potrivite pentru îmbogățire
- ③ Demonstrarea păstrării complexității operațiilor
- ④ Dezvoltarea de noi operații

Cum determin al k-lea element în ordinea sortată dintr-o mulțime de elemente?

**Exemplu:** Cum determin al 4-lea element ca valoare în șirul  $\{5, 1, 0, 4, 7, 11, 24, 3, 14\}$ ?

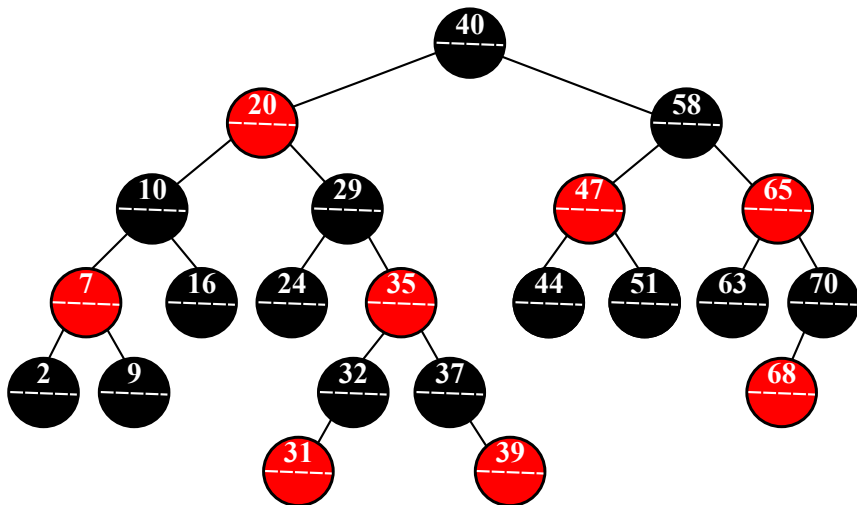
## Întrebare

Cum pot menține o mulțime de elemente astfel încât la orice moment să pot determina  
- EFICIENT - al k-lea element ca valoare din mulțime, indiferent câte inserții și ștergeri s-au efectuat?

# Arbori pentru statistici de ordine

## Răspuns

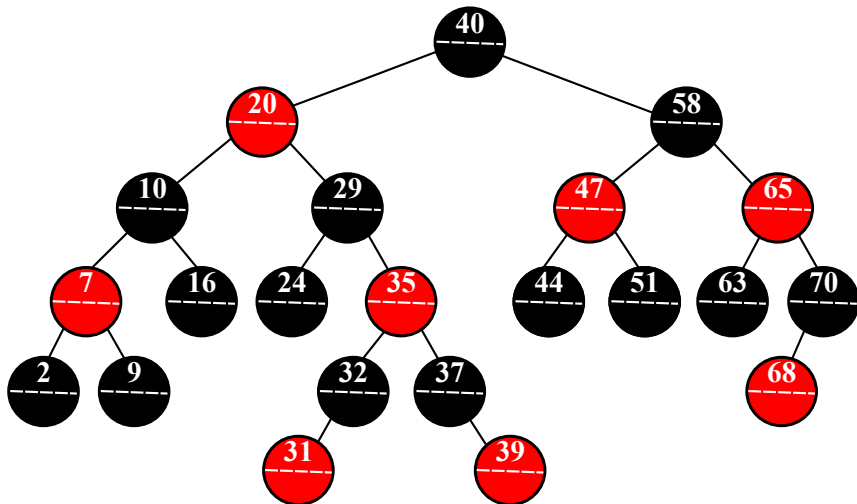
Îmbogățim un ARN! Dar cu ce informație?



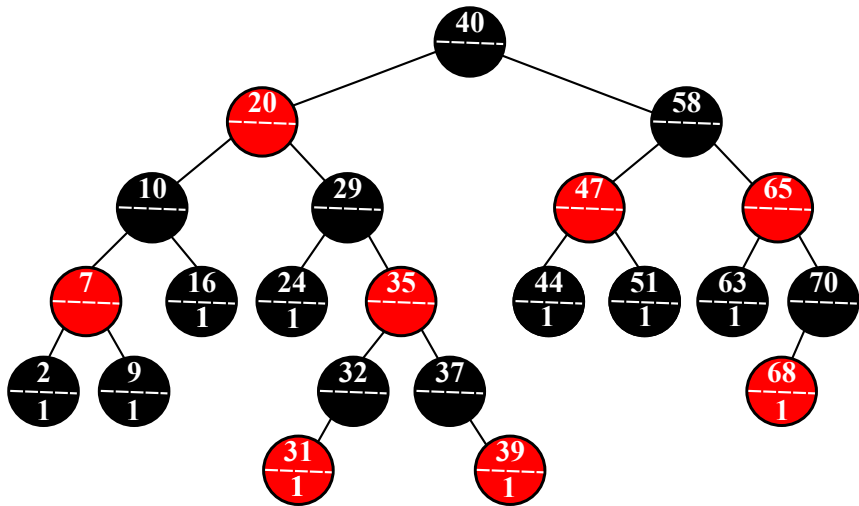
# Arbori pentru statistici de ordine

## Răspuns

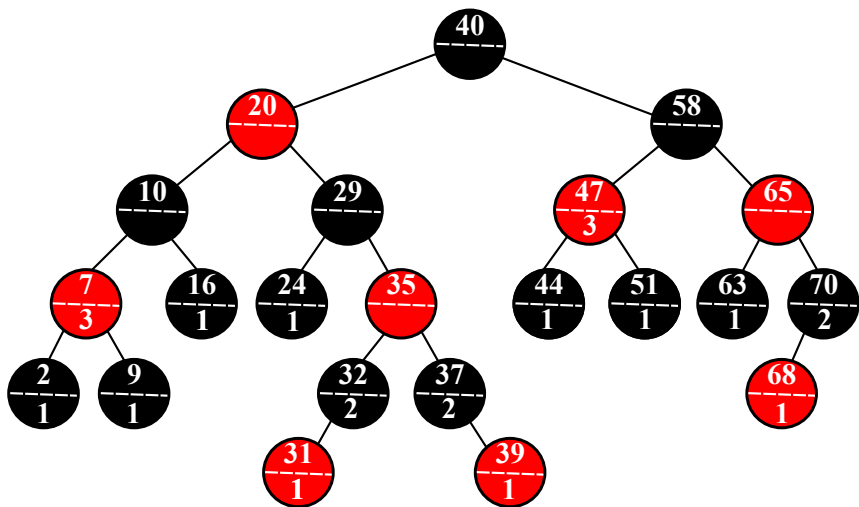
Fiecare nod  $x$  - un câmp  $x.size$  = nr. de noduri din subarborele de rădăcină  $x$ .



# Arbori pentru statistici de ordine



# Arbori pentru statistici de ordine

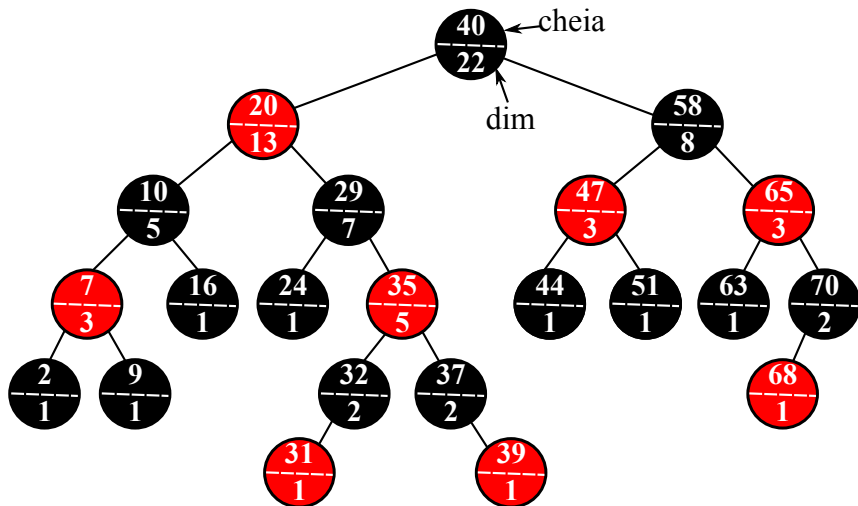


Ce relație există între  $x.size$  și  $x.st.size$ ,  $x.dr.size$ ?



# Arbori pentru statistici de ordine

$$x.size = x.st.size + x.dr.size + 1$$



## Îmbogățirea ARN - Etape

- 1 Alegerea unei structuri de date - ARN

## Îmbogățirea ARN - Etape

- 1 Alegerea unei structuri de date - ARN
- 2 Alegerea informației potrivite pentru îmbogățire - *size*

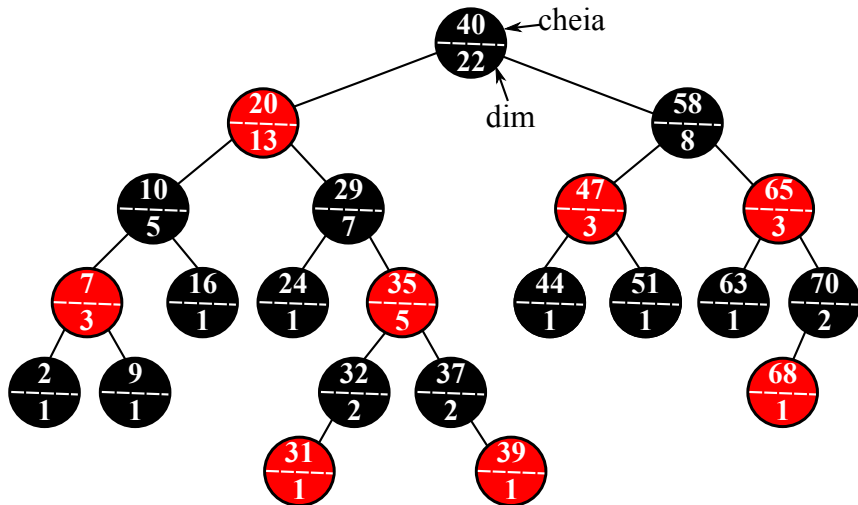
## Îmbogățirea ARN - Etape

- ① Alegerea unei structuri de date - ARN
- ② Alegerea informației potrivite pentru îmbogățire - *size*
- ③ Demonstrarea păstrării complexității operațiilor - urmează

## Îmbogățirea ARN - Etape

- ① Alegerea unei structuri de date - ARN
- ② Alegerea informației potrivite pentru îmbogățire - *size*
- ③ Demonstrarea păstrării complexității operațiilor - urmează
- ④ Dezvoltarea de noi operații - care?

## Căutarea elementului de rang $R$



# Arbori pentru statistici de ordine - Căutarea elementului de rang $R$

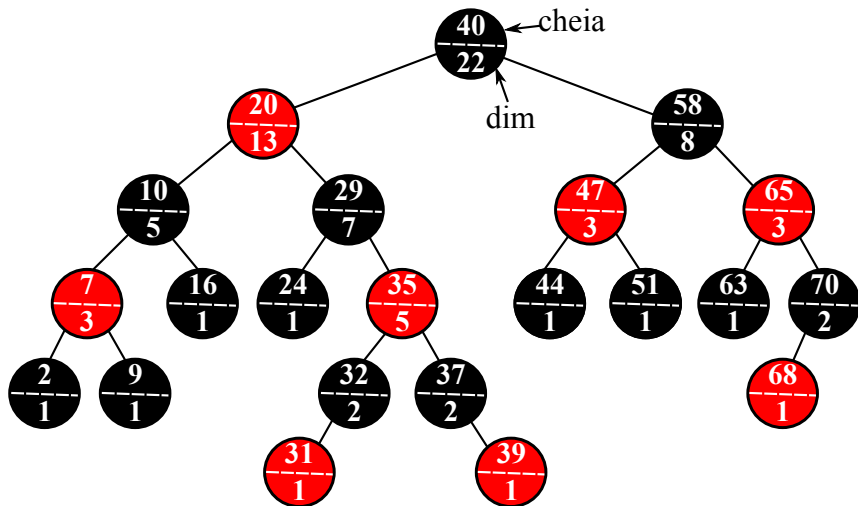
## Algoritm

```
ArbStat_SELECT( $T, R$ )  
   $x = T.rad$   
  cat timp  $x \neq T.Nil$   
     $rang = x.st.size + 1$   
    daca  $R = rang$  atunci  
      RETURN  $x$   
    sfarsit daca  
    daca  $R < rang$  atunci  
       $x = x.st$   
    altfel  
       $x = x.dr$   
       $R = R - rang$   
    sfarsit daca  
  sfarsit cat timp  
  RETURN  $x$ 
```

**Complexitate:**  $O(\log_2 n)$

# Arbori pentru statistici de ordine - Operații noi

## Rangul unui nod $x$



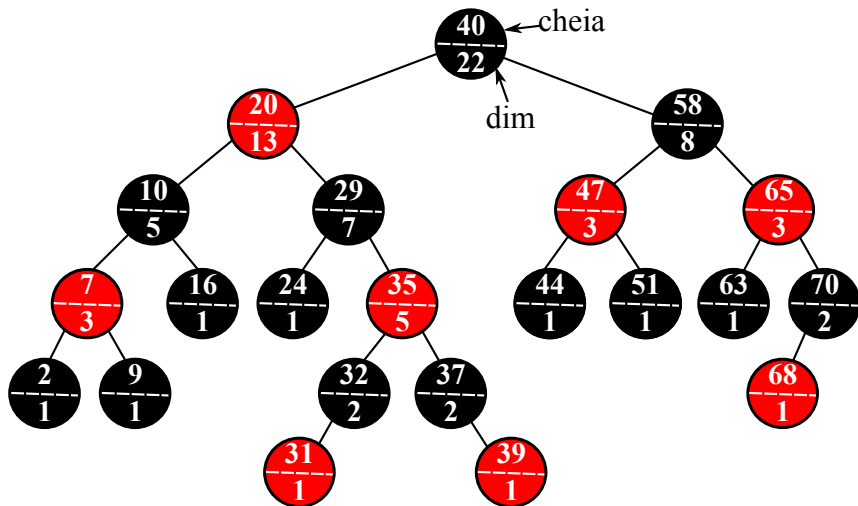


Observații:

- Dacă  $x = x.p.st$  atunci  $Rangul(x)$  în subarborele de rădăcină  $x.p$  este  $x.st.dim + 1$
- Dacă  $x = x.p.dr$  atunci  $Rangul(x)$  în subarborele de rădăcină  $x.p$  este  $x.p.st.dim + 1 + x.st.dim + 1$ .

# Arbori pentru statistici de ordine - Căutarea elementului de rang $R$

Rangul unui nod  $x$



## Algoritm

```
ArbStat_RANG( $T, x$ )  
  rang =  $x.st.size + 1$   
  cat timp  $x \neq T.rad$   
     $y = x.p$   
    daca  $x = y.dr$  atunci  
      rang = rang +  $y.st.size + 1$   
    sfarsit daca  
     $x = y$   
  sfarsit cat timp  
  RETURN rang
```

**Complexitate:**  $O(\log_2 n)$

**Teoremă:** Considerăm un atribut/câmp suplimentar  $f$  prin care se îmbogățește un arbore roșu-negru  $T$  cu  $n$  noduri. Dacă pentru oricare nod  $x$  valoarea câmpului  $x.f$  depinde doar de informațiile din nodurile  $x$ ,  $x.st$ ,  $x.dr$  și eventual de valorile  $x.st.f$  și  $x.dr.f$ , atunci valoarea câmpului suplimentar poate fi actualizată după orice operație de inserție/ștergere fără a afecta complexitatea  $O(\log_2 n)$  a acestor operații.

## **Demonstrație** - Inserția nodului $x$

- dacă nodul  $x$  - fiu al nodului  $y \Rightarrow x.f$  se calculează în  $O(1)$  (depinde de  $x$  și  $T.nil$ ).

## Demonstrație - Inserția nodului $x$

- dacă nodul  $x$  - fiu al nodului  $y \Rightarrow x.f$  se calculează în  $O(1)$  (depinde de  $x$  și  $T.nil$ ).
- evtl. trebuie recalcultat  $y.f$  - tot  $O(1)$  (depinde de  $x$  și celălalt fiu al lui  $y$ )

## Demonstrație - Inserția nodului $x$

- dacă nodul  $x$  - fiu al nodului  $y \Rightarrow x.f$  se calculează în  $O(1)$  (depinde de  $x$  și  $T.nil$ ).
- evtl. trebuie recalcultat  $y.f$  - tot  $O(1)$  (depinde de  $x$  și celălalt fiu al lui  $y$ )
- ... se continuă la  $y.p$  și eventual până la rădăcină.

## Demonstrație - Inserția nodului $x$

- dacă nodul  $x$  - fiu al nodului  $y \Rightarrow x.f$  se calculează în  $O(1)$  (depinde de  $x$  și  $T.nil$ ).
- evtl. trebuie recalcultat  $y.f$  - tot  $O(1)$  (depinde de  $x$  și celălalt fiu al lui  $y$ )
- ... se continuă la  $y.p$  și eventual până la rădăcină.
- $\Rightarrow O(\log_2 n)$ .



## Demonstrație - Inserția nodului $x$

- dacă nodul  $x$  - fiu al nodului  $y \Rightarrow x.f$  se calculează în  $O(1)$  (depinde de  $x$  și  $T.nil$ ).
- evtl. trebuie recalcultat  $y.f$  - tot  $O(1)$  (depinde de  $x$  și celălalt fiu al lui  $y$ )
- ... se continuă la  $y.p$  și eventual până la rădăcină.
- $\Rightarrow O(\log_2 n)$ .
- pentru refacere a proprietăților RN - nr. limitat de rotații  $\rightarrow O(1)$

**Exemplu:** Arbori pentru statistici de ordine - câmpul *size*

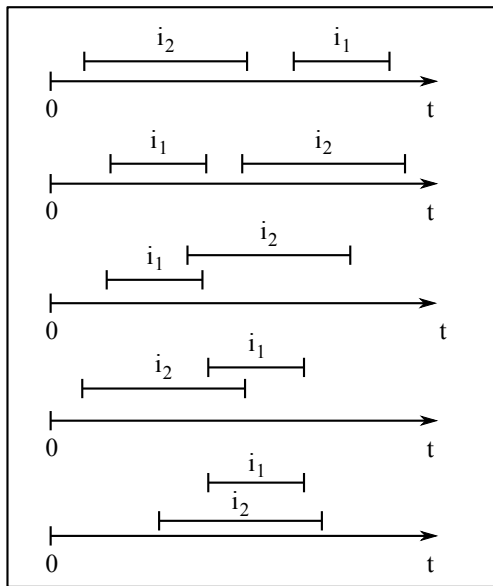
$$x.size = x.st.size + x.dr.size + 1$$

## Îmbogățirea cu un câmp

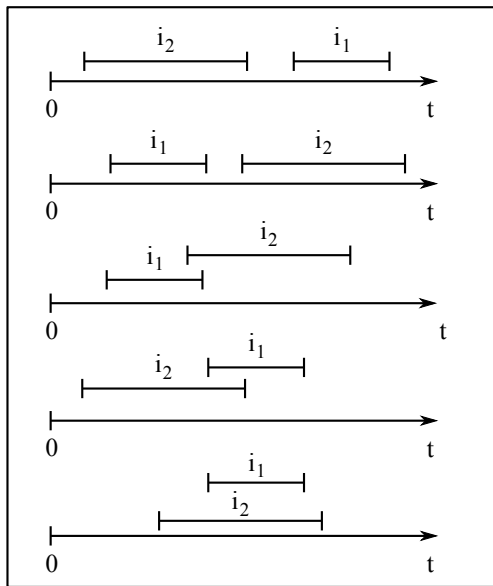
- max / min
- black\_height
- height
- sum
- etc.

- Arbori care au ca informație un interval
- Câmpul cheie  $x.interval$  care are componentele
  - $interval.low$  = limita stângă a intervalului
  - $interval.high$  - limita dreaptă a intervalului

# Arbori pentru intervale - Compararea intervalelor



# Arbori pentru intervale - Compararea intervalelor



a. Nu se intersectează

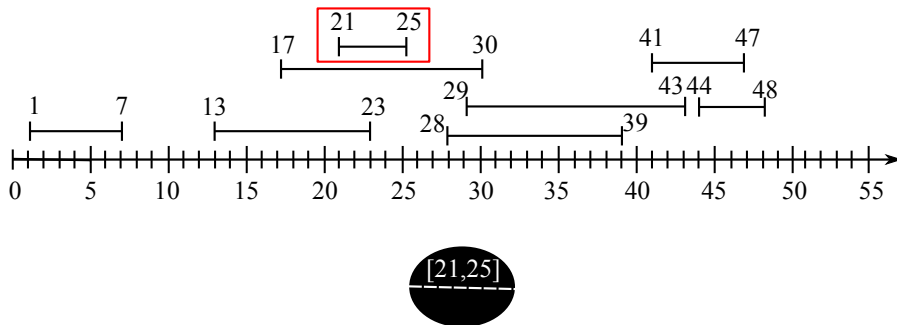
$$\Rightarrow \begin{cases} i_1.high < i_2.low \\ \text{sau} \\ i_2.high < i_1.low \end{cases}$$

b. Se intersectează, atunci

$$i_1.low \leq i_2.high \text{ și } i_2.low \leq i_1.high$$

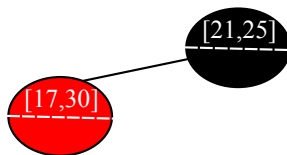
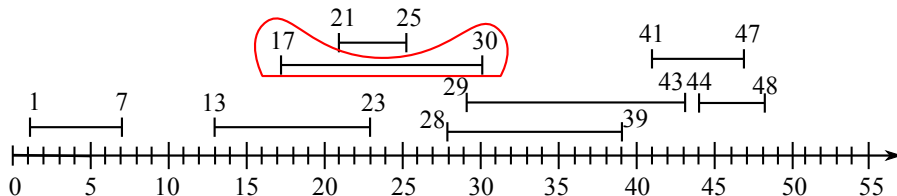
# Arbore RN pentru intervale - construcție

Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$



# Arbore RN pentru intervale - construcție

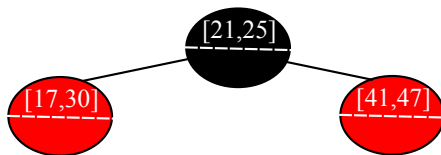
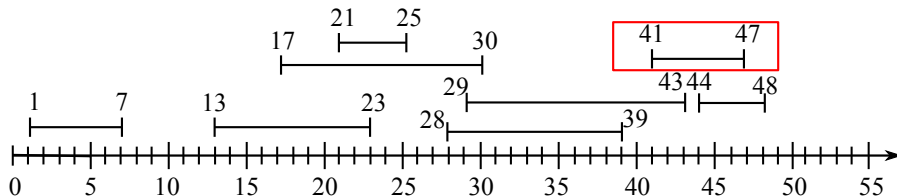
Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.\text{interval.low}$





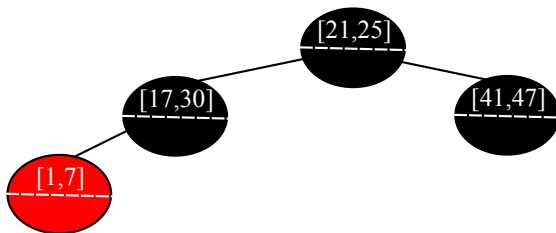
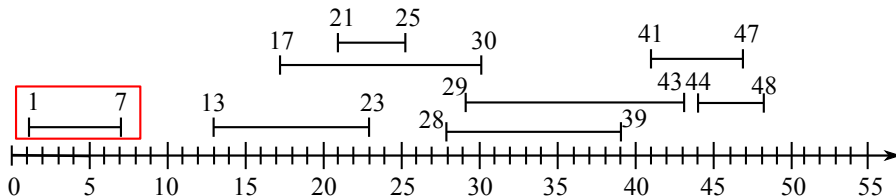
# Arbore RN pentru intervale - construcție

Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$



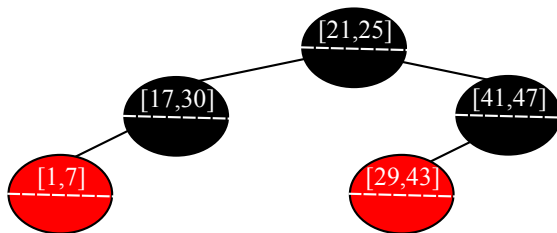
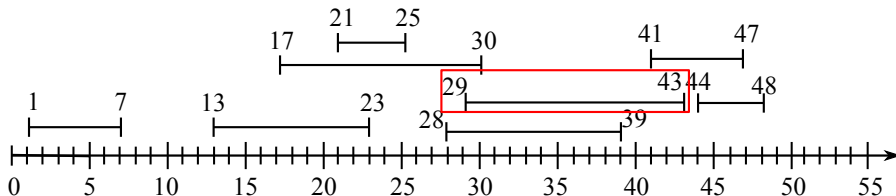
# Arbore RN pentru intervale - construcție

Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$



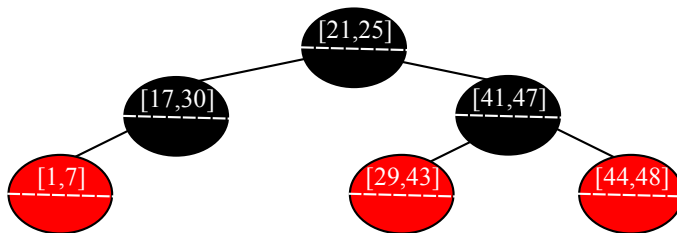
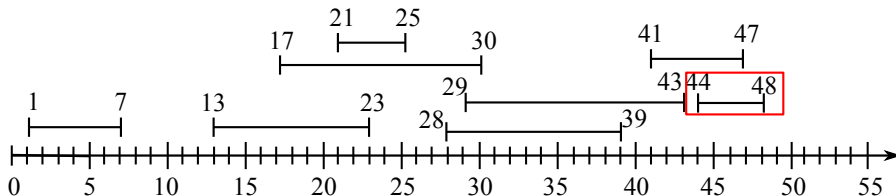
# Arbore RN pentru intervale - construcție

Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$



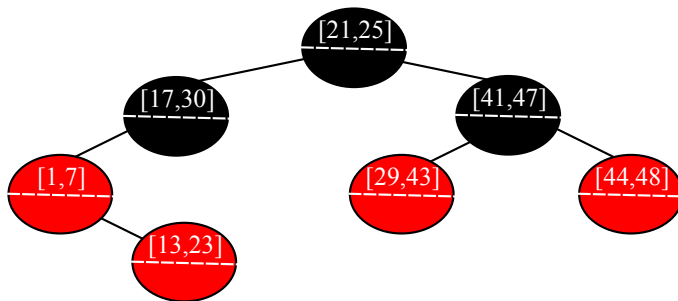
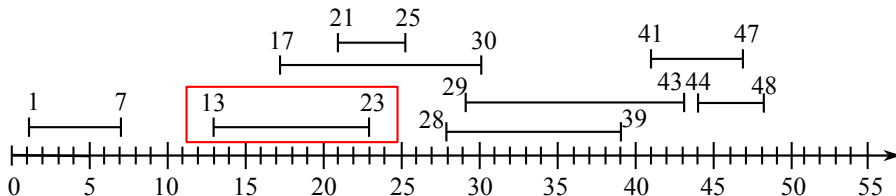
# Arbore RN pentru intervale - construcție

Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$



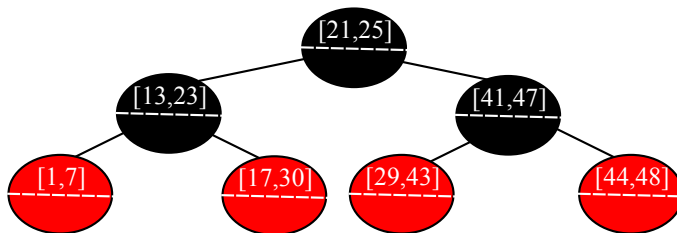
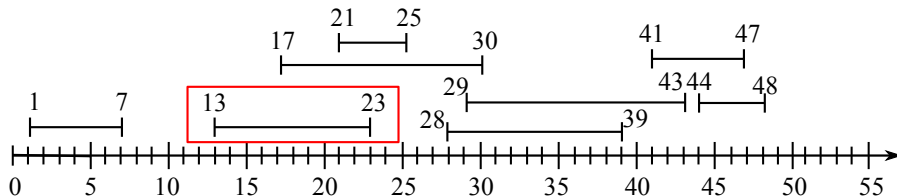
# Arbore RN pentru intervale - construcție

Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$



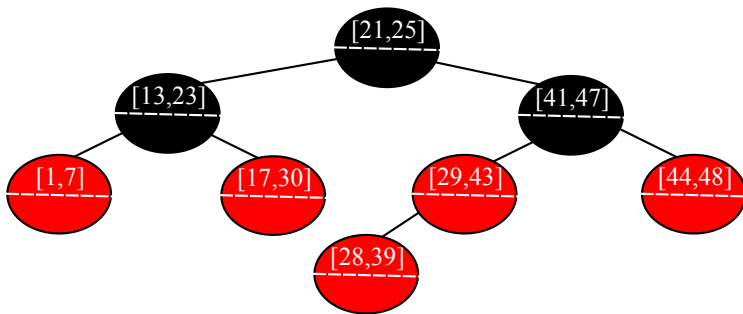
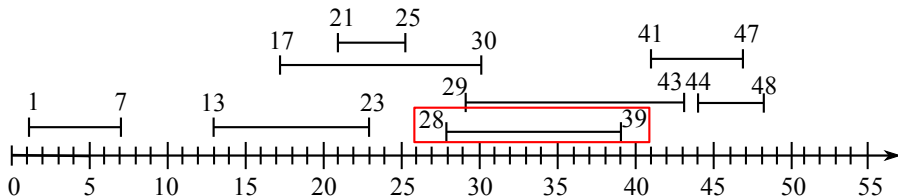
# Arbore RN pentru intervale - construcție

Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$



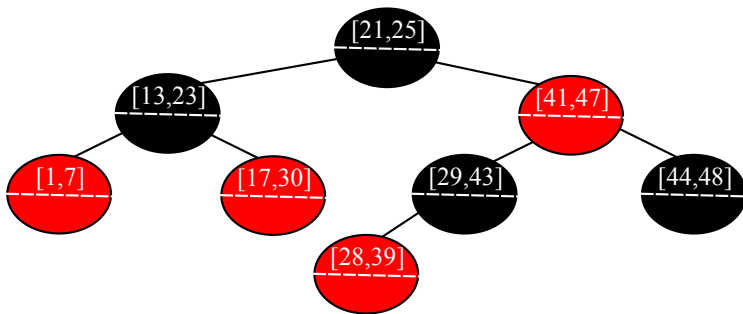
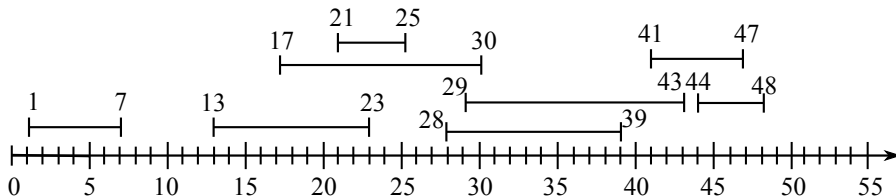
# Arbore RN pentru intervale - construcție

Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$



# Arbore RN pentru intervale - construcție

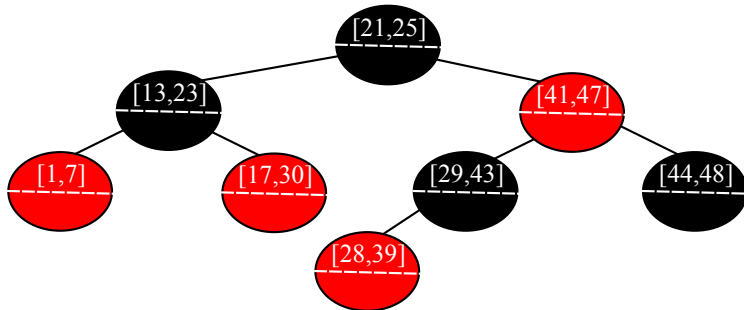
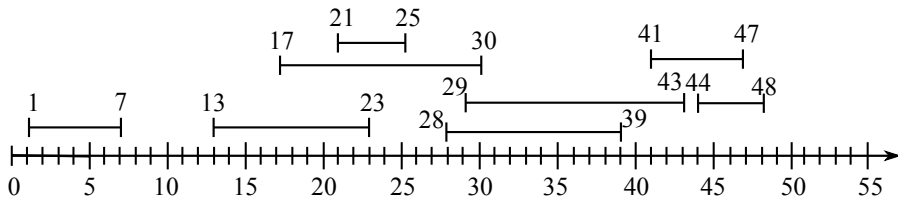
Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este  $x.interval.low$





# Arbore RN pentru intervale - construcție

Căutarea unui interval în arbore care intersectează un interval  $I$  dat. Ex:  $I = [26, 27]$



## Îmbogățirea ARN - Etape

- 1 Alegerea unei structuri de date - ARN

## Îmbogățirea ARN - Etape

- ① Alegerea unei structuri de date - ARN
- ② Alegerea informației potrivite pentru îmbogățire - *max\_ih*

## Îmbogățirea ARN - Etape

- 1 Alegerea unei structuri de date - ARN
- 2 Alegerea informației potrivite pentru îmbogățire - *max\_ih*
- 3 Demonstrarea păstrării complexității operațiilor

## Îmbogățirea ARN - Etape

- 1 Alegerea unei structuri de date - ARN
- 2 Alegerea informației potrivite pentru îmbogățire - *max\_ih*
- 3 Demonstrarea păstrării complexității operațiilor
  - $x.max\_ih = \max(x.st.max\_ih, x.dr.max\_ih, x.int.high)$

## Îmbogățirea ARN - Etape

- 1 Alegerea unei structuri de date - ARN
- 2 Alegerea informației potrivite pentru îmbogățire - *max\_ih*
- 3 Demonstrarea păstrării complexității operațiilor
  - $x.max\_ih = \max(x.st.max\_ih, x.dr.max\_ih, x.int.high)$
- 4 Dezvoltarea de noi operații:

## Îmbogățirea ARN - Etape

- ① Alegerea unei structuri de date - ARN
- ② Alegerea informației potrivite pentru îmbogățire - *max\_ih*
- ③ Demonstrarea păstrării complexității operațiilor
  - $x.max\_ih = \max(x.st.max\_ih, x.dr.max\_ih, x.int.high)$
- ④ Dezvoltarea de noi operații:
  - Căutarea unui interval din arbore, care intersectează un anumit interval dat *I*

## Îmbogățirea ARN - Etape

- ① Alegerea unei structuri de date - ARN
- ② Alegerea informației potrivite pentru îmbogățire -  $max\_ih$
- ③ Demonstrarea păstrării complexității operațiilor
  - $x.max\_ih = \max(x.st.max\_ih, x.dr.max\_ih, x.int.high)$
- ④ Dezvoltarea de noi operații:
  - Căutarea unui interval din arbore, care intersectează un anumit interval dat  $I$
  - Căutarea unui interval din arbore, care intersectează un anumit interval dat  $I$  și are capătul stâng minim

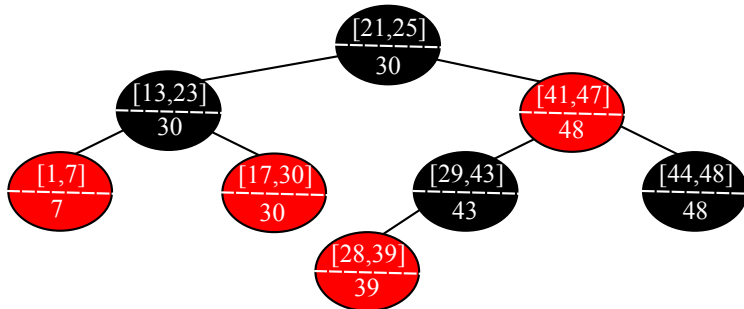
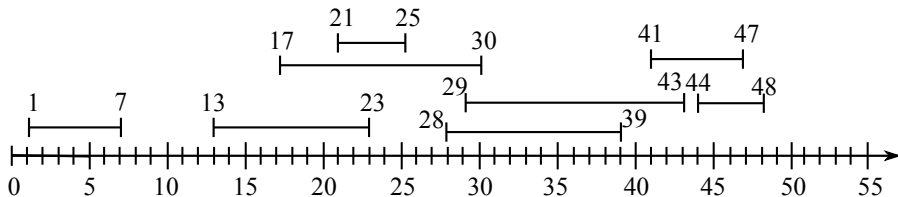


## Îmbogățirea ARN - Etape

- ① Alegerea unei structuri de date - ARN
- ② Alegerea informației potrivite pentru îmbogățire -  $max\_ih$
- ③ Demonstrarea păstrării complexității operațiilor
  - $x.max\_ih = \max(x.st.max\_ih, x.dr.max\_ih, x.int.high)$
- ④ Dezvoltarea de noi operații:
  - Căutarea unui interval din arbore, care intersectează un anumit interval dat  $I$
  - Căutarea unui interval din arbore, care intersectează un anumit interval dat  $I$  și are capătul stâng minim
  - Căutarea tuturor intervalelor din arbore, care intersectează un anumit interval dat  $I$

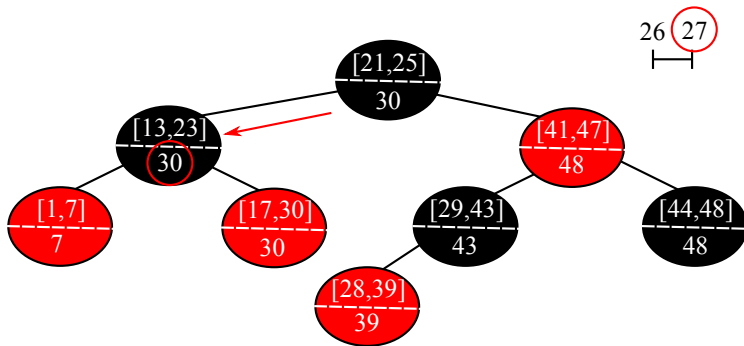
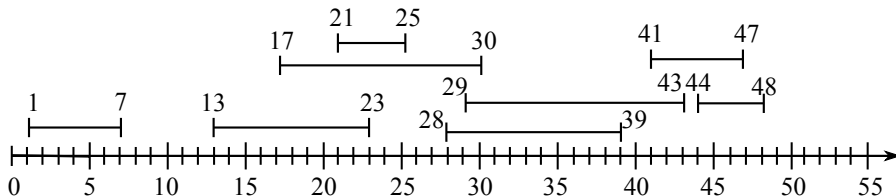
# Arbore RN pentru intervale - construcție

Căutarea unui interval în arbore care intersectează un interval  $I$  dat. Ex:  $I = [26, 27]$



# Arbore RN pentru intervale - construcție

Compar: Dacă  $I \cap x.int = \emptyset$  atunci dacă  $I.low \leq x.st.max\_ih \Rightarrow$  cobor pe stânga, altfel cobor pe dreapta.



# Arbore RN pentru intervale - construcție

Compar: Dacă  $I \cap x.int = \emptyset$  atunci

dacă  $I.low \leq x.st.max\_ih \Rightarrow$  cobor pe stânga, altfel cobor pe dreapta.

