Îmbogățirea structurilor de date

Universitatea ”Transilvania” din Brașov

30 aprilie 2022

Îmbogățirea unei structuri de date

Se referă la: atașarea de informații suplimentare elementelor structurii - adică adăugarea de câmpuri suplimentare fiecărui nod

Îmbogățirea unei structuri de date

Se referă la: atașarea de informații suplimentare elementelor structurii - adică adăugarea de câmpuri suplimentare fiecărui nod

Scop:

- efectuarea altor operații - cereri - în afara celor uzuale
- eficientizarea operațiilor uzuale

Îmbogățirea unei structuri de date

Etape

1. Alegerea unei structuri de date

Îmbogățirea unei structuri de date

Etape

1. Alegerea unei structuri de date
2. Alegerea informației potrivite pentru îmbogățire

Îmbogățirea unei structuri de date

Etape

1. Alegerea unei structuri de date
2. Alegerea informației potrivite pentru îmbogățire
3. Demonstrarea păstrării complexității operațiilor

Îmbogățirea unei structuri de date

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?

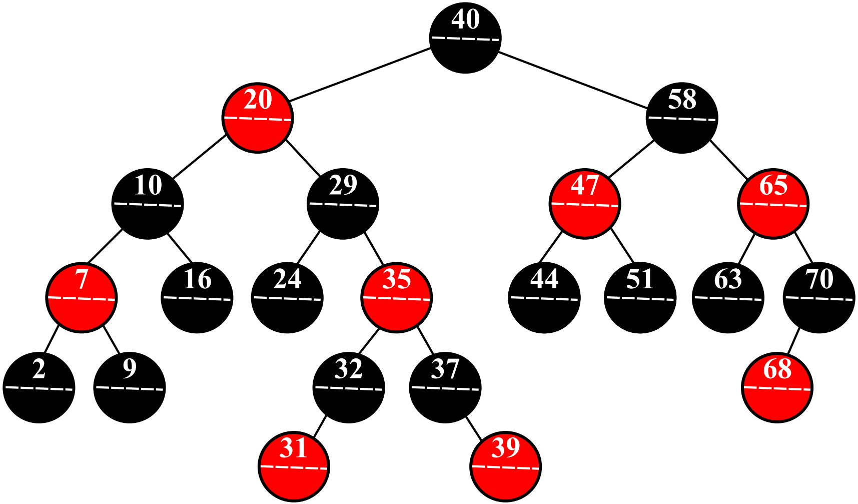
Exemplu: Cum determin al 4-lea element ca valoare în șirul $\{5, 1, 0, 4, 7, 11, 24, 3, 14\}$?

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 ștergei s-au efectuat?

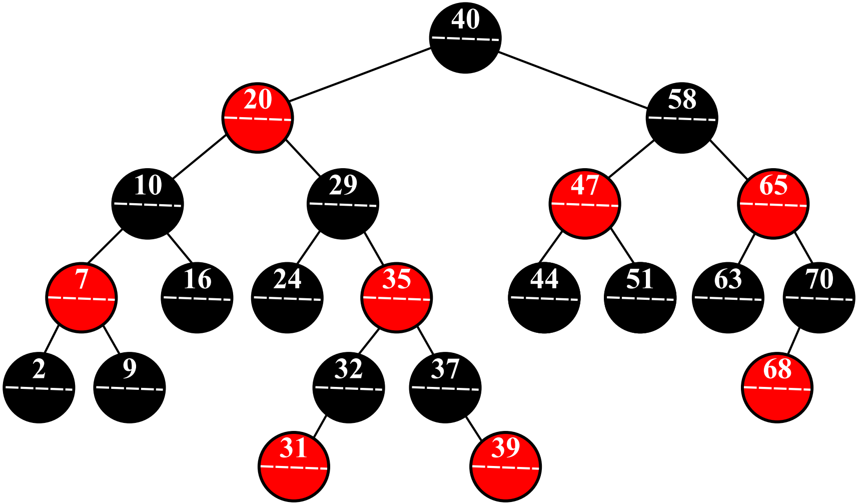
Răspuns

Îmbogățim un ARN! Dar cu ce informație?

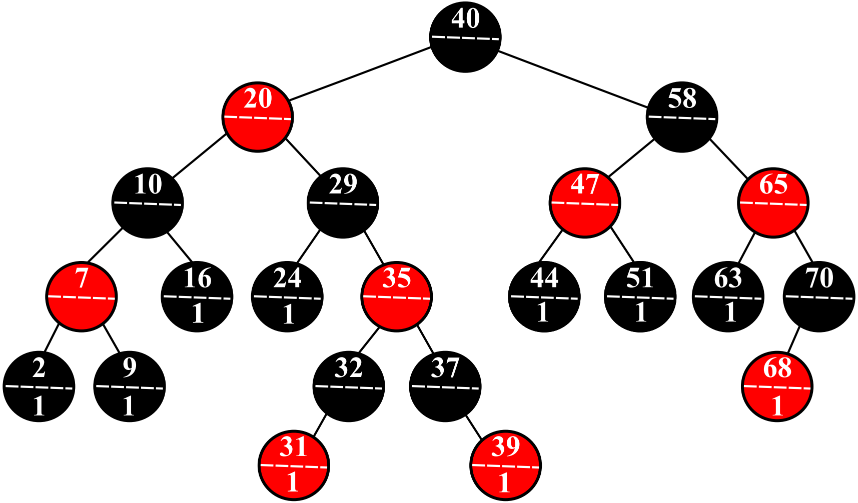


Răspuns

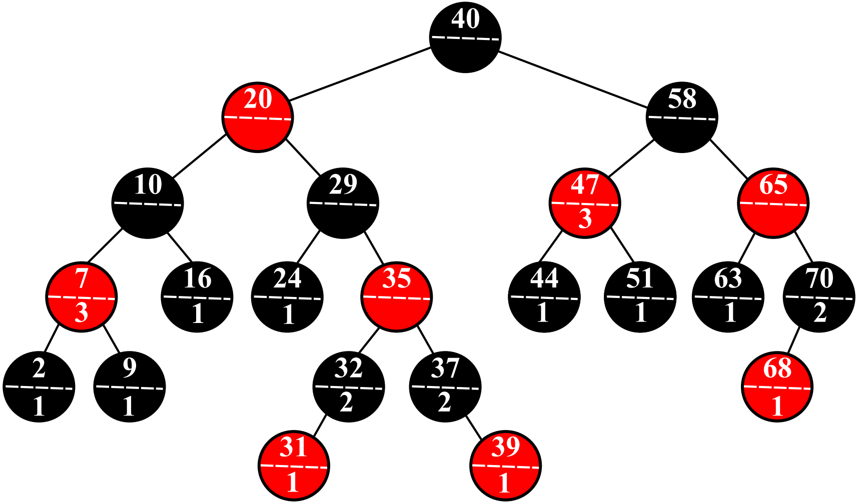
Fiecare nod x - un câmp $x.size$ = nr. de noduri din subarborele de rădăcină x.



Arbori pentru statistici de ordine

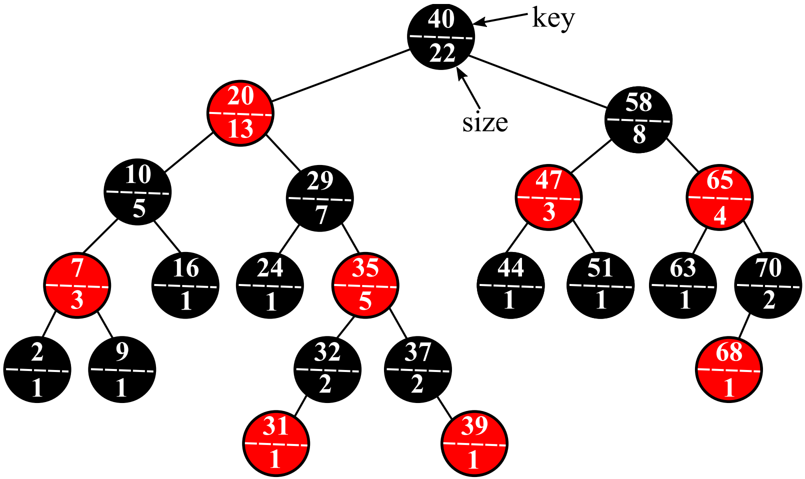


Arbori pentru statistici de ordine



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

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



Arbori pentru statistici de ordine

Un *arbore pentru statistici de ordine* este un ARN, în care fiecare nod x conține un câmp suplimentar, *size*, care reprezintă numărul de noduri - fără frunzele nil - ale subarborelui cu rădăcina x.
În cazul nodului santinelă $T.nil$, se consideră $T.nil.dim = 0$. Se observă ușor relația:

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

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

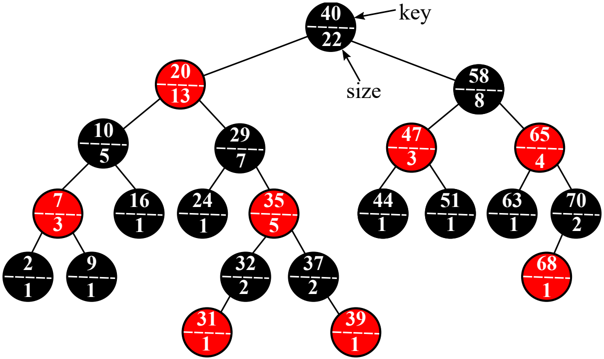
1. Alegerea unei structuri de date - ARN
2. Alegerea informației potrivite pentru îmbogățire - *size*
3. Demonstrarea păstrării complexității operațiilor - urmează
4. Dezvoltarea de noi operații - care?

(I) Căutarea elementului de rang i = elementul a cărui cheie s-ar afla pe poziția i în șirul sortat al cheilor din arbore.

Arbori pentru statistici de ordine - Operații noi

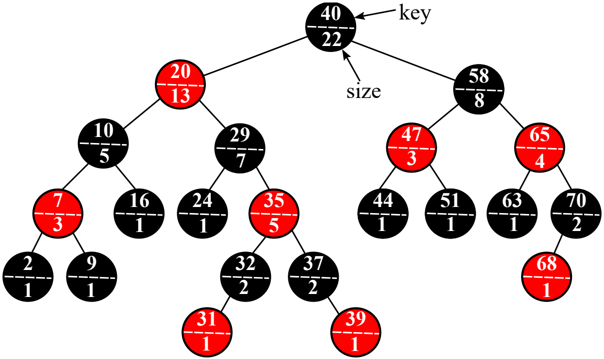
(I) Căutarea elementului de rang i = elementul a cărui cheie s-ar afla pe poziția i în șirul sortat al cheilor din arbore.

(II) Determinarea rangului unui element = poziția pe care s-ar afla cheia elementului respectiv în șirul sortat al cheilor din arbore.



Select(R)

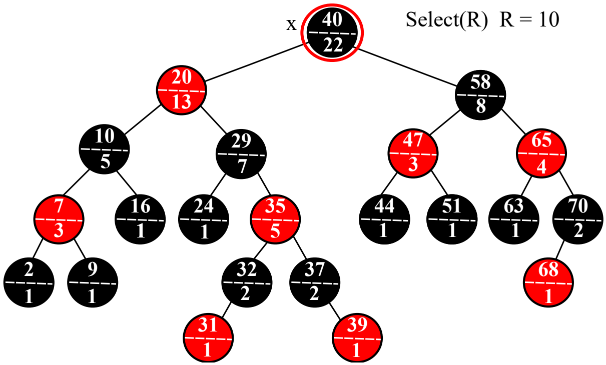
Observație: rangul rădăcinii = ?



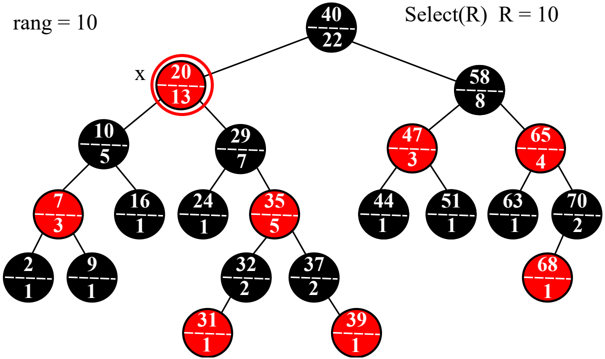
Select(R)

Observație: rangul rădăcinii $=$ T.rad.st.size $+ 1 = 14$

Pornind de la un nod x, rangul lui x în cadrul mulțimii formate din elementele aflate în subarborele x este $x.st.size + 1$.



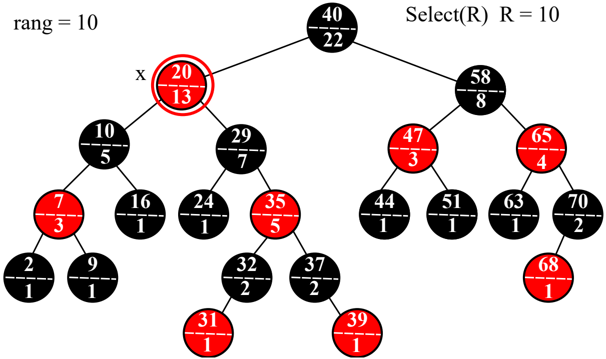
Select(R): $R = 10$
Pornim cu $x = T.rad \Rightarrow rang(x) = T.rad.st.size + 1 = 14 \Rightarrow$ nodul cu rangul 10 trebuie căutat în stânga rădăcinii.



Select(R): $R = 10$

Pornim cu $x = T.rad \Rightarrow rang(x) = T.rad.st.size + 1 = 14 \Rightarrow$ nodul cu rangul 10 trebuie căutat în stânga rădăcinii.

$\Rightarrow x = T.rad.st$. Rangul lui x în cadrul subarborelui este $x.st.size + 1 = 6 < 10 \Rightarrow x \leftarrow x.dr$

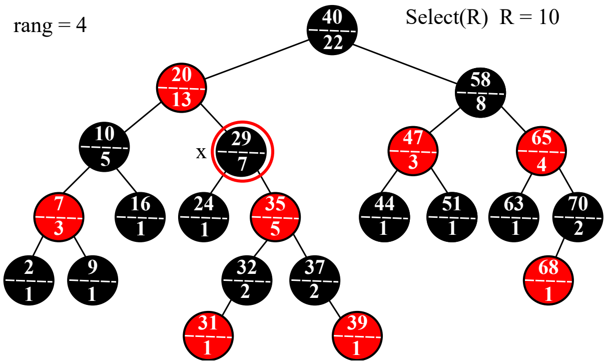


Select(R): $R = 10$

Pornim cu $x = T.rad \Rightarrow rang(x) = T.rad.st.size + 1 = 14 \Rightarrow$ nodul cu rangul 10 trebuie căutat în stânga rădăcinii.

$\Rightarrow x = T.rad.st$. Rangul lui x în cadrul subarborelui este $x.st.size + 1 = 6 < 10 \Rightarrow x \leftarrow x.dr$

Observație: când se coboară pe dreapta pentru a ajunge la nodul curent trebuie întâi parcurse toate nodurile din stânga părintelui + părintele $\Rightarrow$ înainte de a coborâ pe dreapta se face $R \leftarrow R - (x.st.size + 1)$

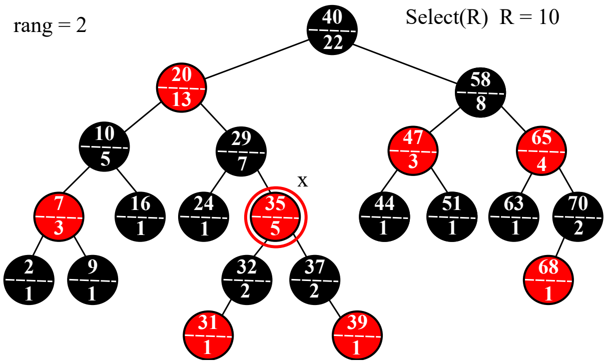


Deci:

$R \leftarrow R - x.st.size - 1$

$x \leftarrow x.dr.$

Acum $rang(x) = 2 < R = 4 \Rightarrow$



Deci:

$R \leftarrow R - x.st.size - 1$

$x \leftarrow x.dr.$

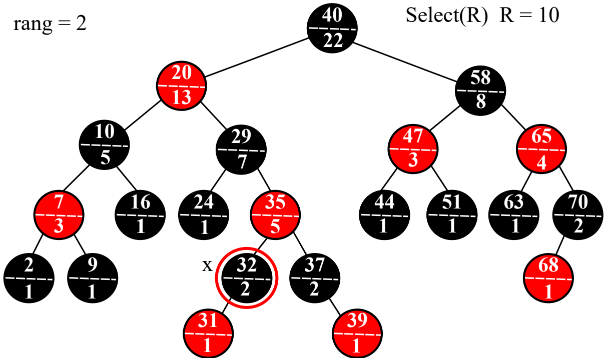
Acum $rang(x) = 2 < R = 4 \Rightarrow$

Cobor pe dreapta:

$R \leftarrow R - x.st.size - 1$

$x \leftarrow x.dr.$

Acum $rang(x) = 3 > R = 2 \Rightarrow$



Deci:

$R \leftarrow R - x.st.size - 1$

$x \leftarrow x.dr.$

Acum $rang(x) = 2 < R = 4 \Rightarrow$

Cobor pe dreapta:

$R \leftarrow R - x.st.size - 1$

$x \leftarrow x.dr.$

Acum $rang(x) = 3 > R = 2 \Rightarrow$

Cobor pe stânga:

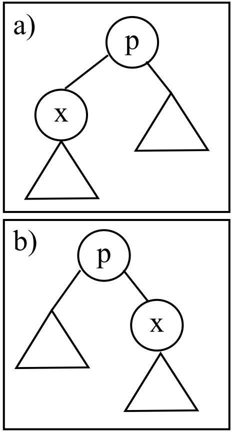
$x \leftarrow x.st$, rang nu se modifică.

Acum $rang(x) = 2 = R \Rightarrow$ STOP

Algoritm 1: Select(T, R)

```
x ← T.rad
cat_timp x ≠ T.Nil executa
    rang ← x.st.size + 1
    daca R = rang atunci
        RETURN x
    sfarsit_daca
    daca R < rang atunci
        x ← x.st
    sfarsit_daca
    altfel
        x ← x.dr
        R ← R − rang
    sfarsit_daca
sfarsit_cat_timp
RETURN x
```

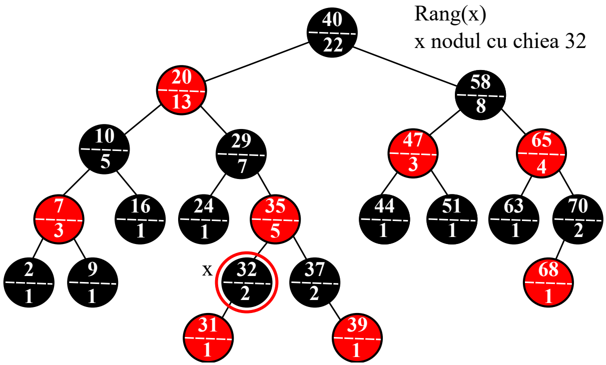
Complexitate: $O(\log_2 n)$



Notăm cu $rang_m(n)$ rangulu nodului n în mulțimea reprezentată de nodurile din subarborele de rădăcină m. Atunci:

- Dacă $x = p.st$ atunci $rang_p(x) = p.st.size + 1 = x.size + 1 = rang_x(x)$
- Dacă $x = p.dr$ atunci $rang_p(x) = p.st.size + 1 + x.st.size + 1 = p.st.size + 1 + rang_x(x)$

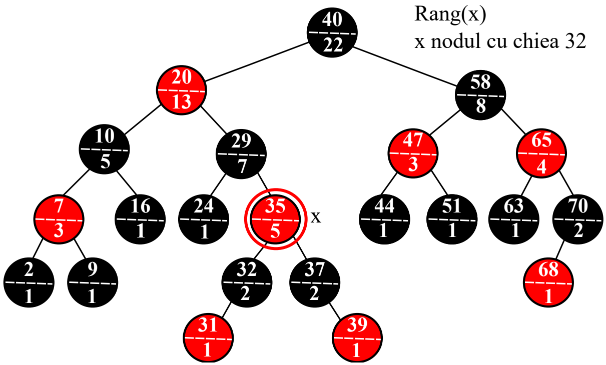
$\Rightarrow$ calculam ranglul unui nod x în manieră bottom-up pornind de la x.



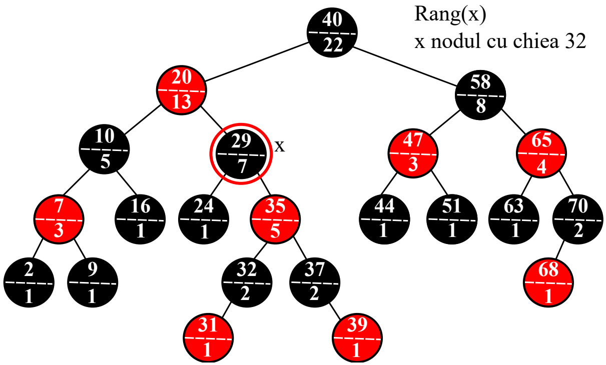
Rang(x) $x =$ nodul cu cheia 32

Inițial $rang = x.st.size + 1 = 2$.

$x = x.p.st \Rightarrow$ în arborele de rădăcină $x.p$ rangul său rămâne 2. Se urcă la părinte.



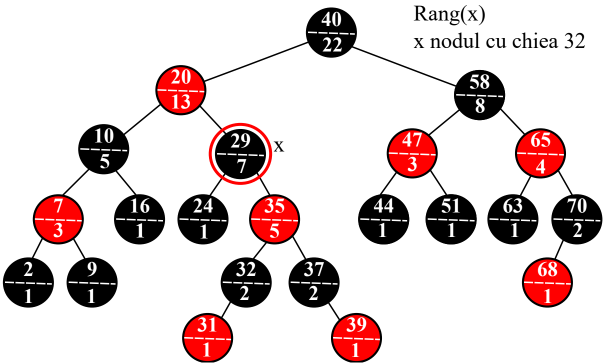
Rang(x) x = nodul cu cheia 32
Inițial $rang = x.st.size + 1 = 2$.
$x = x.p.st \Rightarrow$ în arborele de rădăcină $x.p$ rangul său rămâne 2. Se urcă la părinte.



Rang(x) x = nodul cu cheia 32
Inițial $rang = x.st.size + 1 = 2$.
$x = x.p.st \Rightarrow$ în arborele de rădăcină $x.p$ rangul său rămâne 2. Se urcă la părinte.

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine

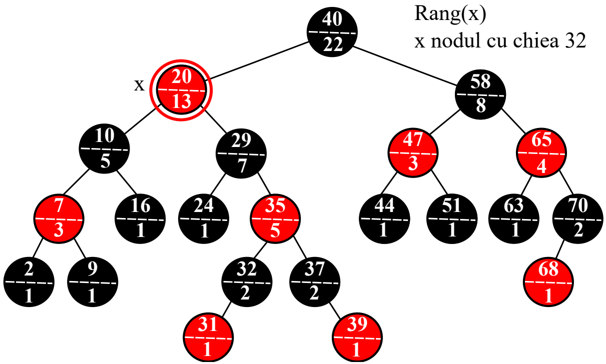
Arbori pentru statistici de ordine - Determinarea rangului unui nod x



Rang(x) $x =$ nodul cu cheia 32
Inițial $rang = x.st.size + 1 = 2$.
$x = x.p.st \Rightarrow$ în arborele de rădăcină $x.p$ rangul său rămâne 2. Se urcă la părinte.

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine $rang + x.p.st.size + 1 = 4$ și $x \leftarrow x.p$

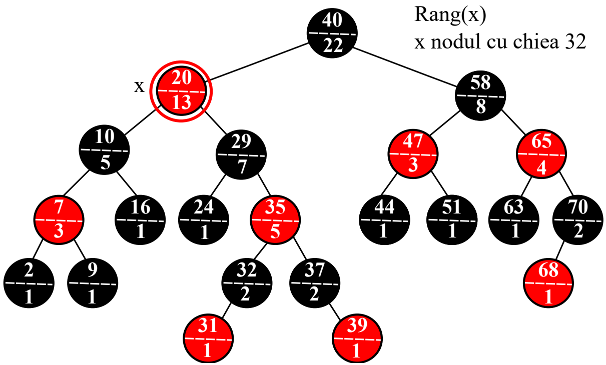
Arbori pentru statistici de ordine - Determinarea rangului unui nod x



Rang(x) $x =$ nodul cu cheia 32
Inițial $rang = x.st.size + 1 = 2$.
$x = x.p.st \Rightarrow$ în arborele de rădăcină $x.p$ rangul său rămâne 2. Se urcă la părinte.

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine
$rang + x.p.st.size + 1 = 4$ și $x \leftarrow x.p$

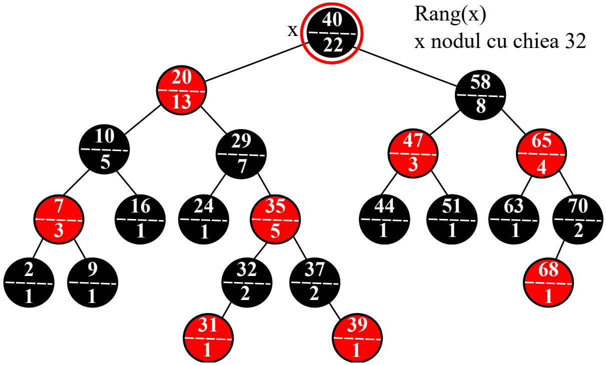
$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine



Rang(x) $x =$ nodul cu cheia 32
Inițial $rang = x.st.size + 1 = 2$.
$x = x.p.st \Rightarrow$ în arborele de rădăcină $x.p$ rangul său rămâne 2. Se urcă la părinte.

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine
$rang + x.p.st.size + 1 = 4$ și $x \leftarrow x.p$

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine
$rang + x.p.st.size + 1 = 10$ și $x \leftarrow x.p$

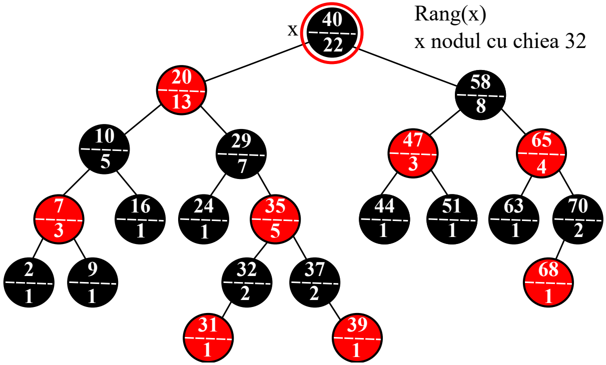


Rang(x) x = nodul cu cheia 32
Inițial $rang = x.st.size + 1 = 2$.
$x = x.p.st \Rightarrow$ în arborele de rădăcină $x.p$ rangul său rămâne 2. Se urcă la părinte.

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine $rang + x.p.st.size + 1 = 4$ și $x \leftarrow x.p$

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine $rang + x.p.st.size + 1 = 10$ și $x \leftarrow x.p$

$x = x.p.st \Rightarrow$ rangul se menține 10. În plus $x \leftarrow x.p$.



Rang(x) $x =$ nodul cu cheia 32
Inițial $rang = x.st.size + 1 = 2$.
$x = x.p.st \Rightarrow$ în arborele de rădăcină $x.p$ rangul său rămâne 2. Se urcă la părinte.

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine $rang + x.p.st.size + 1 = 4$ și $x \leftarrow x.p$

$x = x.p.dr \Rightarrow$ în arborele de rădăcină $x.p$, rangul devine $rang + x.p.st.size + 1 = 10$ și $x \leftarrow x.p$

$x = x.p.st \Rightarrow$ rangul se menține 10. În plus $x \leftarrow x.p$.
Ajung la rădăcină $\Rightarrow$ STOP și obțin $rang(32) = 10$.

Algoritm 2: Rang(x)

$rang \leftarrow x.st.size + 1$
cat_timp $x \neq T.rad$ **executa**
  $y \leftarrow x.p$
  daca $x = y.dr$ **atunci**
    $rang \leftarrow rang + y.st.size + 1$
  sfarsit_daca
  $x \leftarrow y$
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 calculază în $O(1)$ (depinde de x și $T.nil$).

Demonstrație - Inserția nodului x

- dacă nodul x - fiu al nodului $y \Rightarrow x.f$ se calculază 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)

Teorema complexității

Demonstrație - Inserția nodului x

- dacă nodul x - fiu al nodului $y \Rightarrow x.f$ se calculază 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.

Teorema complexității

Demonstrație - Inserția nodului x

- dacă nodul x - fiu al nodului $y \Rightarrow x.f$ se calculază 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.
- $\Rightarrow O(\log_2 n)$.

Teorema complexității

Demonstrație - Inserția nodului x

- dacă nodul x - fiu al nodului $y \Rightarrow x.f$ se calculază 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

Îmbogățirea arborilor binari de căutare- Alte exemple

Îmbogățirea cu un câmp

- max / min
- black_height

Îmbogățirea arborilor binari de căutare- Alte exemple

Îmbogățirea cu un câmp

- max / min
- black_height
- height

Îmbogățirea arborilor binari de căutare- Alte exemple

Îmbogățirea cu un câmp

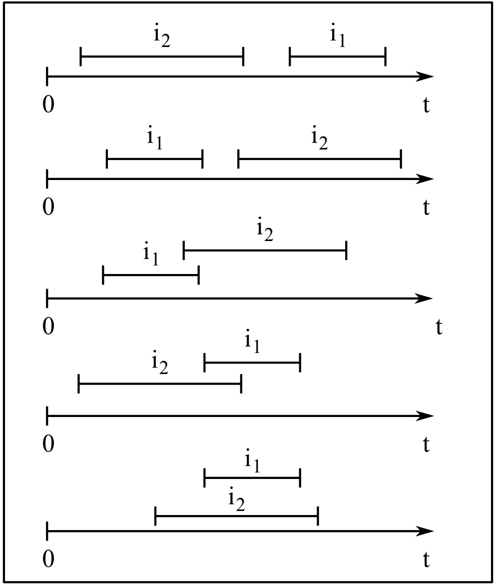
- max / min
- black_height
- height
- sum

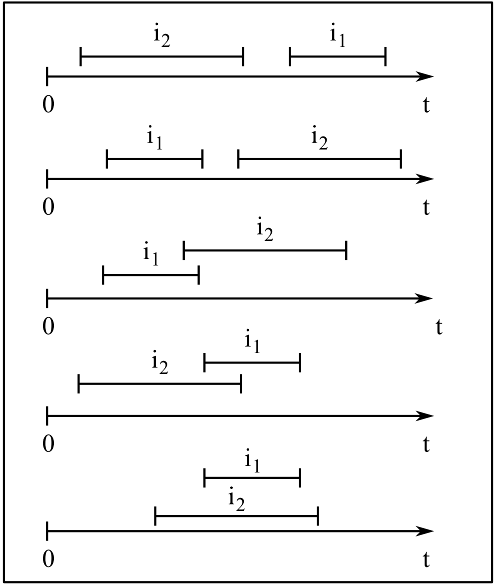
Îmbogățirea cu un câmp

- max / min
- black_height
- height
- sum
- etc.

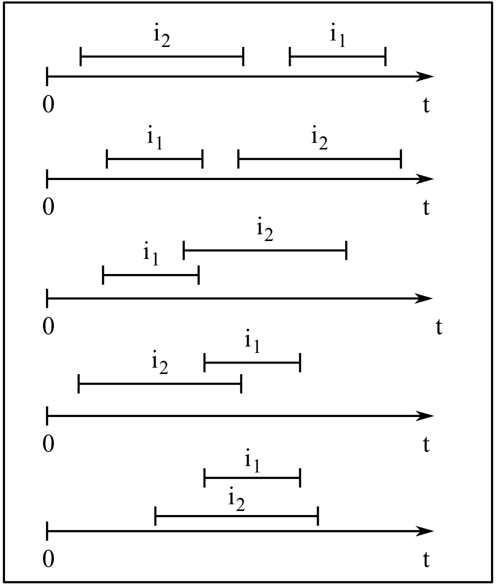
- Arbori care au ca informație un interval $i = [t_1, t_2]$
- Câmpul cheie $x.interval$ care are componentele
  - $interval.low$ $=$ limita stângă a intervalului t_1
  - $interval.high$ - limita dreaptă a intervalului t_2

Arbori pentru intervale - Compararea intervalelor



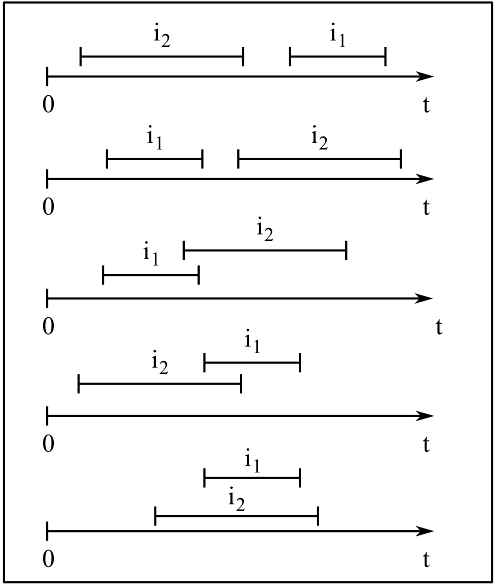


a. Nu se intersectează



a. Nu se intersectează

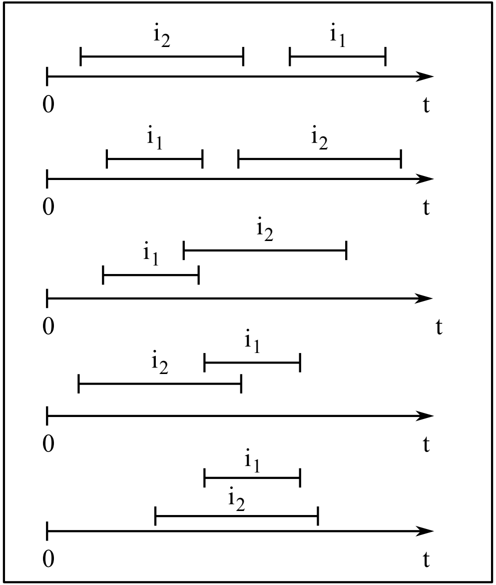
$$\Rightarrow \begin{cases} i_1.high < i_2.low \\ \text{sau} \\ i_2.high < i_1.low \end{cases} .$$



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:

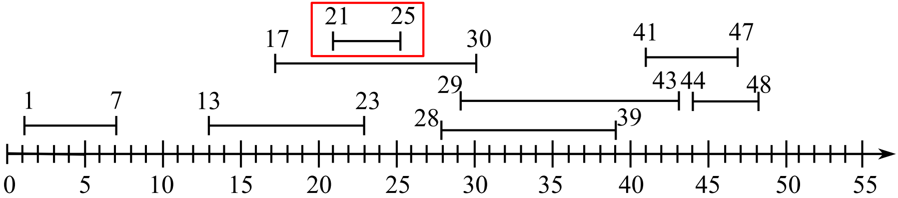


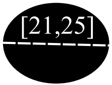
a. Nu se intersectează

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

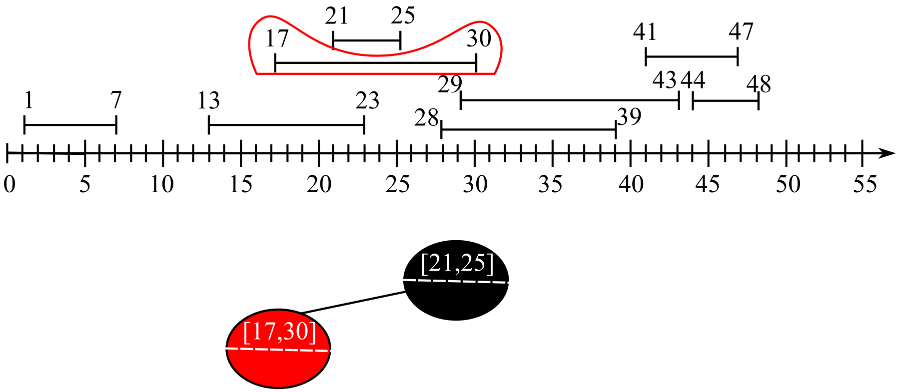
b. Se intersectează: $i_1.low \leq i_2.high$ și $i_2.low \leq i_1.high$

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

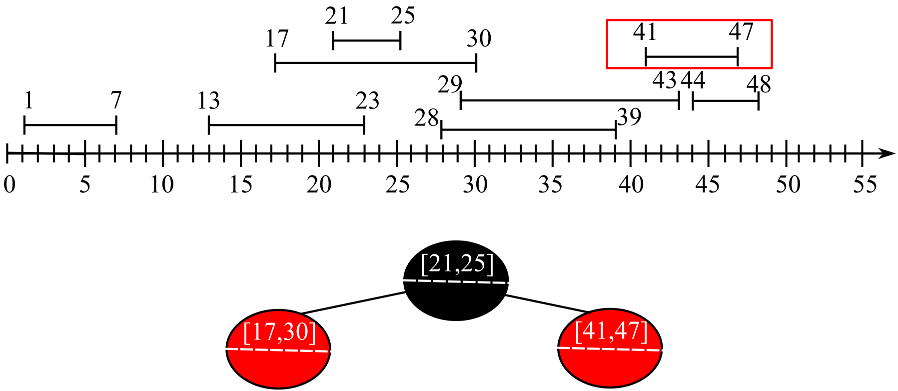




Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este $x.interval.low$

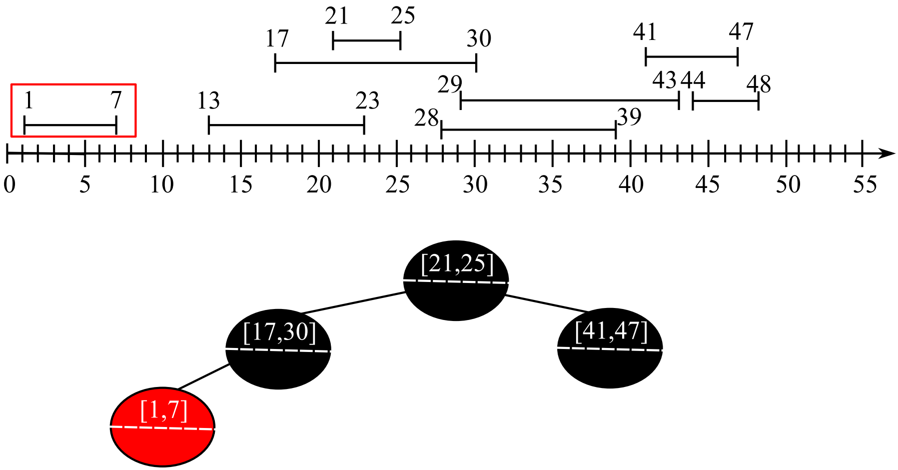


Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este $x.interval.low$



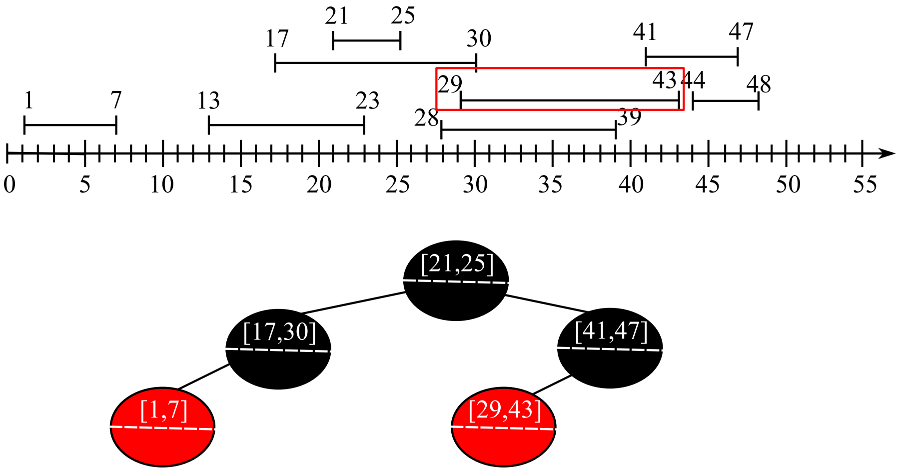
Arbore RN pentru intervale - construcție

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

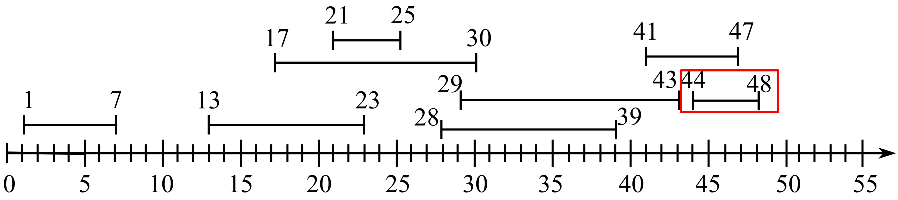


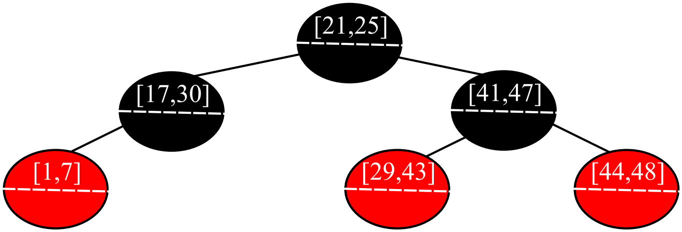
Arbore RN pentru intervale - construcție

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



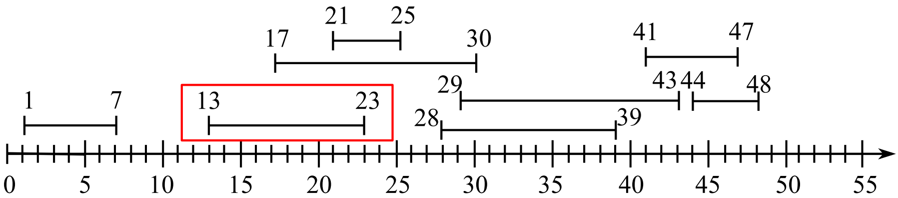
Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este *x.interval.low*

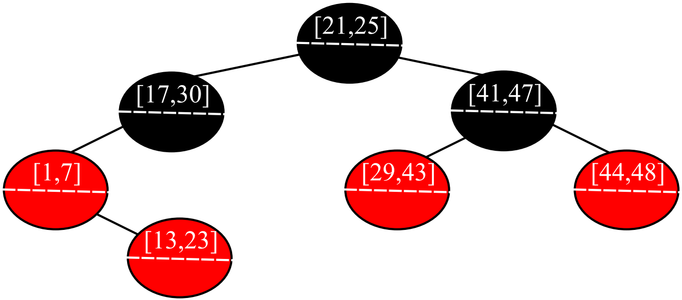




Arbore RN pentru intervale - construcție

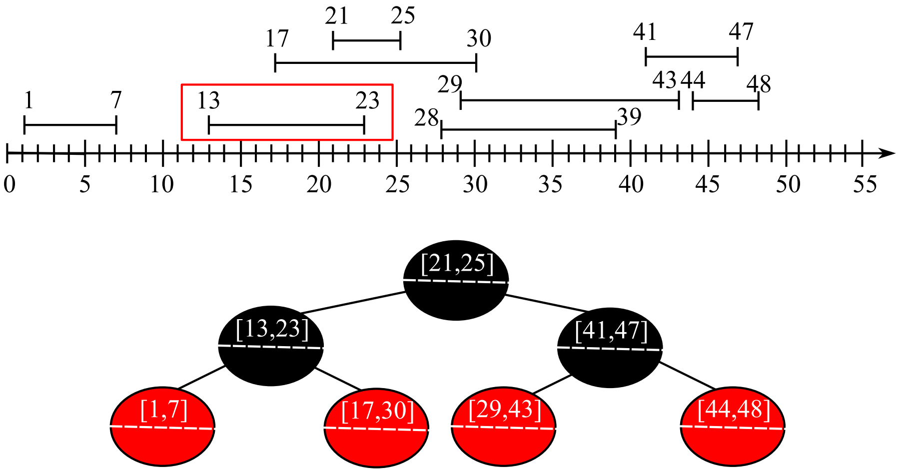
Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este $x.interval.low$





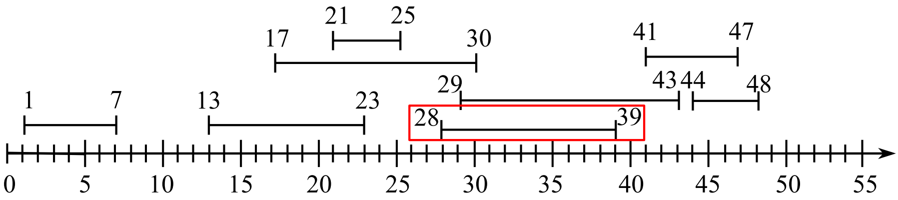
Arbore RN pentru intervale - construcție

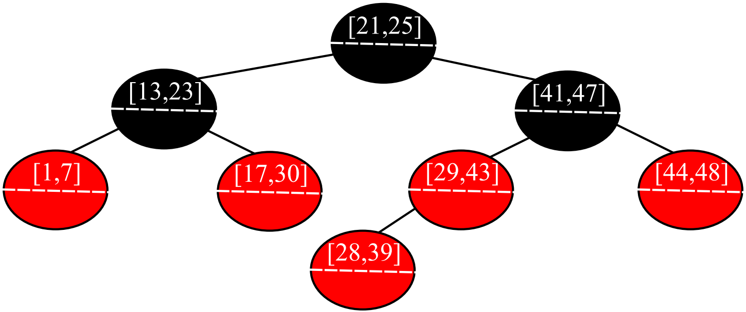
Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este *x.interval.low*



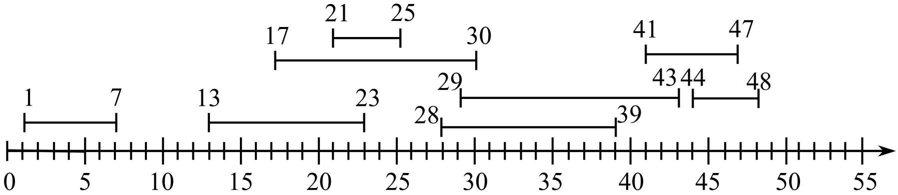
Arbore RN pentru intervale - construcție

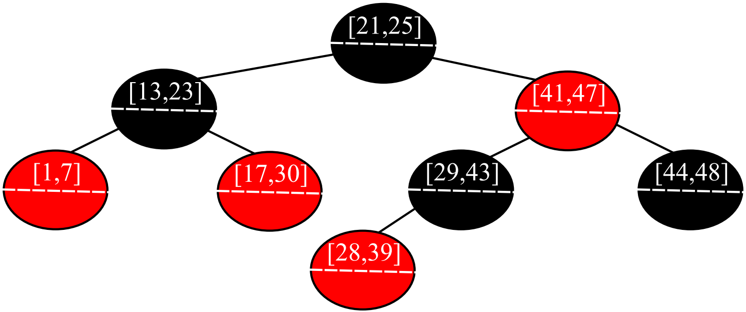
Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este *x.interval.low*



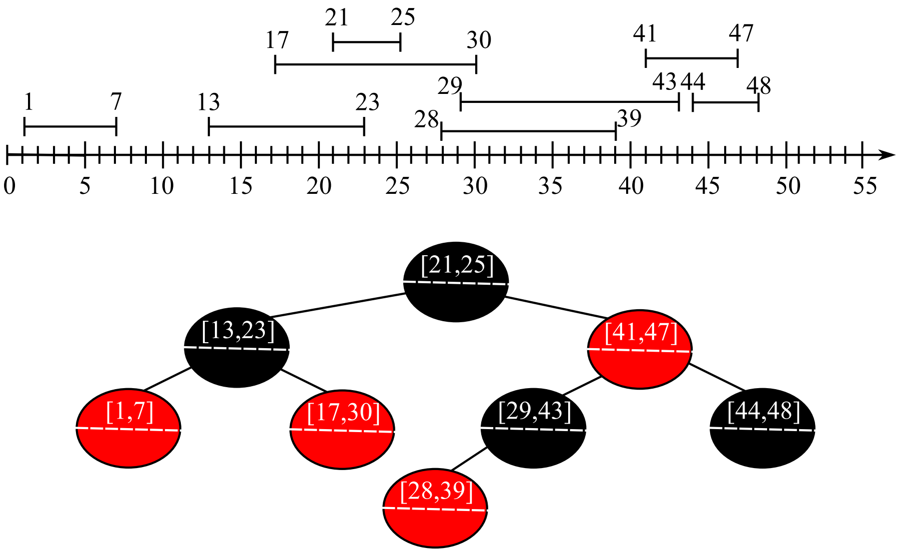


Cheia după care se realizează operațiile de INSERT, SEARCH, DELETE este $x.interval.low$





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

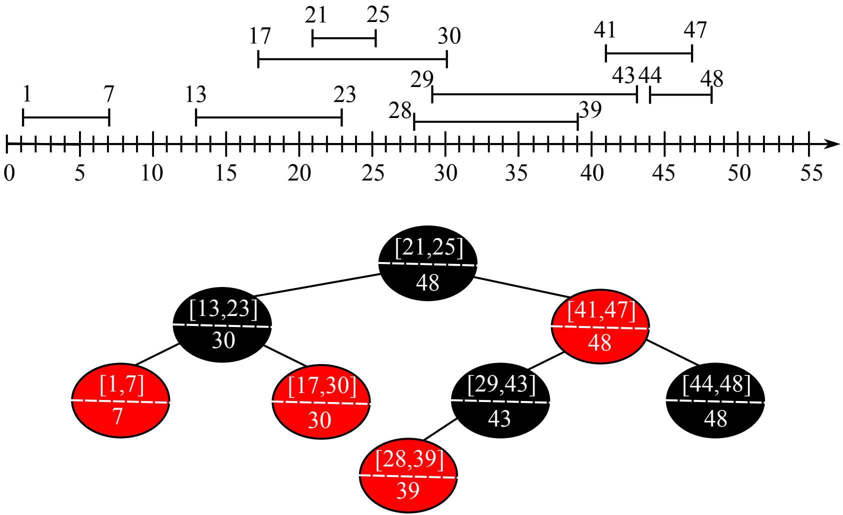
1. Alegerea unei structuri de date - ARN
2. Alegerea informației potrivite pentru îmbogățire - *max_ih*
3. Demonstrarea păstrării complexității operațiilor
    - $x.max\_ih = \max(x.st.max\_ih, x.dr.max\_ih, x.int.high)$
4. Dezvoltarea de noi operații:
    - Căutarea unui interval din arbore, care intersectează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

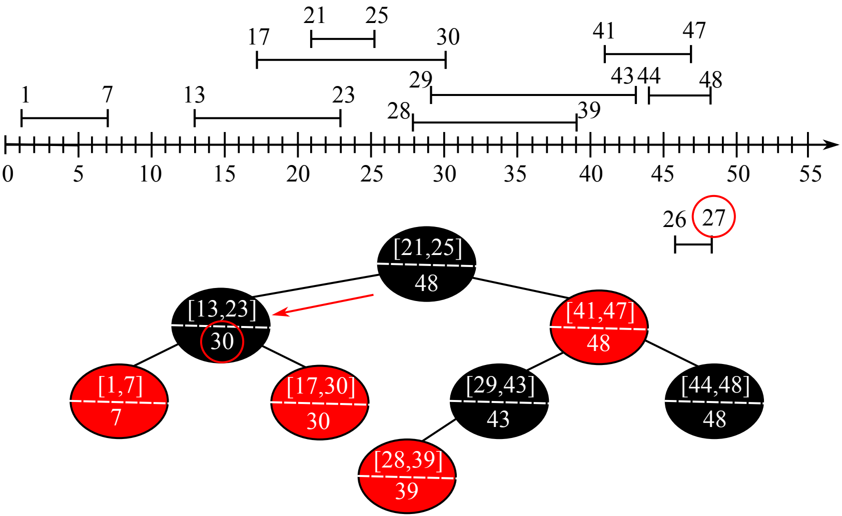
1. Alegerea unei structuri de date - ARN
2. Alegerea informației potrivite pentru îmbogățire - *max_ih*
3. Demonstrarea păstrării complexității operațiilor
    - $x.max\_ih = \max(x.st.max\_ih, x.dr.max\_ih, x.int.high)$
4. Dezvoltarea de noi operații:
    - Căutarea unui interval din arbore, care intersectează 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: $I = [26, 27]$



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



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