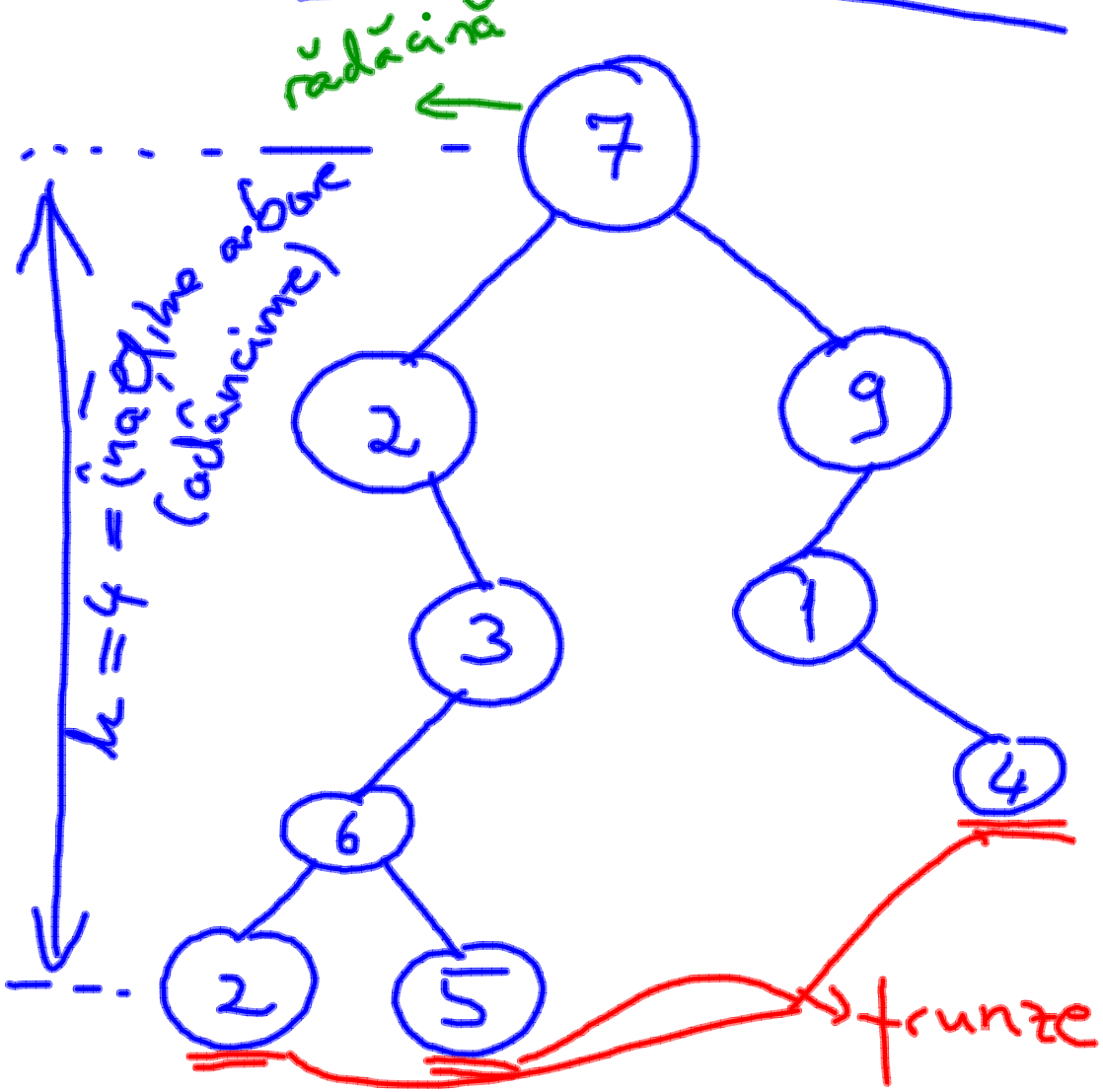
Arbori binari



Parcurgeri arbore

preordine R S D

inordine S R D

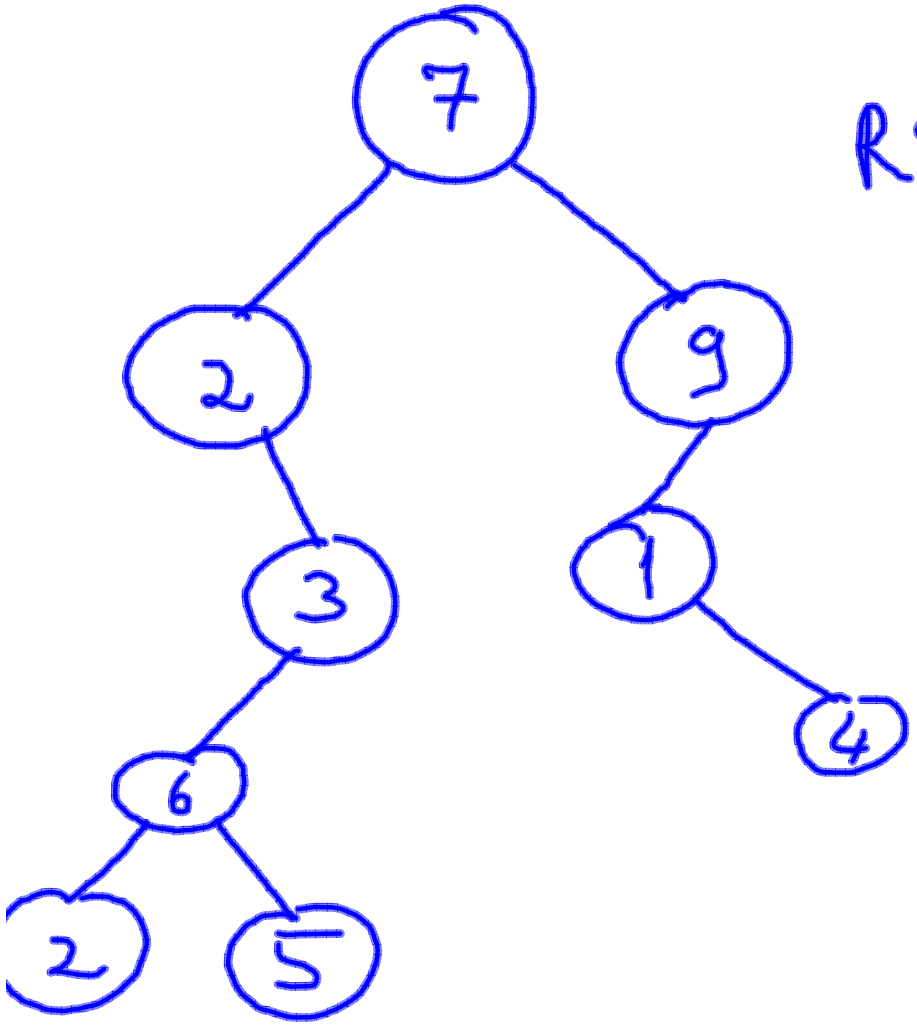
postordine S D R

RSD(a) = { dacă arborele a = vid => NU facem nimic
         { în rest contor { - afișăm rădăcina
                          { - RSD(subarbore stâng)
                          { - RSD(subarbore drept)

Ex:

$RSD(3) = 3\ RSD(\phi)\ RSD(\phi) = 3$

$RSD\left(\begin{array}{ccc} & 3 & \\ 4 & & 2 \end{array}\right) = 3\ RSD(4)\ RSD(2) = 3\ 4\ 2$

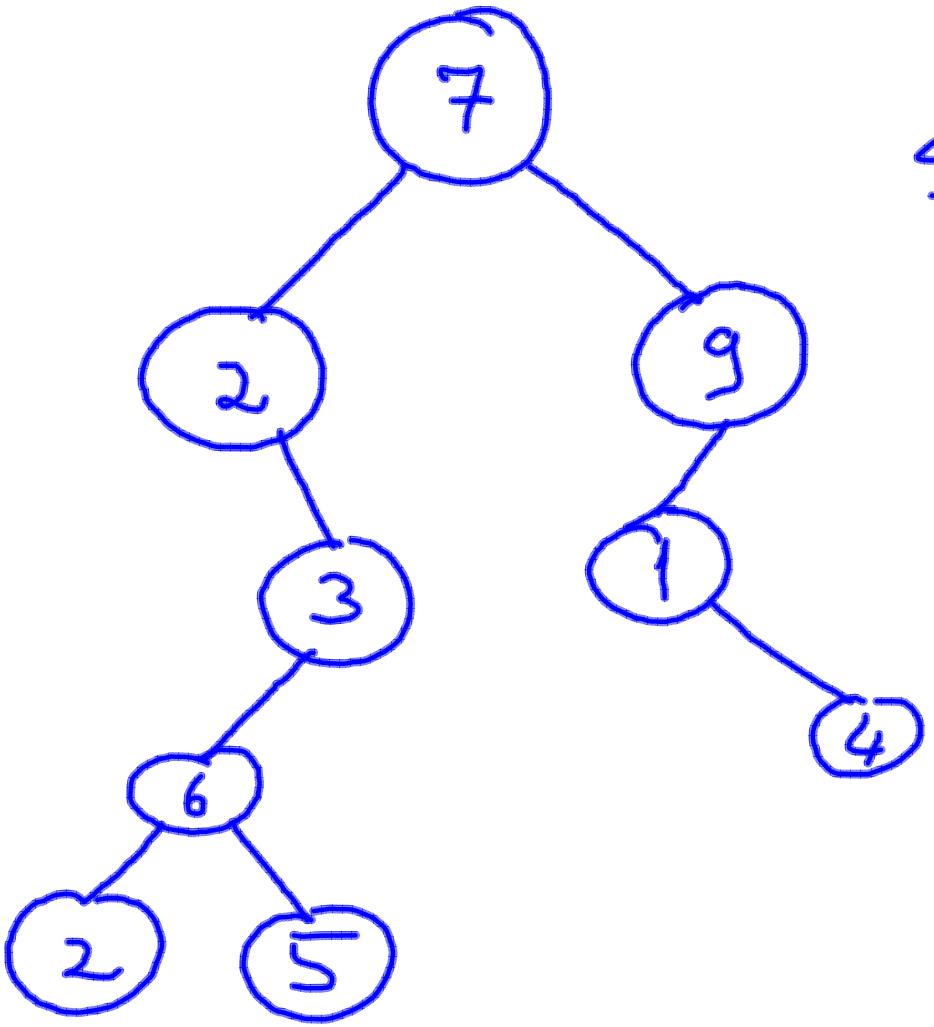


RSD: 7 2 3 6 2 5

9 1 4

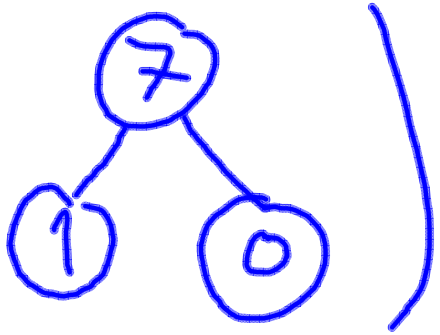
$$SRD(arb) = \begin{cases} \text{dacă } arb.e \text{ nil} - \text{nu facem nimic} \\ \text{dacă e nenil} = \begin{cases} \text{parcurgem stânga} \\ \text{afișăm rădăcina} \\ \text{parcurgem dreapta.} \end{cases} \end{cases}$$

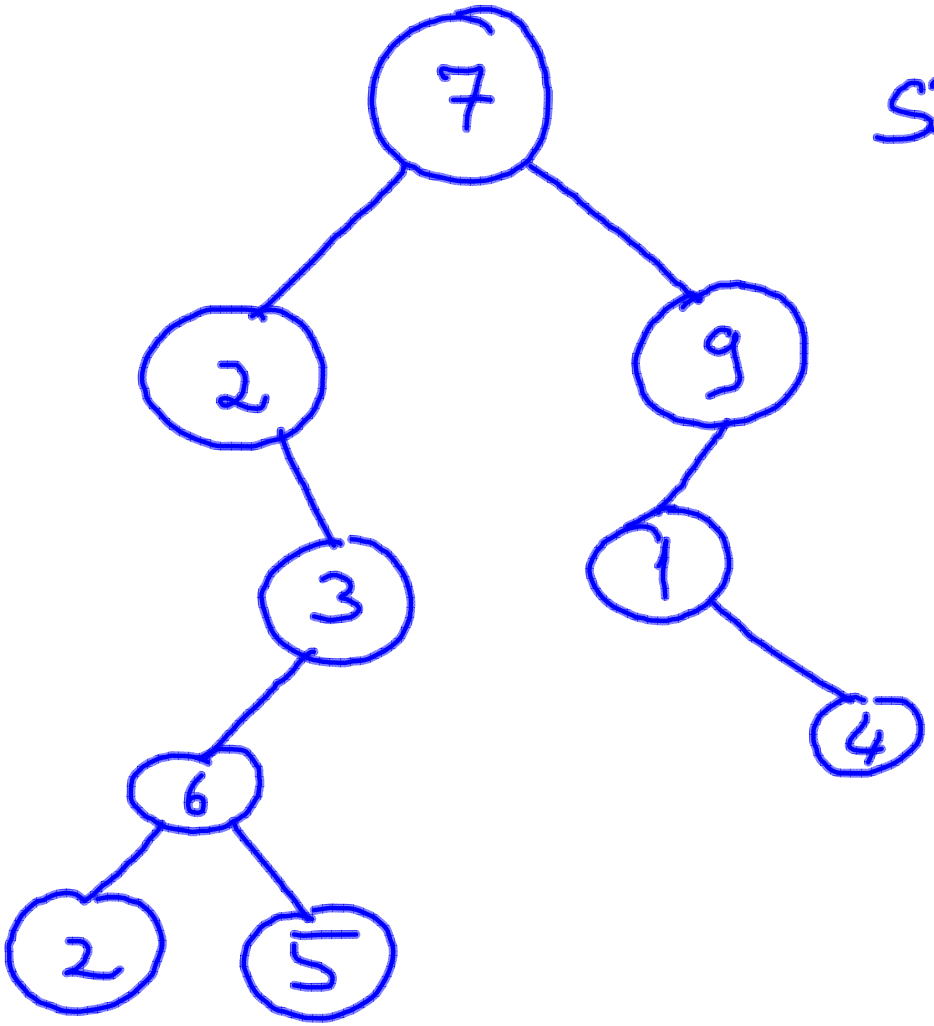
Ex. $SRD\left(\begin{array}{c} 4 \\ 1 \quad 7 \end{array}\right) = 1\,4\,7$



SRD: 2, 2, 6, 5, 3, 7, 1, 4, 9

$$SDR(arb) = \begin{cases} \text{dacă arb e vid nu facem nimic} \\ \text{în caz contrar} \begin{cases} SDR(\text{subarb. stâng}) \\ SDR(\text{subarb. drept}) \\ \text{afișăm rădăcina} \end{cases} \end{cases}$$

$SDR\Big($  $\Big) =$ (1) (0) (7)



SDR : 2, 5, 6, 3, 2
4, 1, 9, 7

Implementare arbori

- cu vectori obișnuiți (statici), și anume 3 vectori: inf, st, dr în care, la un anumit indice k sunt reținute informații ale unui nod:

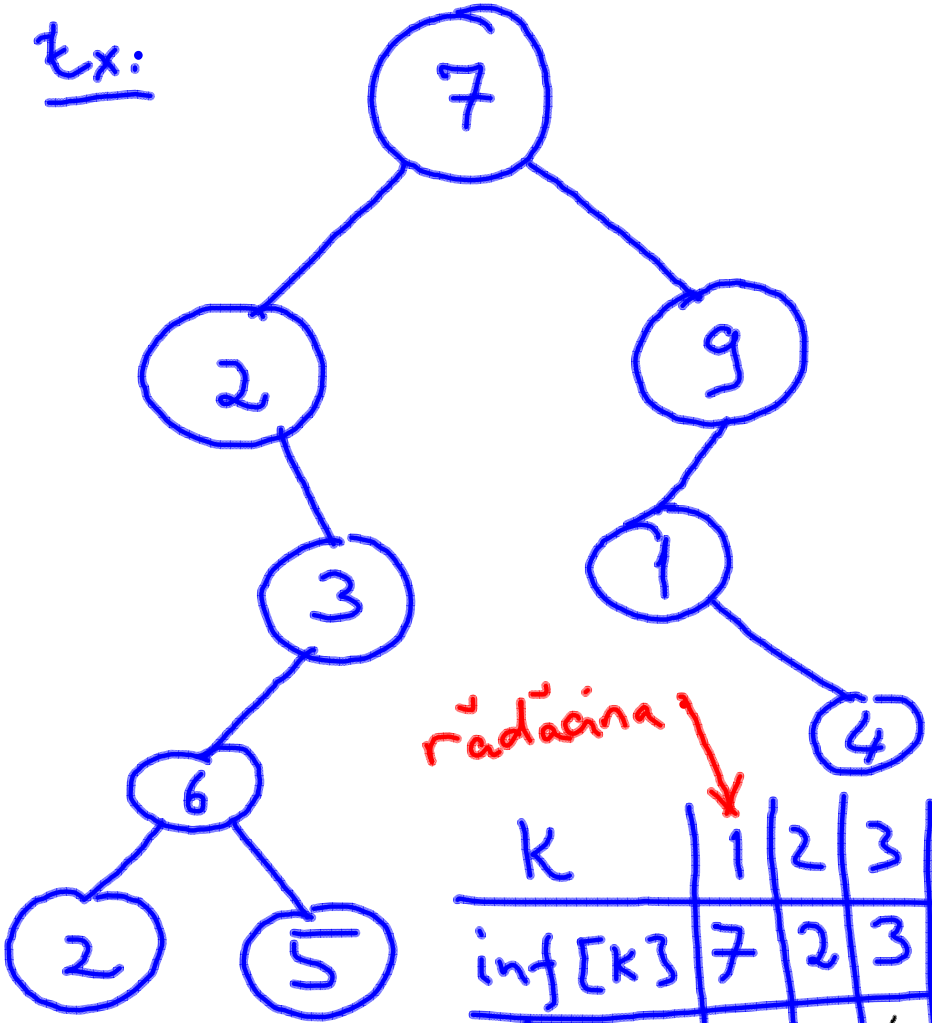
  inf[k] = valoarea cheii nodului

  st[k] = indicele la care e memorat nodul din stânga. Dacă ∄ - memorăm -1

  dr[k] = analog - nodul din dreapta.

Evid. mai trebuie să știm la ce indice e memorată rădăcina.

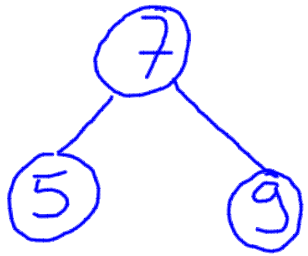
Ex:



rădăcina

k	1	2	3	4	5	6	7	8	9
inf[k]	7	2	3	6	2	5	9	1	4
st[k]	2	-1	4	5	-1	-1	8	-1	-1
dr[k]	7	3	-1	6	-1	-1	-1	9	-1

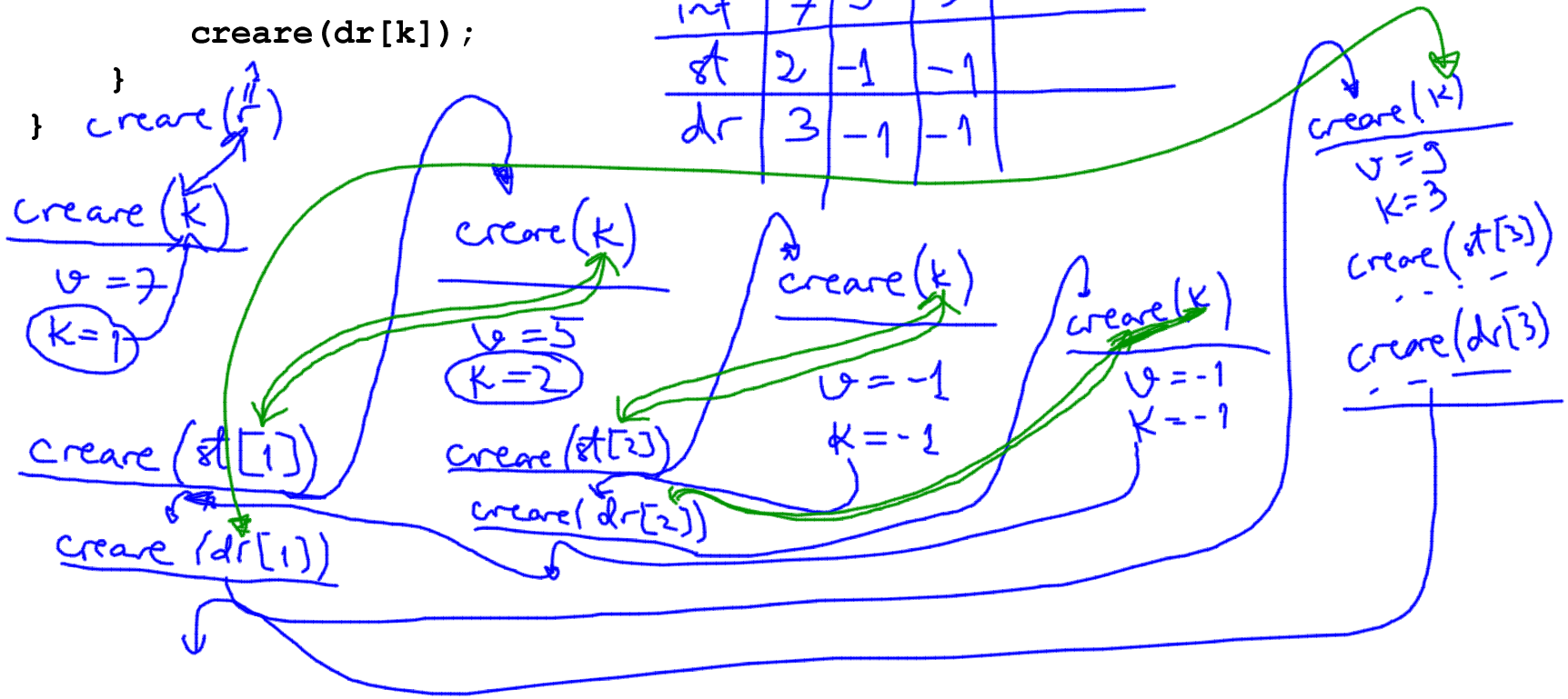
```
void creare(int &k)
{
     fin>>v;
     if(v==-1)
              k=-1;
     else
     {
          ++n;
          k=n;
          inf[k]=v;
          creare(st[k]);
          creare(dr[k]);
     }
}
```



$n = \cancel{0}\ \cancel{1}\ \cancel{2}\ 3$

7 5 -1 -1 9 -1 -1

k	1	2	3
inf	7	5	9
st	2	-1	-1
dr	3	-1	-1



Arbori binari de căutare

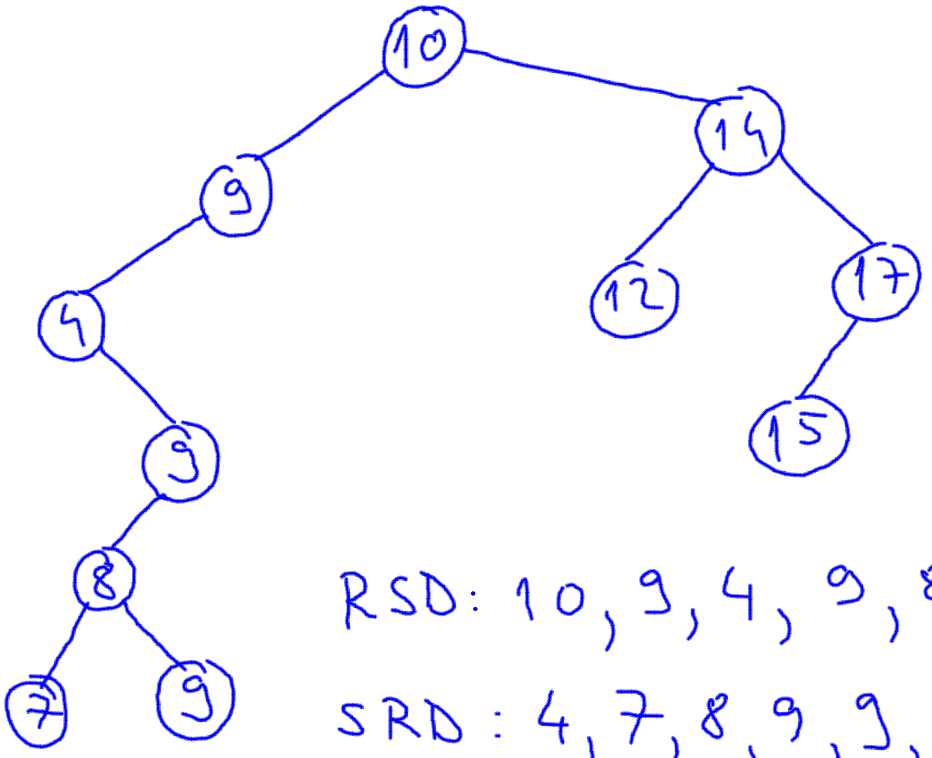
Este un arbore binar în care rădăcina oricărui subarbore este mai mare sau egală cu rădăcina subarborelui stâng și mai mică sau egală decât rădăcina subarborelui drept.

Construirea unui astfel de arbore plecând de la valori se face astfel:

- se pornește din rădăcină și se compară valoarea de inserat cu valoarea nodului curent. Dacă este ≤ se merge pe stânga, dacă nu, pe dreapta.

În primul loc găsit liber se inserează valoarea dorită.

Ex.: Să generăm arb. binar de căutare cu urm. valori: 10, 14, 9, 4, 9, 8, 12, 17, 7, 15



RSD: 10, 9, 4, 9, 8, 7, 9, 14, 12, 17, 15

SRD: 4, 7, 8, 9, 9, 9, 10, 12, 14, 15, 17

SDR: 7, 9, 8, 9, 4, 9, 12, 15, 17, 14, 10