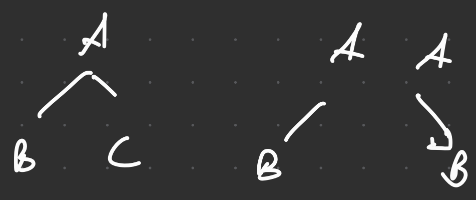
Arborii Binari



-> maxim 2 copii

-> subarbore stâng / drept

Transf. A Gen -> A Bin

prim fiu, frate dreapta

Implementare

Index	
Data	
Stanga	
dreapta	

Nr. max de noduri de pe nivelul i -> 2^{i-1}

Nr. max noduri : $\sum_{i=1}^{h} 2^{i-1} \quad -1$ $\quad h > 0$

-> arbore plin de înălțime h

Arbore binar complet

Parintele nodului i are indicele i/2

Fiul stâng al lui i -> 2 * i
drept 2 * i + 1

2 * i + 1 > n
-> nu are fii

Implementare cu pointeri

```
struct {
    val
    * st
    * dr
}
```

Parcurgeri -> RSD
SRD
SDR

Threaded Trees

```
void pre (AB * r) {
    if r != NULL
        print r -> cheie
        pre( r -> st)
        pre( r -> dr
}
```

-> contine ~~pointer~~ pointer la părinte

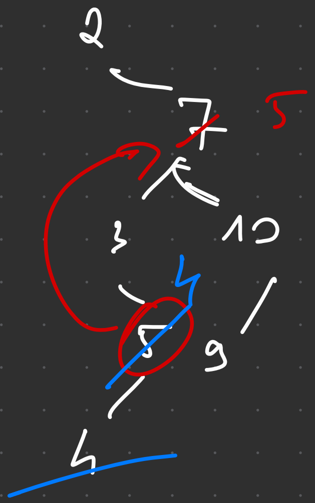
Arbori Binari Ordonati

$S \le R \le D$ -> trebuie balansat pt căutări în $\log_2 n$

- Inserția -> < root -> stânga
  > root -> dreapta

```
AB * insert (AB * r, int c)
    if r! NULL        if c <= r -> cheie
                          insert (r -> st, c)
                      else insert (r -> dr, c)
    else r -> malloc  r -> cheie = c   r -> st
    ret r
```

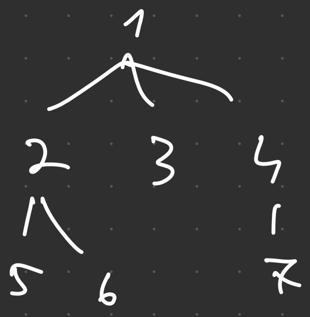
Suprimare



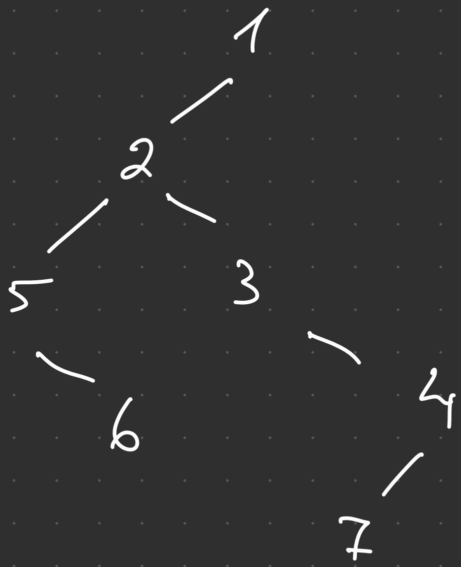
ultimul nod de dinainte la parcurgerea în inordine

SRD

FCRS



B



SRD

5 2 6 1 3 7 4

5 6 2 3 7 4 1