

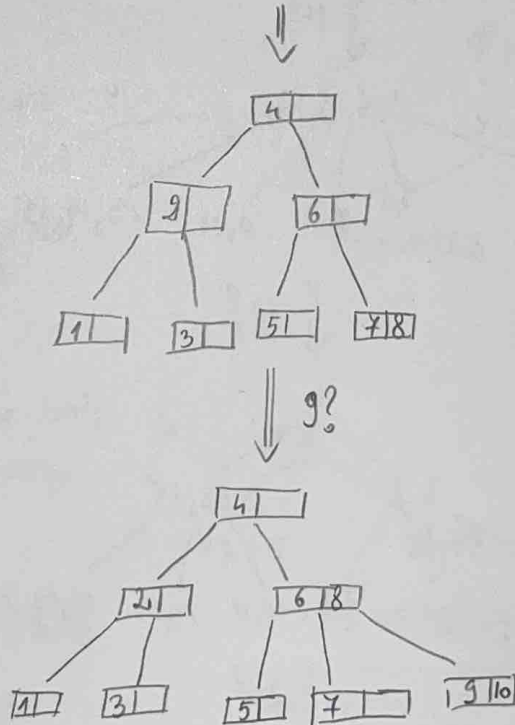
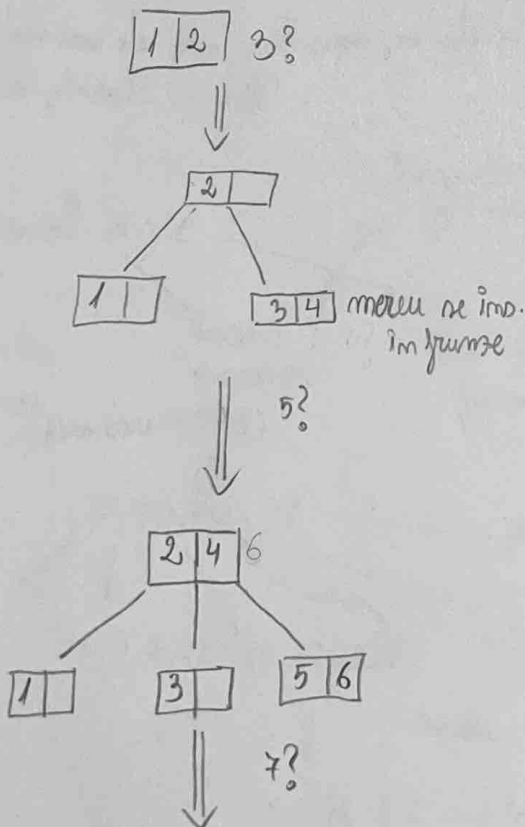
B-TREES

- balanced m-way tree
- generalizes a binary search tree
- can have more than one key and more than 2 children
- maintains sorted data
- toate frunzele sunt la același nivel!
- B-tree of order m has following properties
 - * every node has maximum m children
 - * min children: leaf: 0
 - root: 2
 - internal nodes: $\lceil \frac{m}{2} \rceil$ veri ce ați înțeles

$$\lceil \frac{5}{2} \rceil = \lceil 2.5 \rceil = 3$$

- * every node has max (m-1) keys
- * min keys: root node $\rightarrow 1$
- all other nodes $\rightarrow \lceil \frac{m}{2} \rceil - 1$

inserare simplă în B-tree m=3



split mereu middle up
incerc mereu să ure în părinte

→ create a B-tree of order 5 by inserting values from 1 to 20

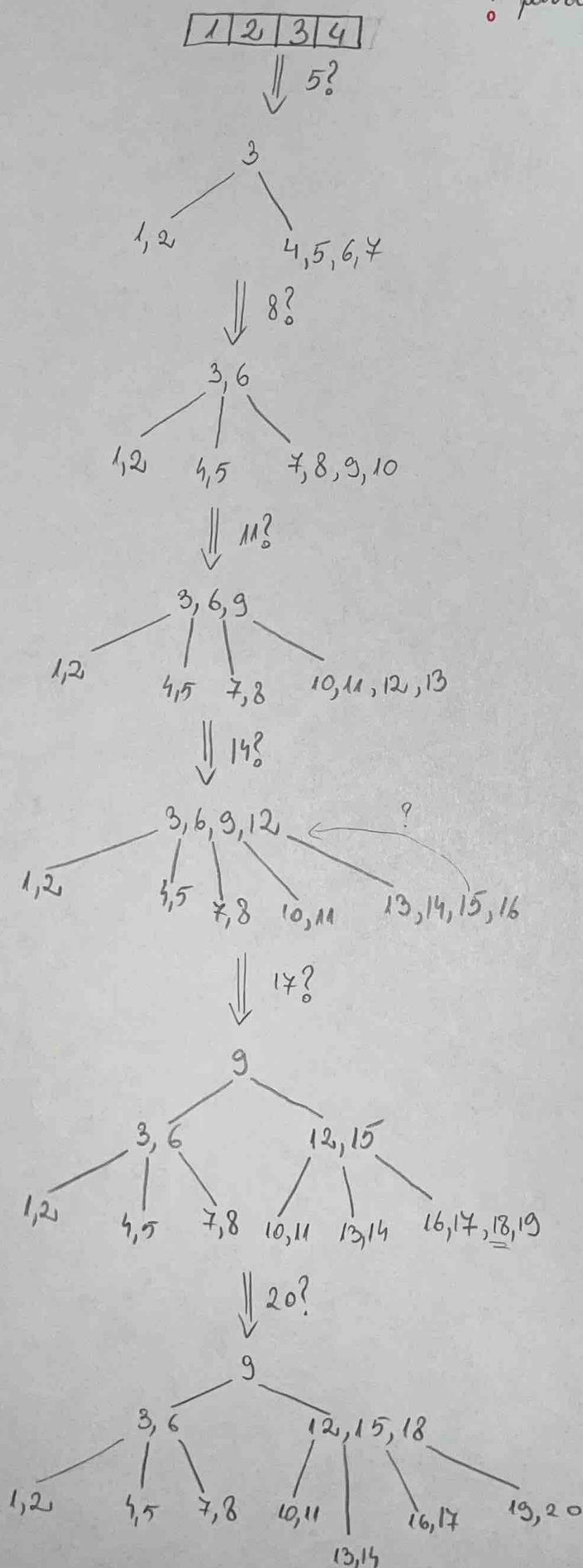
max ch: $m = 5$

max keys: $m-1 = 4$

? dacă trebuie să aleg middle element și e nr
par de elemente pe care-l iau?

3 5 9 12 \Rightarrow luăm oricare \Rightarrow

se numesc left biased și right biased



basically modul plin se desparte în

1. middle

2. ce-i înainte de middle

3. ce-i după middle

DELETE FROM B-TREE

① e în frunză

leaf node → contains more than min no of keys (store direct)
contains min no of keys

order (m) = 5

min children = $\lceil \frac{m}{2} \rceil = 3$

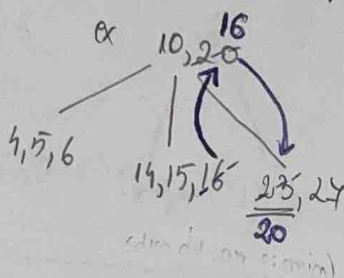
max children = 5

min keys = $2(\lceil \frac{m}{2} \rceil - 1)$

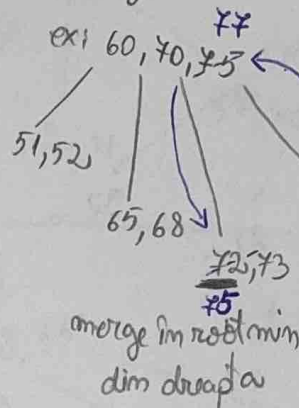
max keys = 4

borrow from left sibling

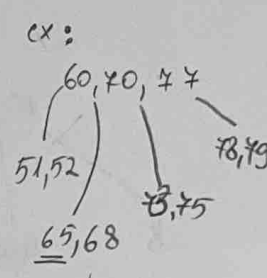
merge in root (max direct)



borrow from right sibling



merge cu left or right

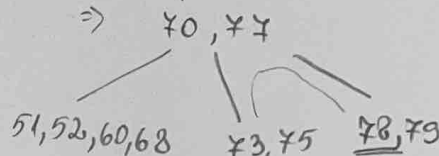


* dar dacă m-au mici frați mici părinți?

păi întrebă și el la frați, apoi la părinți
↓
părinți

* m-o să am ca de prea mulți copii, că atunci i-ar fi putut împrumuta un frate

! și the parent (cu 60 aici și partea dreaptă)



② internal node

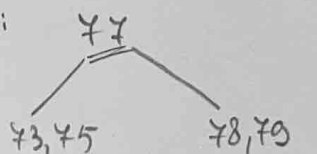
al mai dăruș din copil drept

inorder predecessor (evidențiat că merge)

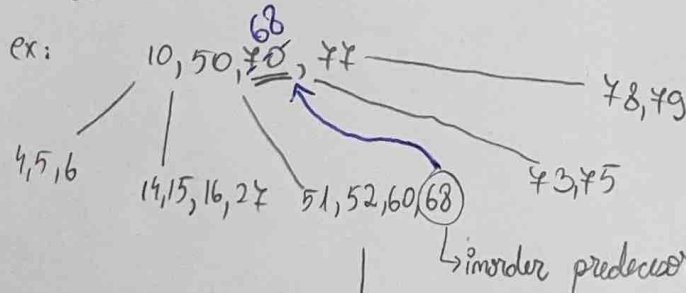
inorder successor

dacă nu se poate facem merge

ex:



⇒ 73, 75, 78, 79

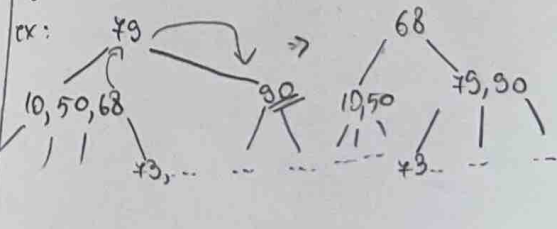


verify if it contain > min no keys

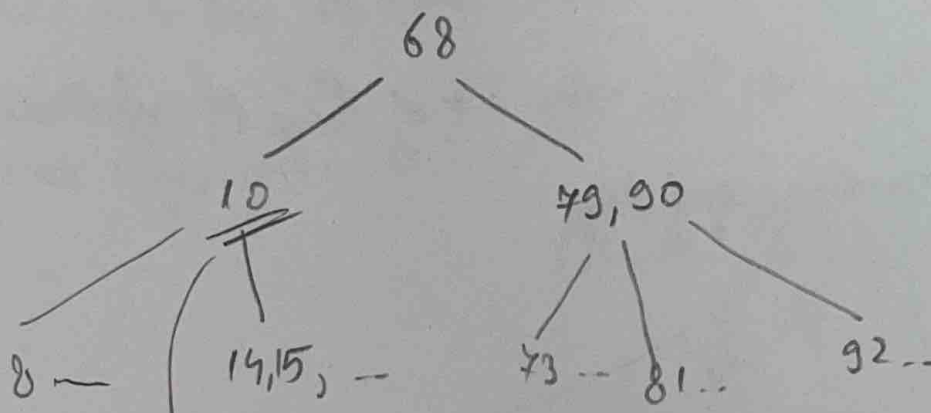
al mai dreapta din copil stang

dacă rămân cu < no of keys în părinte
⇒ cere la frate → schimb root

! CU TOT CU COPII



alt exemplu:



am sters de aici -> merge cu radacina

