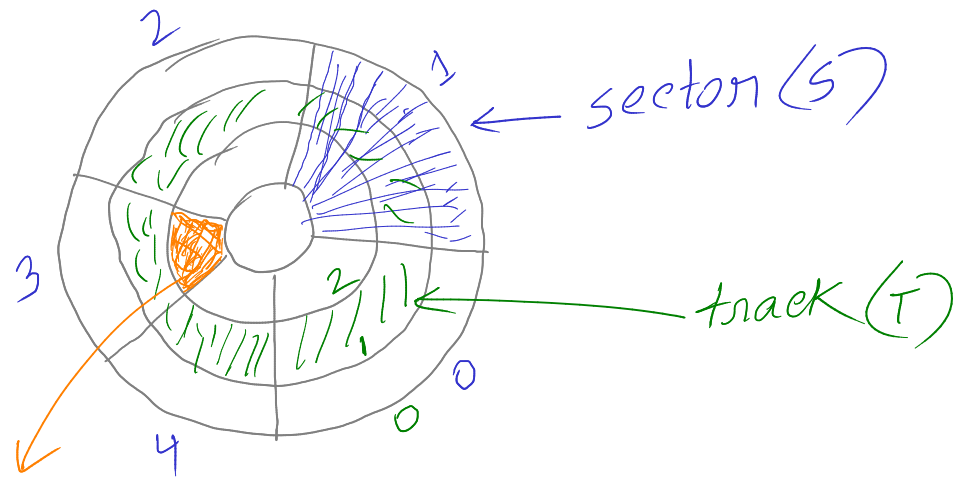
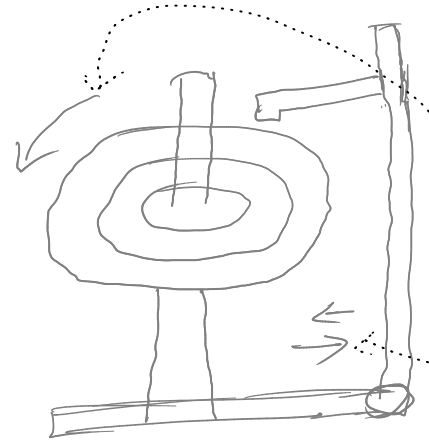
Disk Structure

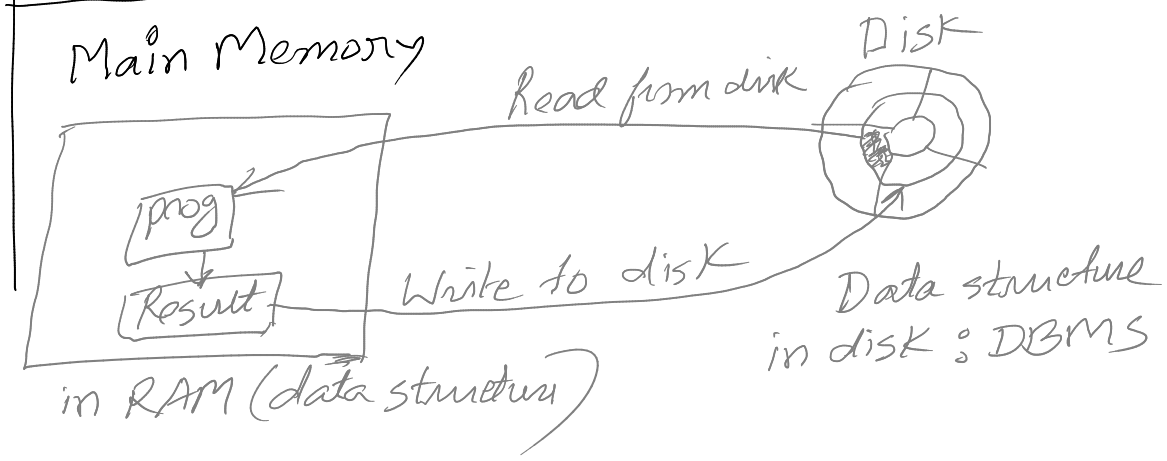


Block (512 bytes)
intersection of
T and S
(i.e 2, 3)

Actual position is determined by
(T+S) + offset (displacement relative



T, S can be
changed by
spinning or/and
movement.

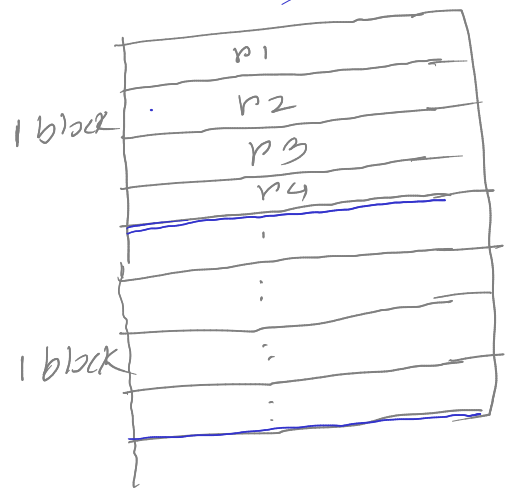


□ How data (i.e records) are stored on disk —

Example: Suppose we have: Students (SID, Name, Dept, Section, Address)

SID = 10 bytes
Name = 50
Dept = 10
Section = 8
Address = 50

Total = 128 bytes (each record)



Suppose, we have 100 records.

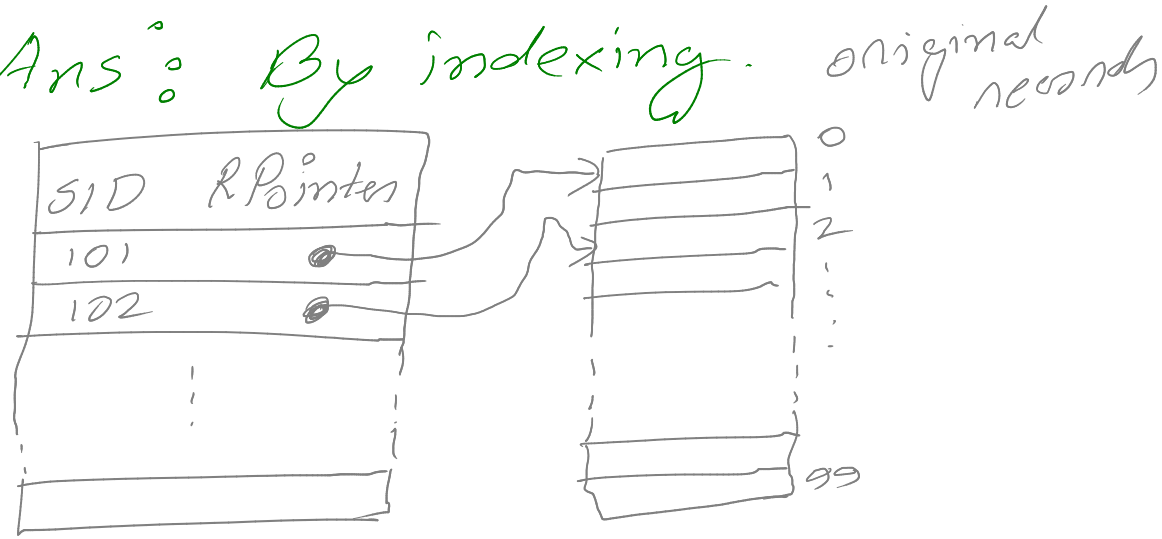
No of records per block = $\frac{512}{128} = 4$

So, For 100 records we need = $100/4 = 25$ blocks

Now, "select Dept from students where SID = 201" we must search 25 blocks since we don't know the exact location. Can we reduce the search?

Can we reduce the search?

Ans: Yes, How? Ans: By indexing.



Space: SID = 10 Bytes
Rpointer = 6 Bytes
16 bytes

No of entries per block = $\frac{512}{16}$

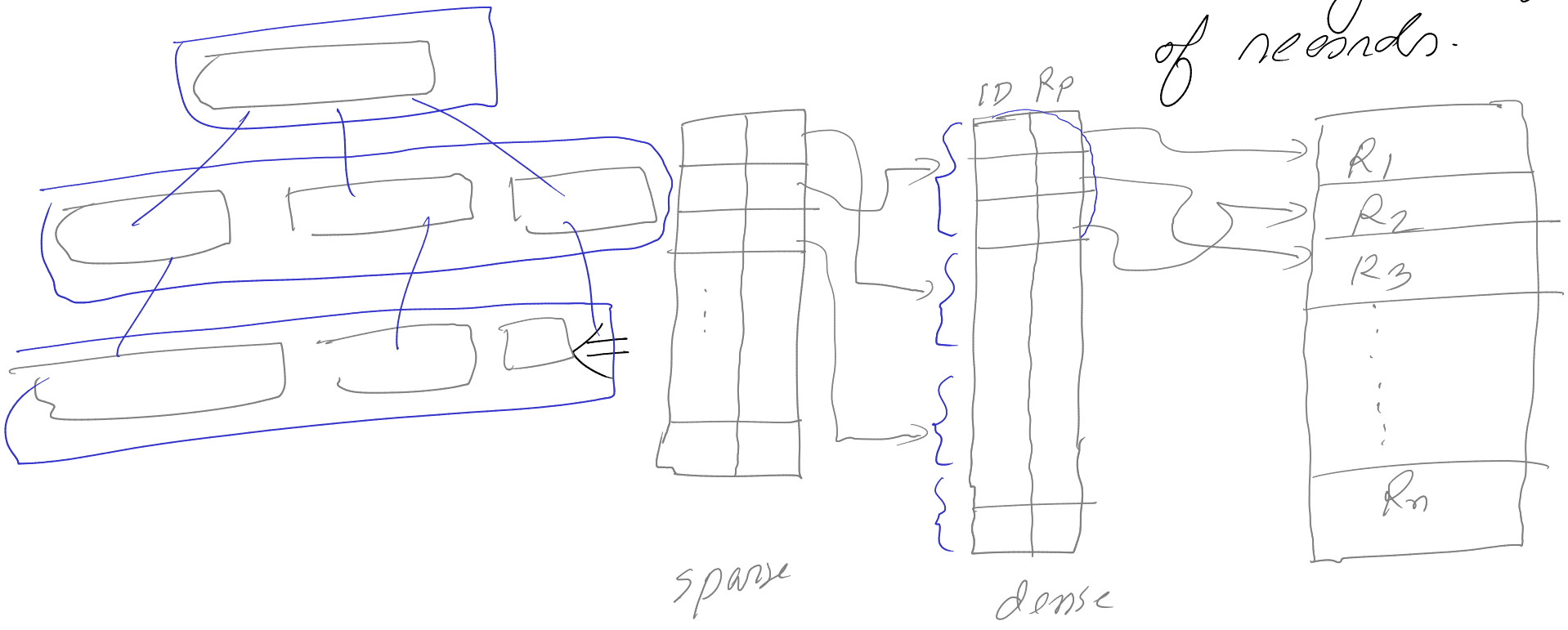
$= 32$

For 100 records we need = $\frac{100}{32} = 3.2 \simeq 4$ blocks

4 blocks for reading index
1 block " actual record
5 blocks for one "select..."

So, Without indexing we need = 25 blocks read
With indexing we need = 5 blocks "

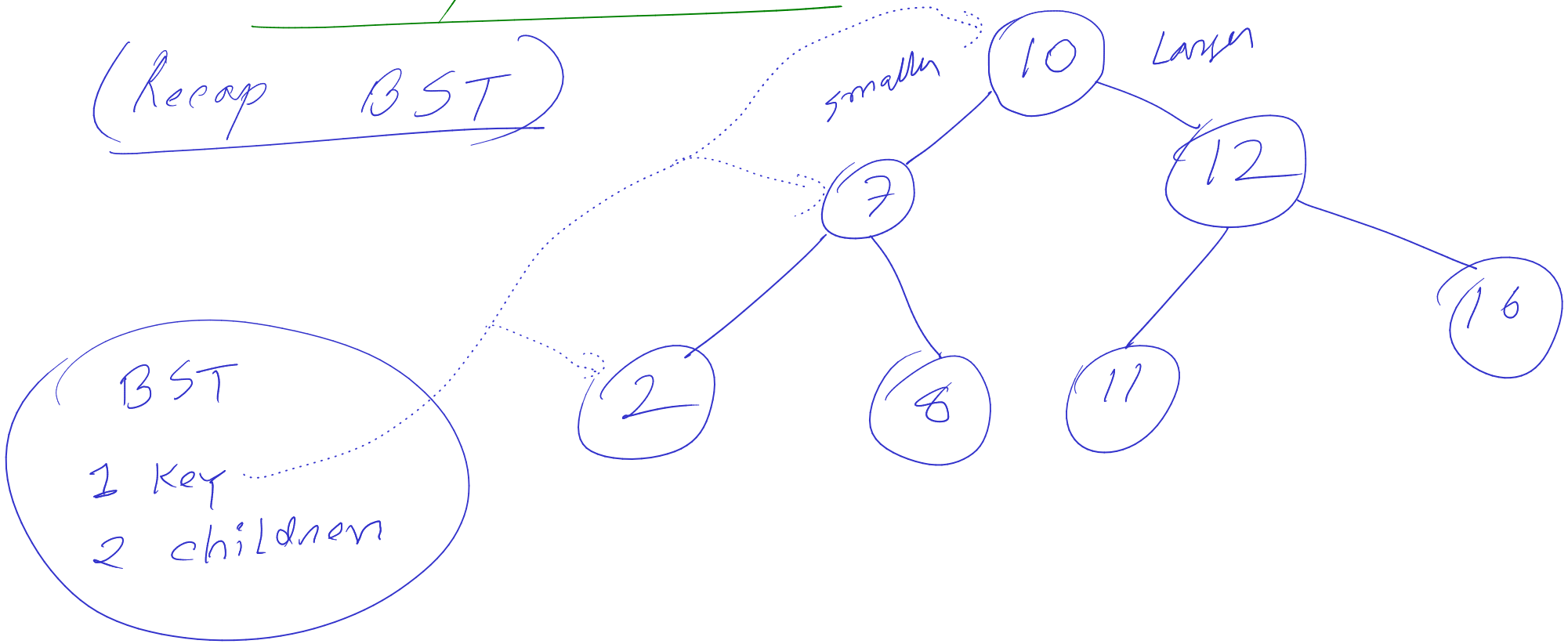
- Multi-level Indexing: Simple index size will grow unmanageable for millions of records.



Now, re-orient this structure, a tree will be formed.

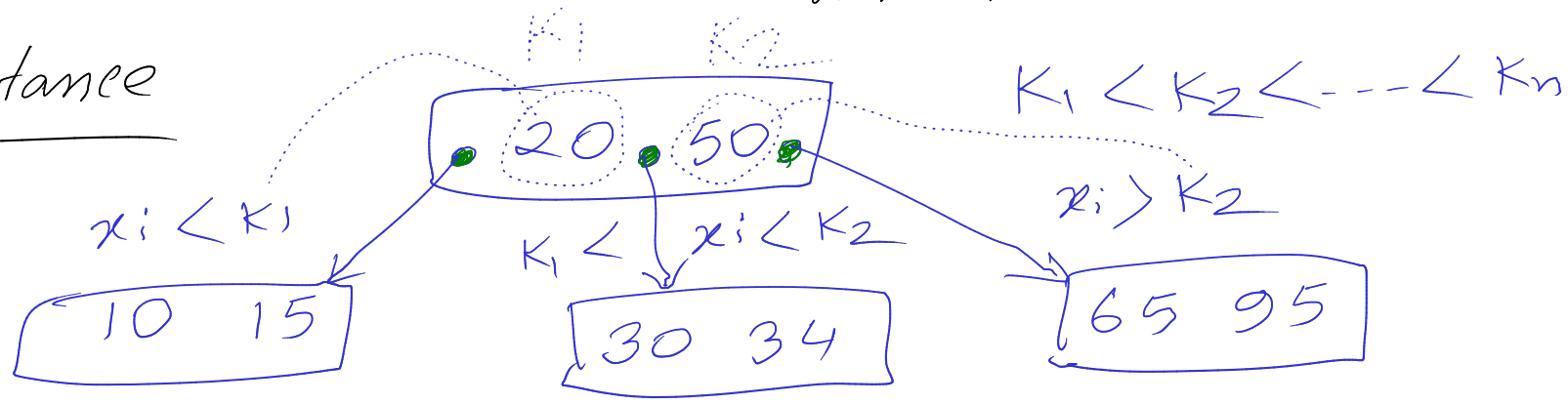
m-way search tree

(Recap BST)



In m-way search tree it may have more than 1 key and more than 2 children

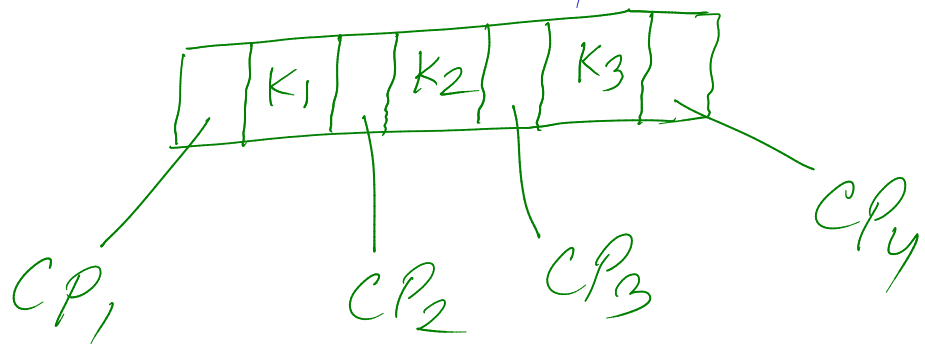
For instance



m-way search tree, here 2 keys
3 children

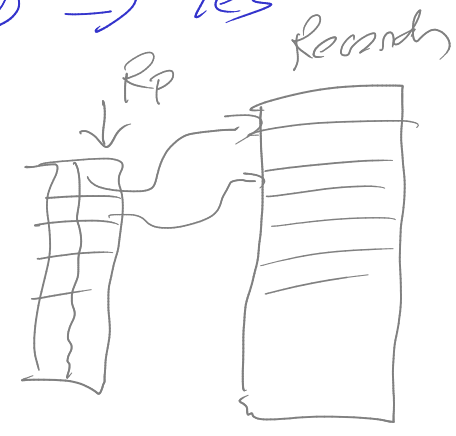
In general, m-way search tree
⇒ (m-1) no. of keys

- Lets see how 4-way ST looks (4-1=3 keys)

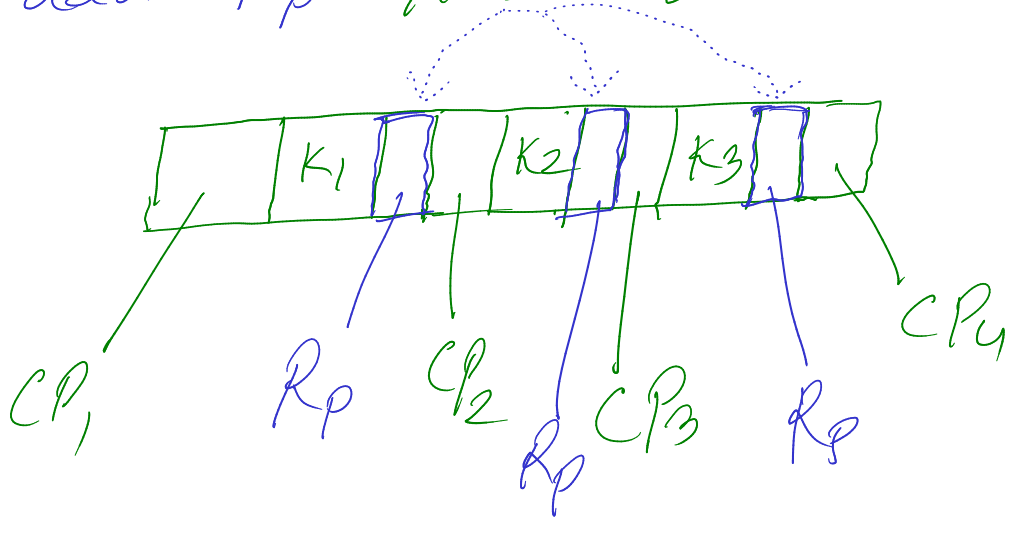


CP = child pointer

- Can we use m-way ST to store indexes ⇒ Yes
  - Need to add Rp Record Pointer

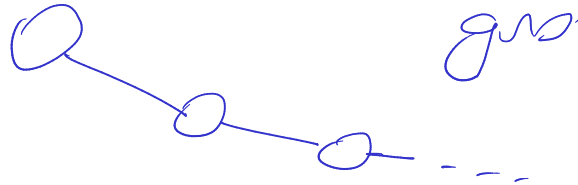


Node structure:



Store: 3 keys
4 childpointers
3 Recordpointers

- But in m-way search tree the insertion is not under any control i.e.



grows in one direction. (Not a balanced tree)

- So, B Tree = m-way ST + Rules

Rules are —

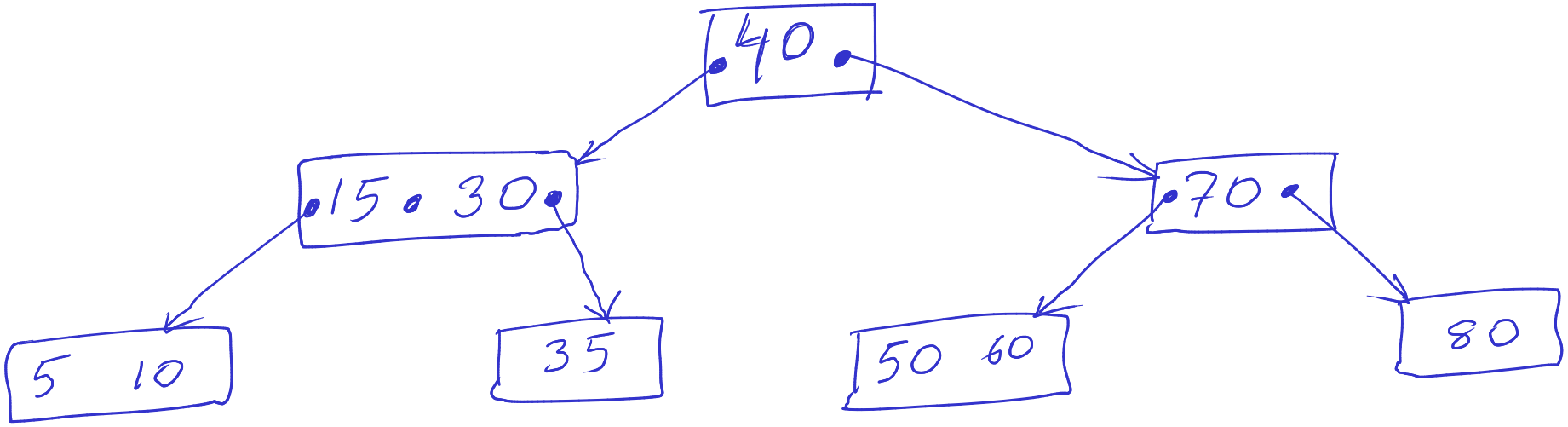
1. $\lceil \frac{m}{2} \rceil$ children (for other nodes)
2. Root can have min 2 children
3. All leaves are at the same level (Balanced)
4. Bottom-up process (construction)

☐ Step by step demo: construction of a m-way ST.

$m = 4$

$K = 3$

(max 3 Key)



For B-tree each key will have a second pointer (Rp)