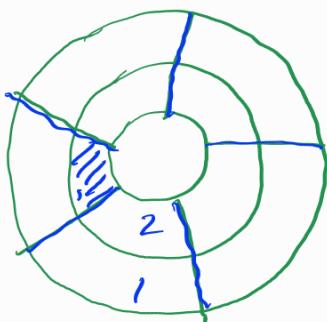


B Tree

Disk Structure



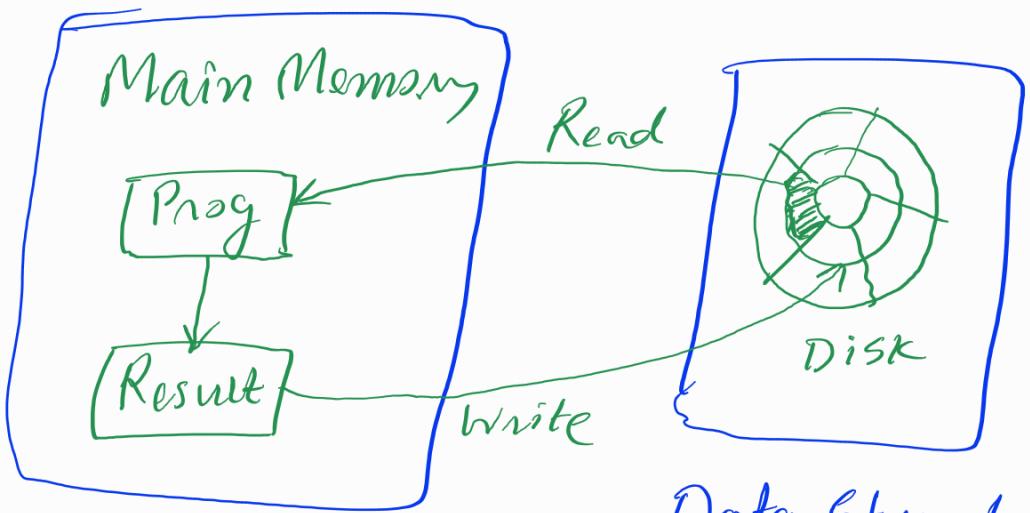
sector S

track T

$(T+S) + \text{offset}$

Block

512 bytes



Data Structure
in Disk
(i.e DBMS)

RAM
Data Structure (in RAM)

Q How records are stored? (on Disk)

Students (ID, Name, Dept, Sec.
Address)

ID = 10 bytes

Name = 50 "

Dept = 10 "

Sec. = 8

Address = 50

Total: 128 bytes (each Record)

We've 100 records.

$$\text{No of records in each Block} = \frac{512}{128} \approx 4$$

$$\text{So, 100 records, No of Blocks} = \frac{100}{4}$$

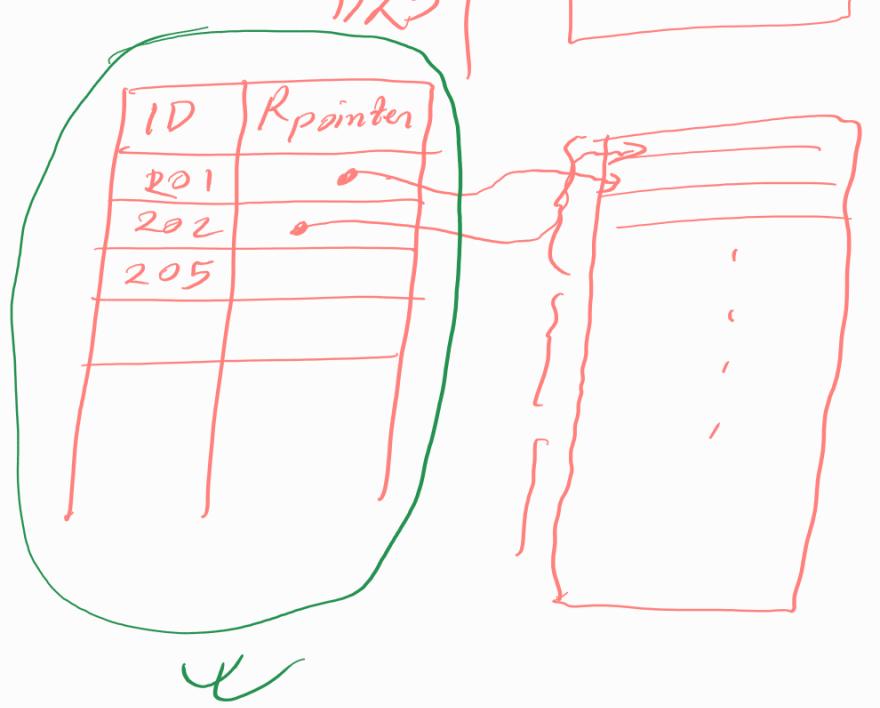
$$= \underline{\underline{25 \text{ blocks}}}$$

Select Dept from students
where $ID = 201$;

All 25 blocks Read

Can we reduce it? Yes
How? By indexing

$$\begin{aligned} ID &= 10 \text{ bytes} \\ R_p &= 6 \text{ bytes} \\ \text{Total} &: 16 \text{ bytes} \end{aligned}$$

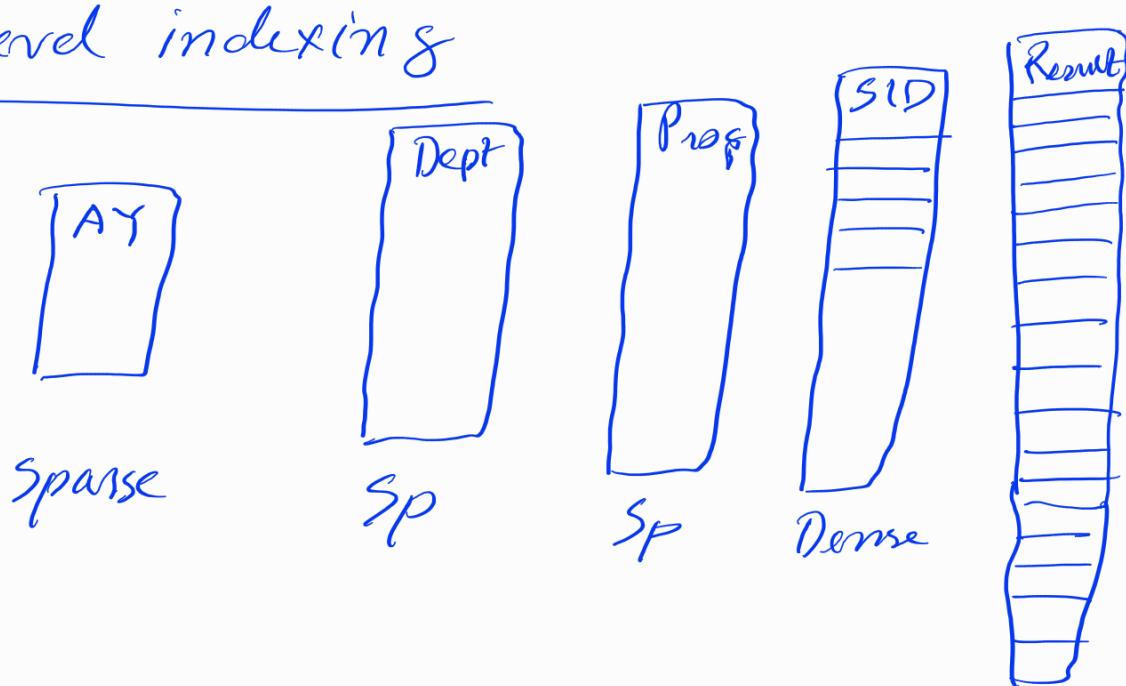


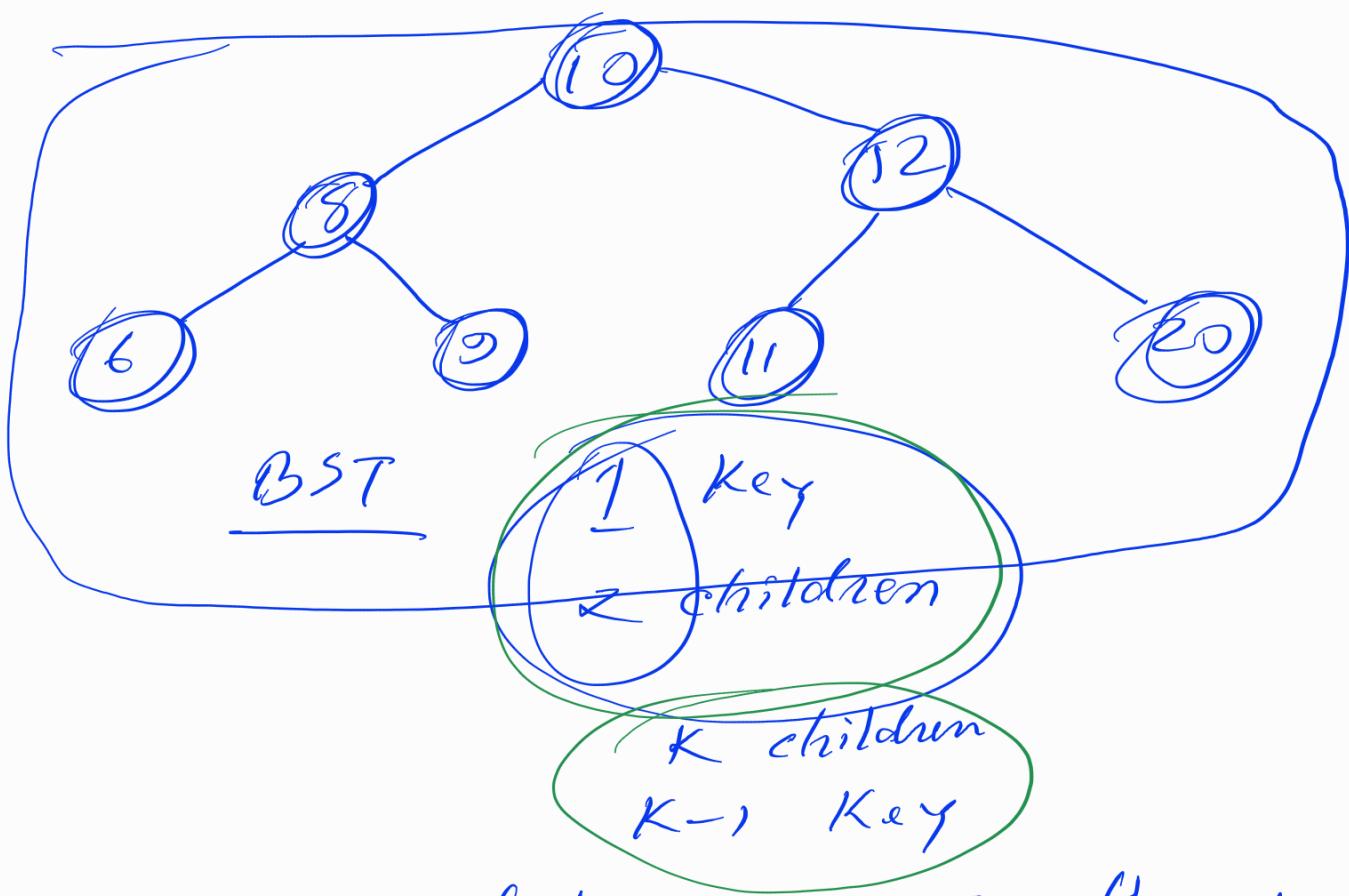
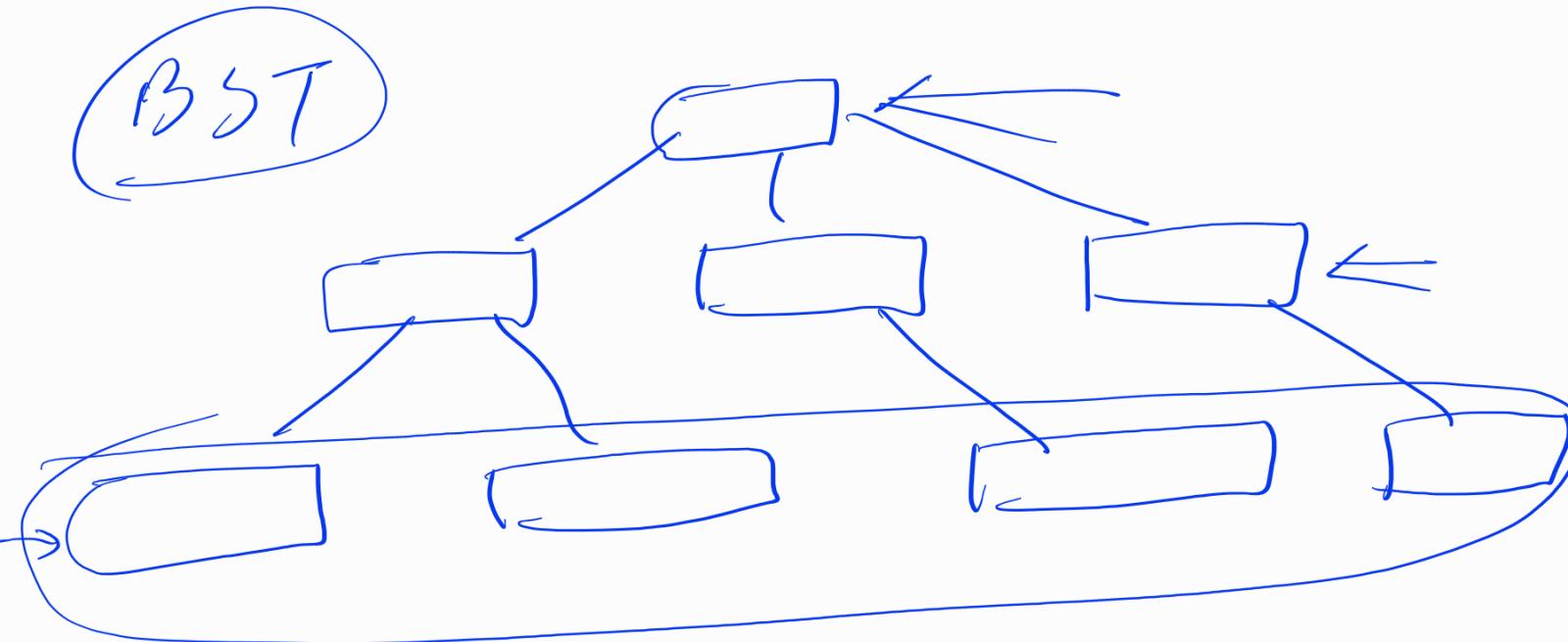
$$\text{No of entries per Block} = \frac{512}{16} \\ = 32$$

$$\text{For 100 records No of Blocks} = \frac{100}{32} \\ = 3.2 \approx 4 \text{ blocks}$$

Without indexing = 25 blocks
 With indexing = $(4+1)$ blocks
 \downarrow
 5 blocks

Multi-level indexing

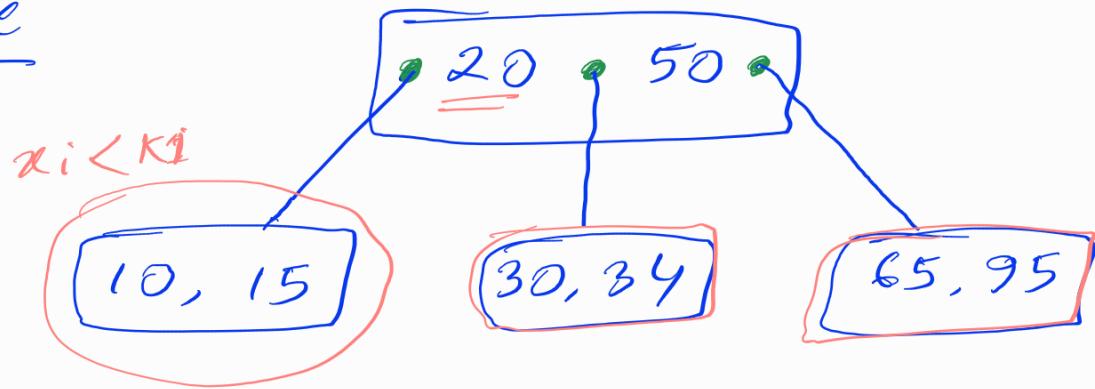




□ m-way search tree → more than 1 key, more than 2 children

$$(k_1 < k_2 < k_3 < \dots < k_n)$$

Example

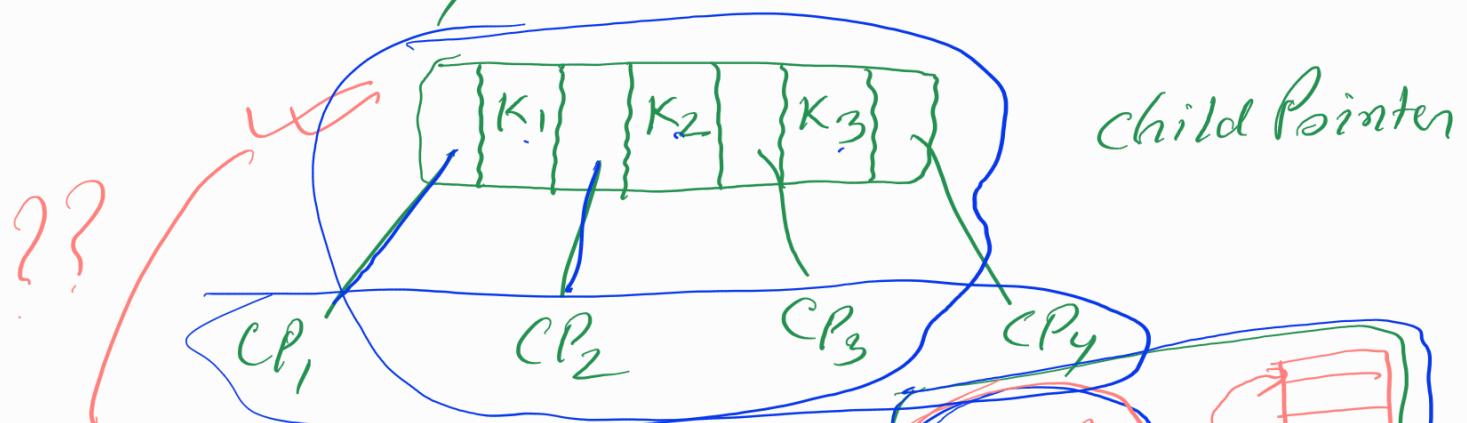


2 Keys
3 children

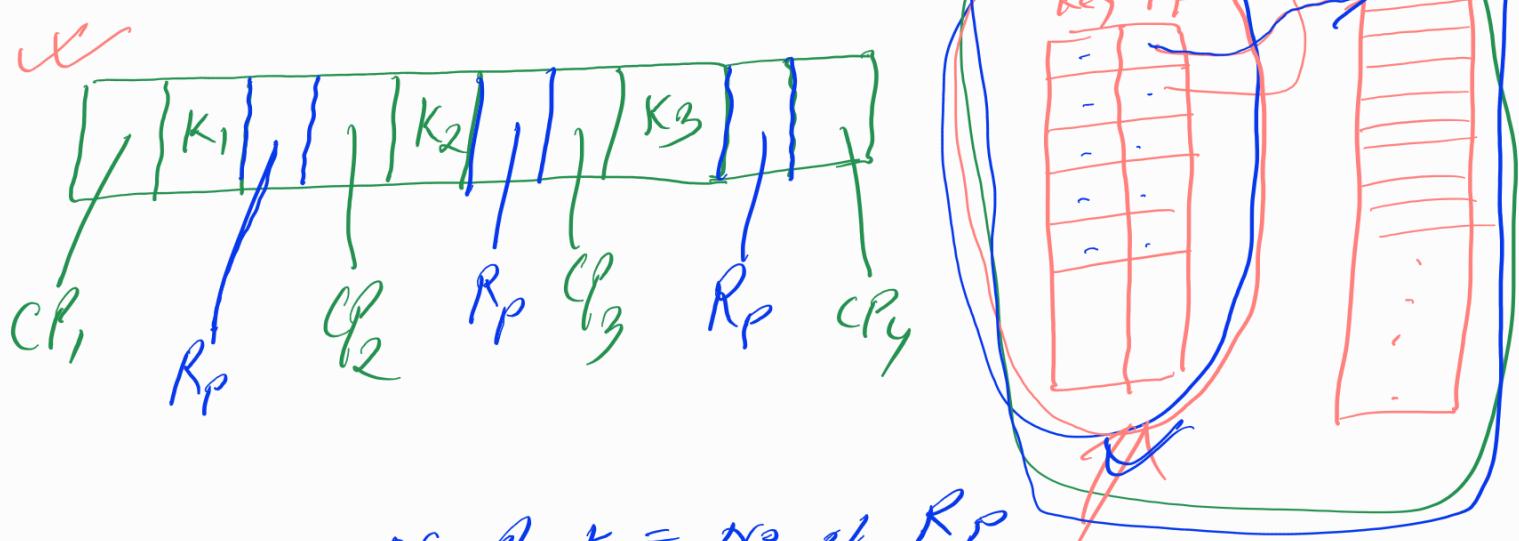
In general, m -way search tree
 $(m-1)$ Keys

this is a 3-way search tree.

4-way S.T.

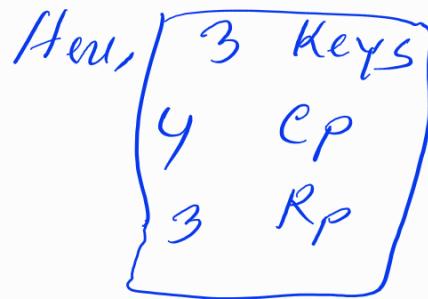


child Pointer



No. of K = No. of RP

m-way S.T.



Problem

It grows in one direction.

It is NOT a balanced tree

Solⁿ: B Tree = m-way ST + Rules

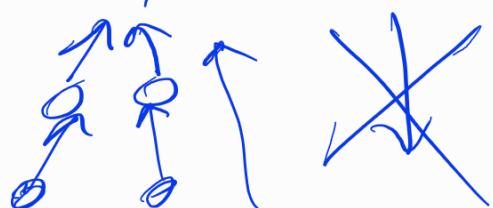
Rules

1. Every node $\lceil \frac{m}{2} \rceil$ children

2. Root can have 2 children
(min)

3. All leaves are at the same level
(i.e. Balanced)

4. Construction is done by Bottom-up



Example

m-way ST
 $m=4$

$$K = \frac{m-1}{m} = \frac{4-1}{4} = \underline{\underline{3}}$$

max size of each node

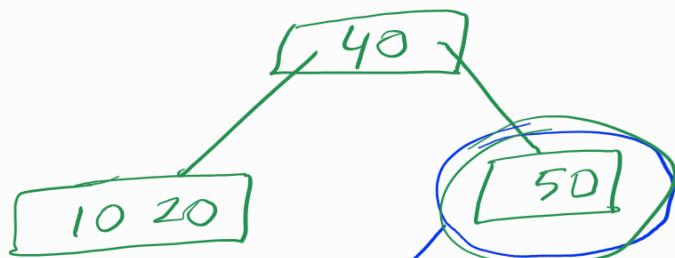
$t_1 = 5$: 10 20 40 50



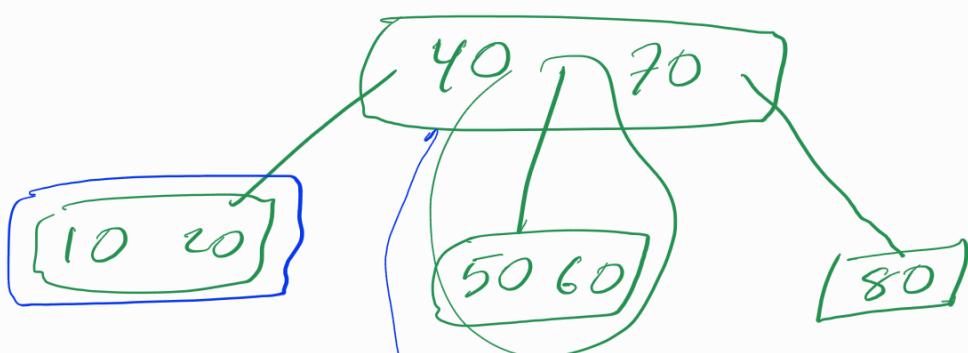
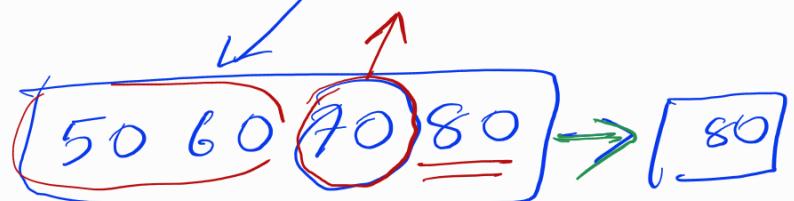
No space



50, go for splitting



$t_2 = 60, 70, 80$



Now, 30, 35 ↗

