D.D.Londhe

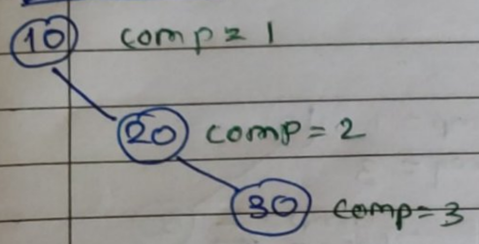
Optimal Binary Search Tree – I

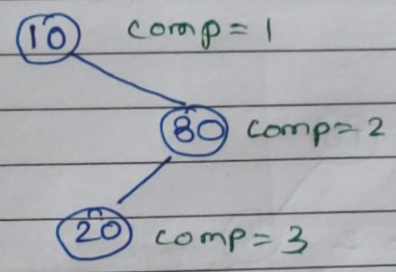
If the keys are given, we want to know which Binary search tree preferred. i.e. The tree with minimum no. of comparisons.

e.g. keys [ ] = {10, 20, 30}

Possible BST's are :- 5.

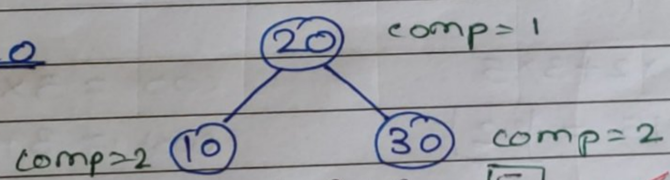
Root = 10.





cost = $\frac{1+2+3}{3} = \frac{6}{3} = \boxed{2}$ cost = $\frac{1+2+3}{3} = \boxed{2}$

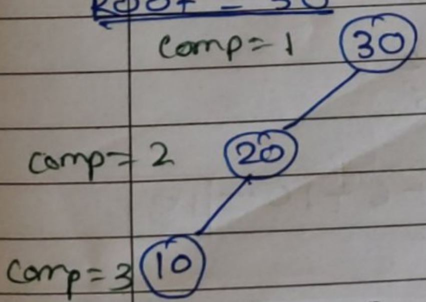
Root = 20

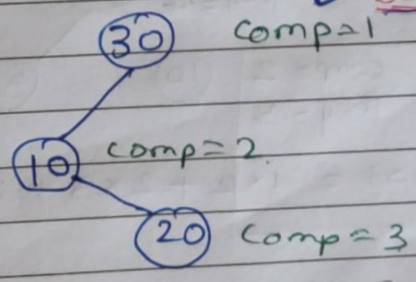


cost = $\frac{1+2+2}{3} = \boxed{\frac{5}{2}}$

Binary Tree with minimum cost.
Reasons = height is less than others
Balanced BST

Root = 30





cost = $\frac{1+2+3}{3} = \frac{6}{3} = \boxed{2}$ cost = $\frac{1+2+3}{3} = \frac{6}{3} = \boxed{2}$

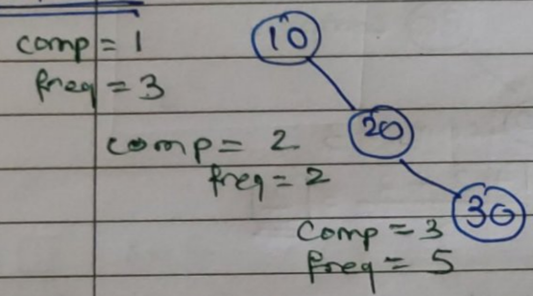
cost of searching the key in BST
= Average No. of comparisons

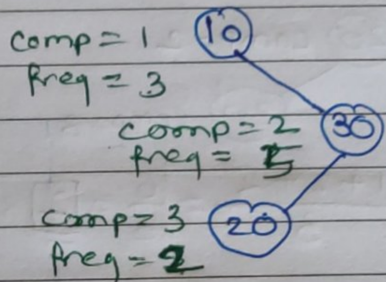
Optimal Binary Search Tree-II

If the frequency for searching key are given. (Some keys are frequently searched)
If frequency are defined for each keys, we should know which tree organization is "optimal."

keys[] = {10, 20, 30}   freq = {3, 2, 5}

root = 10

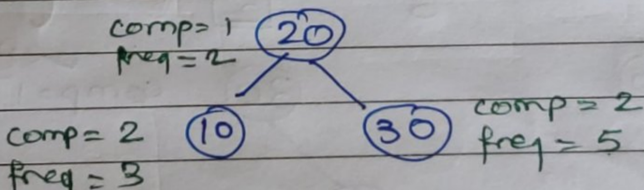




cost = 3×1 + 2×2 + 3×5
cost = 3 + 4 + 15 = $\boxed{22}$

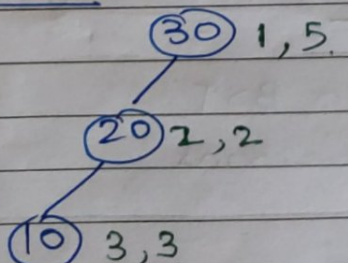
cost = 3×1 + 5×2 + 2×3
cost = 3 + 10 + 6 = $\boxed{19}$

root = 20

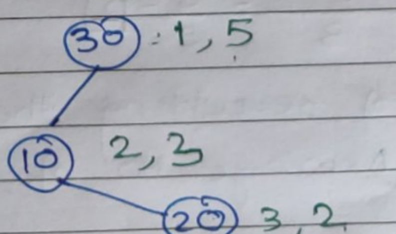


cost = 1×2 + 2×3 + 2×5 = 2 + 6 + 10 = $\boxed{18}$

root = 30



cost = 5 + 4 + 9
cost = $\boxed{18}$



cost = 5 + 6 + 6
cost = 15 + 2 = $\boxed{17}$

Optimal Binary Search Tree

Based on the frequencies given, the tree with cost = 17 is optimal, even if it is not height balanced.

When more than 3 nodes with their frequencies are given then how you will find out OBST. follow ~~to~~ the steps:-

Take the keys & sort it. assign node nos.

Node Nos. →	0	1	2	3
Nodes →	10	20	30	40
frequency →	4	2	6	3

Make Matrix of size n×n.

	0	1	2	3
0	4	8 [0]	20 [2]	26 [2]
1		2	10 [2]	16 [2]
2			6	12 [2]
3				3

Follow the cases to fill the matrix.

Case I When $l=1$ i.e. one node is taken to make BST.

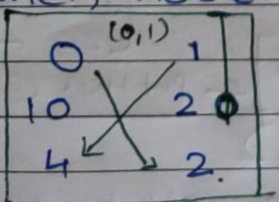
Node	0	1	2	3
	10	20	30	40
	(0,0)	(1,1)	(2,2)	(3,3)
	freq = 4	freq = 2	freq = 6	freq = 3

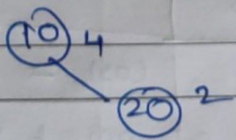
Put this frequency at (0,0), (1,1), (2,2), (3,3) in the matrix above.

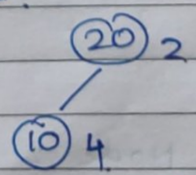
Case II $l=2$ i.e. when 2 nodes are taken to make BST.

1) when node 0 & 1 considered.

(0,1)







$c = 4\times1 + 2\times2$ $\boxed{c = 8}$ ✓

$c = 2\times1 + 4\times2$ $\boxed{c = 10}$

with formula -

$$\text{cost} = 4 + 2 + \min \begin{cases} 2 \leftarrow 0 \text{ is root} \\ 4 \leftarrow 1 \text{ is root} \end{cases}$$

$$= 6 + \min \begin{cases} 2 \leftarrow 0 \\ 4 \leftarrow 1 \end{cases}$$

$$\boxed{\text{cost} = 8^{[0]}} \xleftarrow{\text{root}} \text{put this cost at } (0,1)$$

2) when node no. 1, 2 is considered for BST (node 20 & 30)

	(1,2)
1	2
20	30
2	6

cost = 2 + 6 + min { 6 ← 1 is root ; 2 ← 2 is root ✓ }

cost = 8 + 2

$\boxed{\text{cost} = 10^{[2]}}$ ← put this cost at (1,2)

3) when node nos 2,3 are considered for BST i.e. 30, 40

	(2,3)
2	3
30	40
6	3

cost = 6 + 3 + min { ✓3 ← 2 is root ; 6 ← 3 is root }

= 9 + 3

$\boxed{\text{cost} = 12^{[2]}}$ ← put this cost at (2,3)

Case III l = 3 when 3 nodes are considered for BST

1) 0, 1, 2.

0	1	2
10	20	30
4	2	6

cost = 4 + 2 + 6 + min { 10 ← 0 — (1,2) ; 4 + 6 ← 1 · (0,0) + (2,2) ; ✓ 8 ← 2 (0,1) }

= 12 + 8 = 20²

$\boxed{\text{cost} = 20^{[2]}}$ ← put this cost at (0,2)

2) 1, 2, 3

1	2	3
20	30	40
2	6	3

cost = 2 + 6 + 3 + min { 12 ← 1 (2,3) ; 2+3 ← 2 (1,1) + (3,3) ; 10 ← 3 (1,2) }

= 11 + 5

$\boxed{\text{cost} = 16^{[2]}}$ ← put this cost at (1,3)

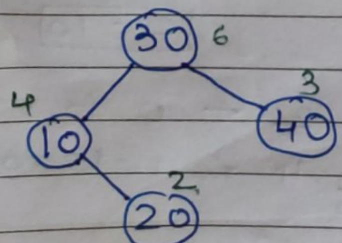
Case IV l = 4 when 4 nodes considered for BST.

0	1	2	3
10	20	30	40
4	2	6	3

cost = 4 + 2 + 6 + 3 + min { 16 ← 0 (1,3) ; 4 + 12 ← 1 (0,0) + (2,3) ; ✓ 8 + 3 ← 2 (0,1) + (3,3) ; 20 ← 3 (0,2) }

$\boxed{\text{cost} = 26^{[2]}}$ ← put this cost at (0,3)

Make BST Node 2 is at root i.e. 30



- Right of 30 will be 40.
- Left of 30 will be either 0 or 1. → check (0,1), it is 8[0]
  This says node 0 is at root.
- So 10 will be left of 30.
- Right of 10 will be remaining node 20.

cost = 6×1 + 4×2 + 3×2 + 2×3

$\boxed{\text{cost} = 26}$ ✓ OBST