CT-2 (2nd internal test)

Date - 14 march 2021

Name - Aman Yadav
Subject - Data Structure
Department - CSIT

Ans No - 05

Binary search: It search a sorted array repeatedly by dividing the search intervals in half. Begin with an interval covering the whole array. if the value of search key is less than the item in the middle of interval, narrow the interval to the other half otherwise narrow it to upper half. Repeatedly check lower until the value is found or the interval is empty.

example: let array be

{8, 10, 12, 21, 27, 34, 42}

if we have to search 27 than we will find mid of the array

mid

{8, 10, 12, (21), 27, 34, 42}

now compare mid ~~width~~ which is 21

here 21 < 27, therefore 27 will be after 21

now we will again find mid in

{27, (34), 42}

mid

now we compare mid with 27 i.e 34 > 27

therefore it will be behind 34

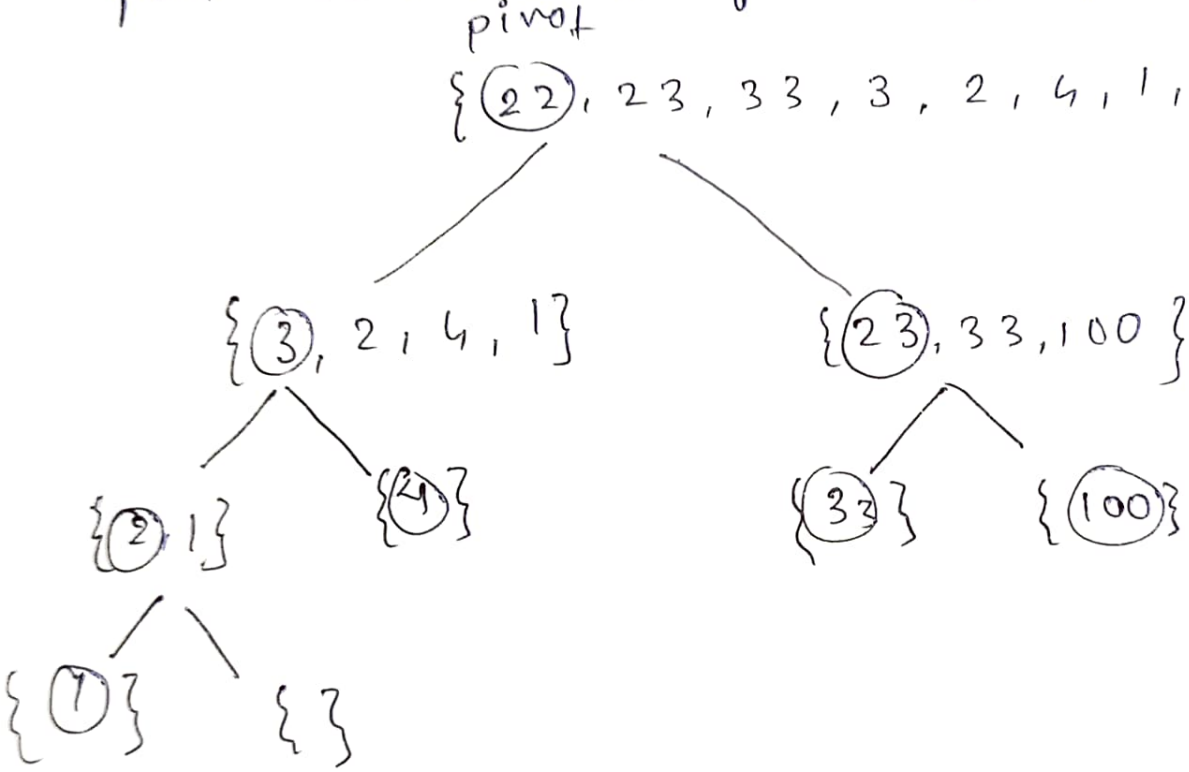
now 27 = 27 : we found the key and we can print the index value of 27.

time complexity is $O(\log_2 n)$

Answer No. 2

Quick Sort: {22, 23, 33, 3, 2, 4, 1, 100}

we have to take 1st element as pivot and the low values from 22 will be on left and high from 22 is on right



sorted elements by quicksort

sorted → {1, 2, 3, 4, 22, 23, 33, 100}

Answer No-3

1st pass → 42, 29, 75, 11, 65, 58, 60, 18
→ 29, 42, 75, 11, 65, 58, 60, 18
→ 29, 42, 75, 11, 65, 58, 60, 18
→ 29, 42, 11, 75, 65, 58, 60, 18
→ 29, 42, 11, 65, 75, 58, 60, 18
→ 29, 42, 11, 65, 58, 75, 60, 18
→ 29, 42, 11, 65, 58, 60, 75, 18
→ 29, 42, 11, 65, 58, 60, 18, 75

2nd pass → 29, 42, 11, 65, 58, 60, 18, 75
→ 29, 11, 42, 65, 58, 60, 18, 75
→ 29, 11, 42, 65, 58, 60, 18, 75
→ 29, 11, 42, 58, 65, 60, 18, 75
→ 29, 11, 42, 58, 60, 65, 18, 75
→ 29, 11, 42, 58, 60, 18, 65, 75

3rd pass → 11, 29, 42, 58, 60, 18, 65, 75
→ 11, 29, 42, 58, 60, 18, 65, 75
→ 11, 29, 42, 58, 60, 18, 65, 75
→ 11, 29, 42, 58, 18, 60, 65, 75

now last three terms are sorted till 3rd pass.

Answer NO-1

given table elements { 43, 22, 1334, 1471, 9679, 1989, 6171, 6173, 4199 }

hash function x mod 9

(1) 4322 % 9 = 2
(2) 1334 % 9 = 2
(3) 1471 % 9 = 4
(4) 9679 % 9 = 4
(5) 1989 % 9 = 0
(6) 6171 mod 9 = 6
(7) 6173 mod 9 = 8
(8) 4199 mod 9 = 5

Aman Yadav (4)

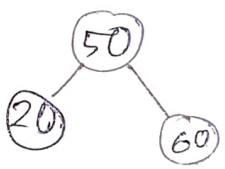
hash table →

index	
0	1989
1	
2	4322
3	1334
4	1471
5	9679
6	6171
7	4199
8	6173

after collision add 1 if empty insert if filled +2 then insert and so on.

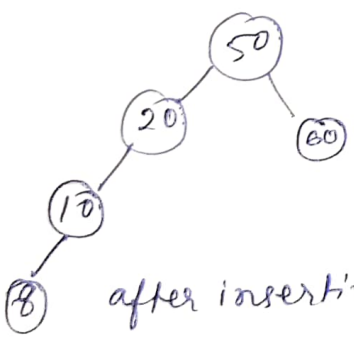
Answer NO – 4

(1) numbers are 50, 20, 60, 10, 8, 15, 32



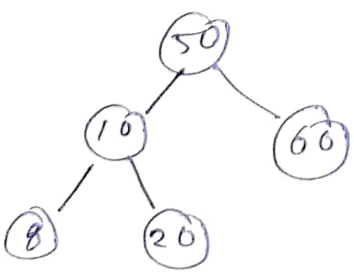
here it is still balanced

(2.) now insert 10

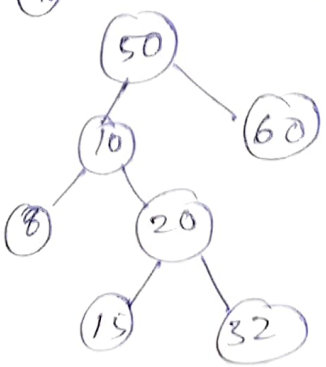


after inserting 8 it is not balanced

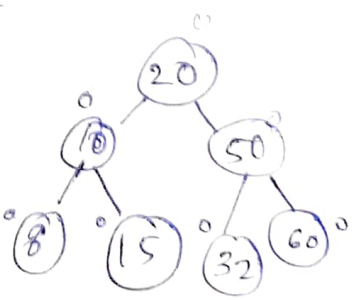
(3.)



(4.) now inset 15



→



this is AVL