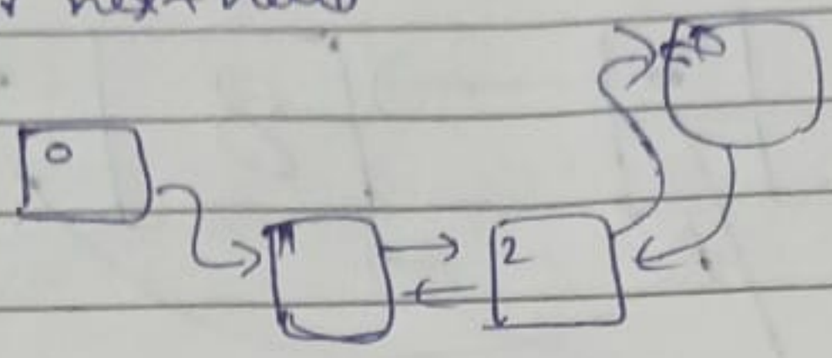
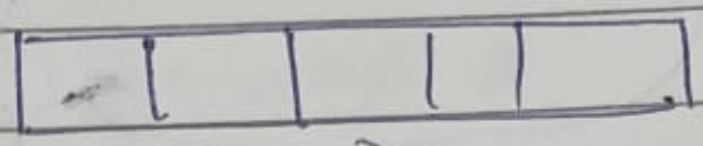
for next Cycle



Cycle Detection & it's proof. [slow pointer (moves one step at a time)
fast pointer (moves two step at a time)]

(146.) LRU Cache :-



(key, value)

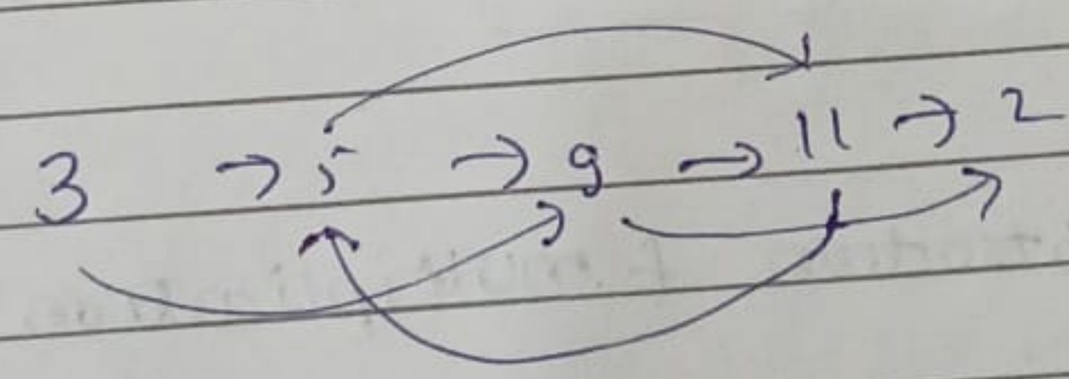
Add most recent used in front.

460 LFU Cache

Cnt (x) =

All One + LRU Cache.

(138)



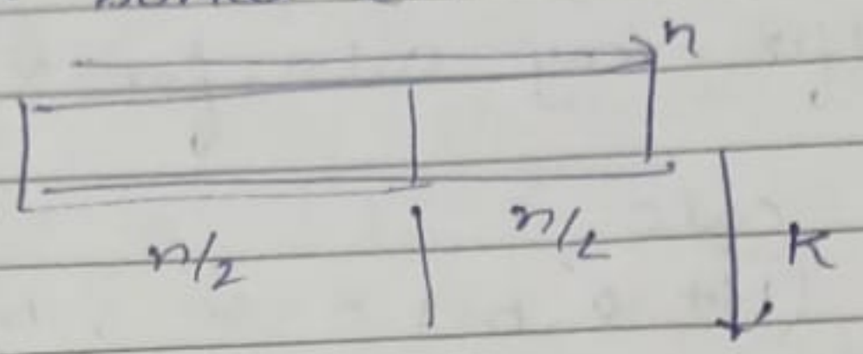
3 → 5 → 9 → 11 → 2

Page No.:
Date: 17/Feb/2016

Linked list
for Merging O(N+M)

Lecture 23

Merge 'K' sorted LL:-



$T(n) = 2T(n/2) + O(nk)$

Time Complexity $O(nk \log n)$

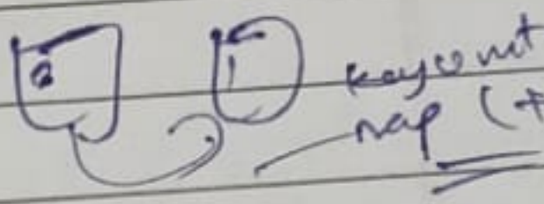
Take dummy tail & dummy head.

25. Reverse Nodes in K-Group

Time $O(N)$

Space $O(n/k)$

(432) All O' one Data Structure

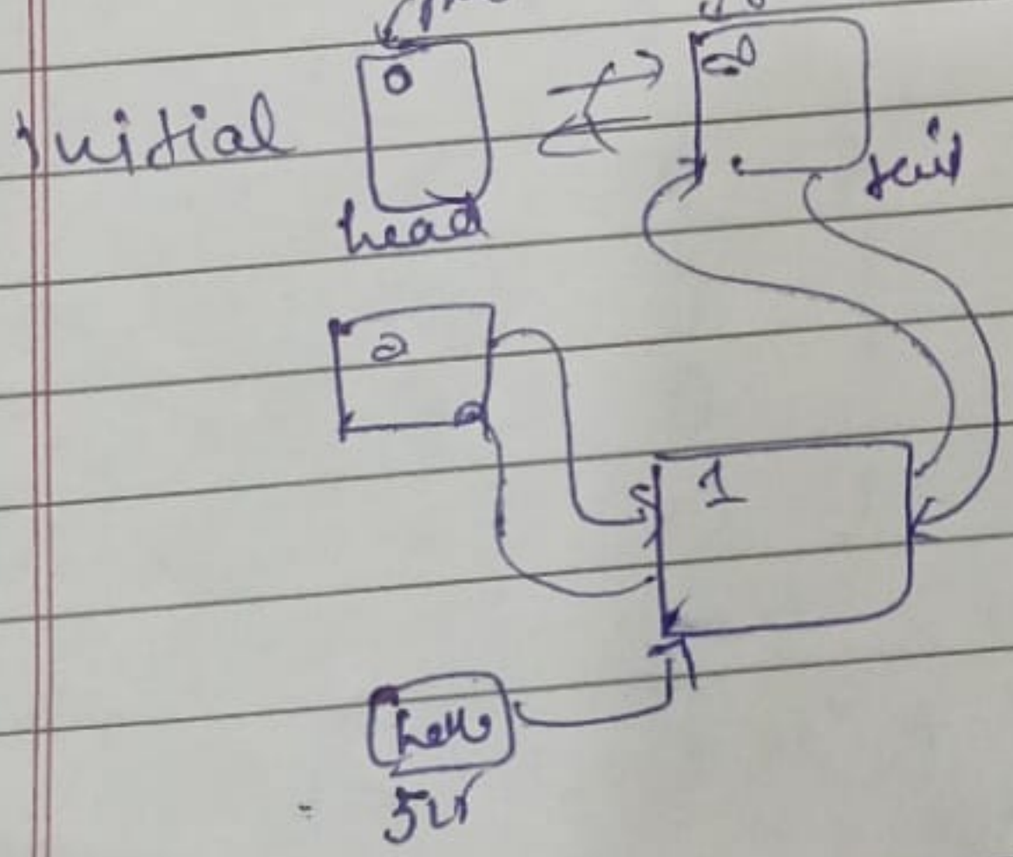


Map for addmers.

(i) freq val
(ii) next
(iii) previous
(iv) Unordered Set

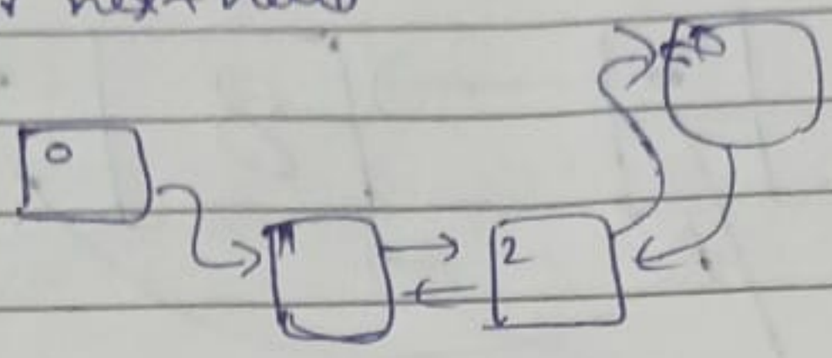
ex:-

inc("hello")
inc("hello")



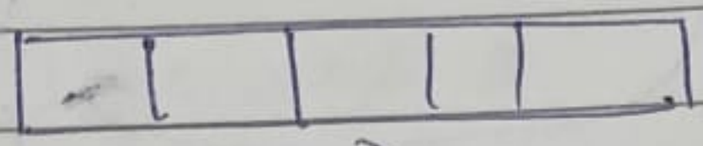
Map
hello: 5

for next Cycle



Cycle Detection & it's proof. [slow pointer (moves one step at a time)
fast pointer (moves two step at a time)]

(146.) LRU Cache :-



(key, value)

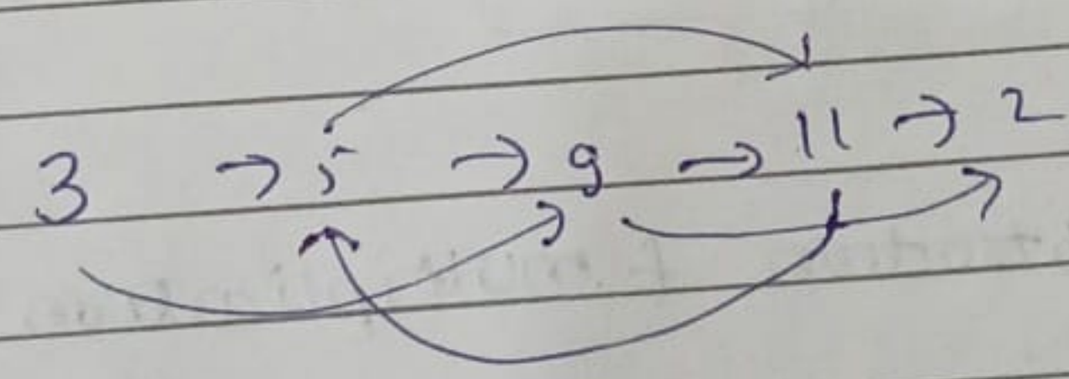
Add most recent used in front.

460 LFU Cache

Cnt (x) =

All One + LRU Cache.

(138)



3 → 5 → 9 → 11 → 2

5 is also less 6 to pop 6

5
5

ans next smaller element is 5.

Largest histogram area :-

Some for next smaller bar.

84 Leetcode.

Contribution of rectangle = (prevSmaller - nextSmaller) * (current).

1	8	7	5	6	7	1

min

← [left]

$5 \times (1 + 2 + 3 + \cdots (i-PS))$

$5 \times (2 + 3 + 4 - - - (i-PS+1))$ [right]

$5 \times (3 + 4 + \cdots (i-PS+2))$

$5 \times (rs-i) + rs-i+1 - - -$

$= \frac{n}{2}(a + a_n) = \frac{n}{2}(a \cdot$

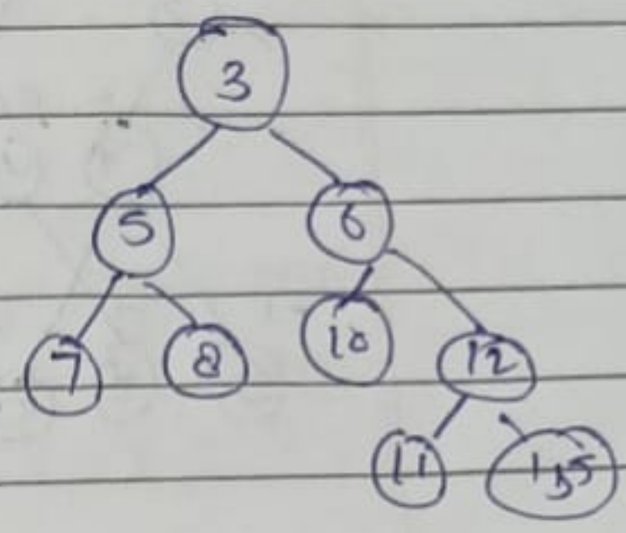
$5 \times \left(\frac{i-PS}{2}\right)$

$\text{ans}(i) \times \frac{ls}{2} \times \frac{rs}{2}\,(ls+1 + ls+1 + (n-1)\,n)$

Page No.
Date 18/02/25

Trees

BFS :- Levelorder Traversal
without Queue
with Arrays
→ curlevel [root]
nextlevel [ ]



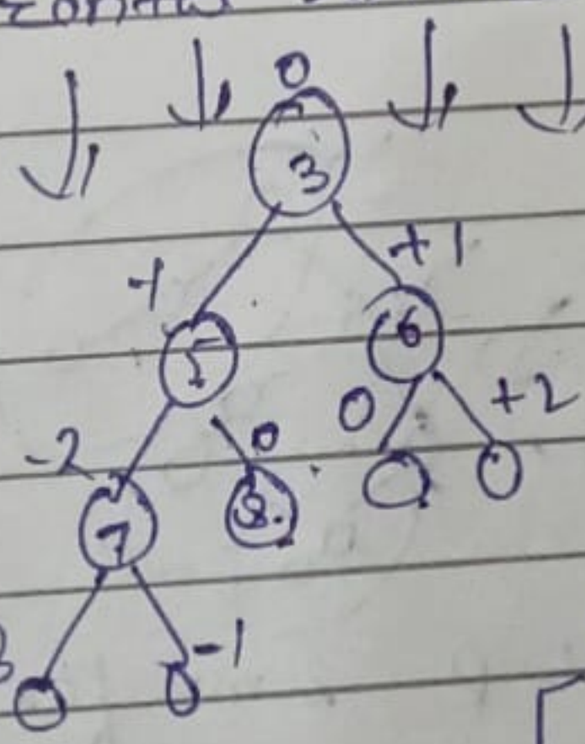
ex :- curlevel [3]
(i) nextlevel [5,6]
(ii) curlevel [5,6]
nextlevel [7,8,10,12]

Given a tree find no. of valid BFS / LOT.

Each level pe elements ka factorial then usko multiplication karo answer.

Left view :-
for ex :- 5, 7 in above example i.e. each level ki first node.

Top view & Bottom view
↓ Horizontal Distance



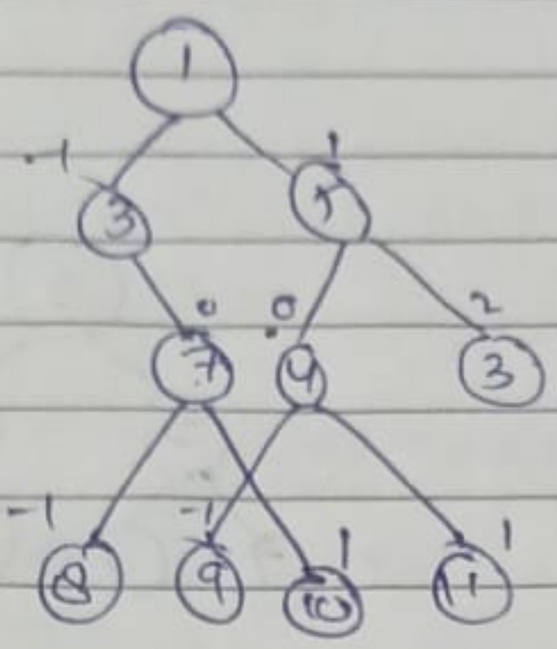
horiz. dist.
↓
curlevel = [0, 3]
nextlevel = [(-1,5), (+1,6)]
curlevel = [-1, 1]
nextlevel = [(-2,7) (0,0)]

Map :-
0 : 3
-1 : 5
1 : 6

if not map
because of collision
then we can use
DU.

Bottom view:

horizontal dist, level : node.



hd level node

0 : [0,1]

-1 : [1,3]

1 [1,5]

for 7 remove 0,

0 : [(2,7)]

-1 : (1,3]
(1,5)

1 :

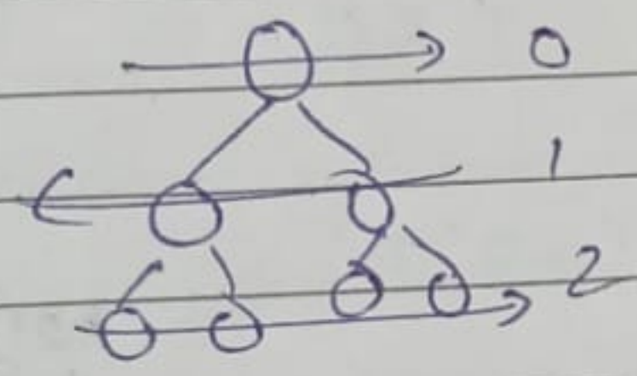
for 4 append bcs hd' is 0.

0 : [(2,7), (2,4)]

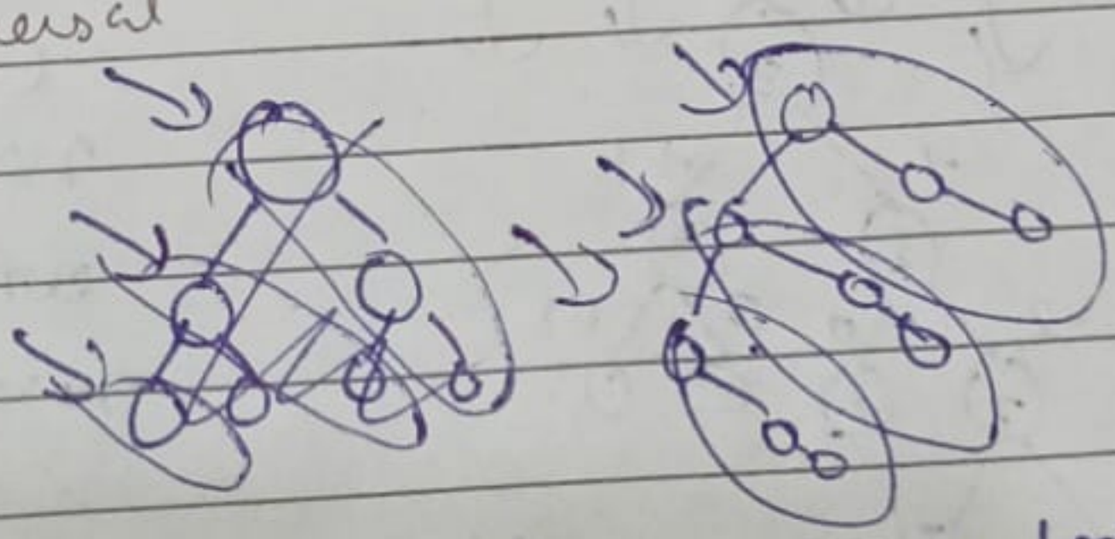
-1 : [(1,3)]

1 : [1,5]

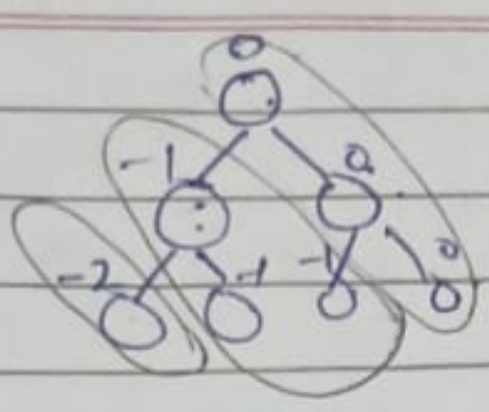
Zig-Zag Traversal:-



Rainy Traversal



Horizdist:- -1 for left but keep same for right

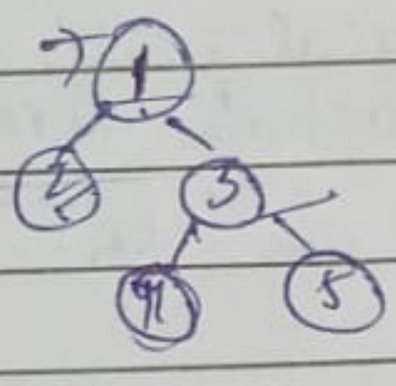


opposite for opp side rain.

(297) Serialize and Deserialize Binary tree

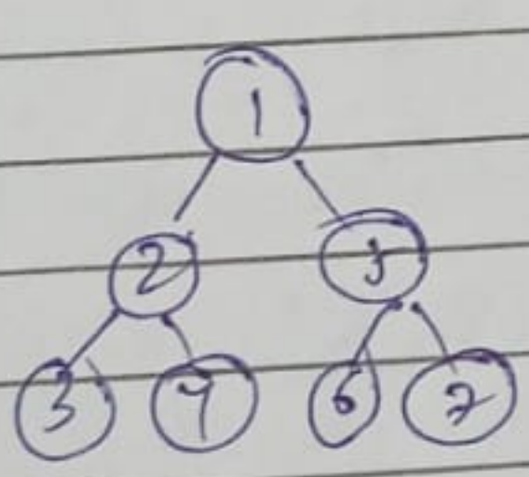
(1028)

We can retrieve a tree from only preorder traversal but they should be store in a diff. way. (like dash (-)).



[1, 2, 3, null, null, 4, 5]

1 - 2 - 3 - - 4 - 5 - - 6 - - 7



Rec(Traversal, s, e)

Rec(Traversal, s+1, idx-1)

Rec (Traversal, idx, e)

$T(n) = T(n-1) + O(n)$

$O(n^2)$

Optimized by this :-

OCC

Page No.:
Date: / /

Occurances store in array
then do bin search.

$$T(n) = T(N-1) + O(\log n)$$
$$= O(N \log n)$$

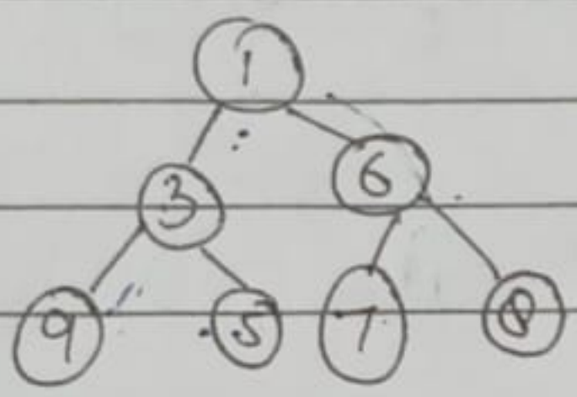
for inorder ⇒ $O(N)$ 3rd Approach.

Ques) ON Reight & diameter :-

Pre
O

ex- 24, 1443.

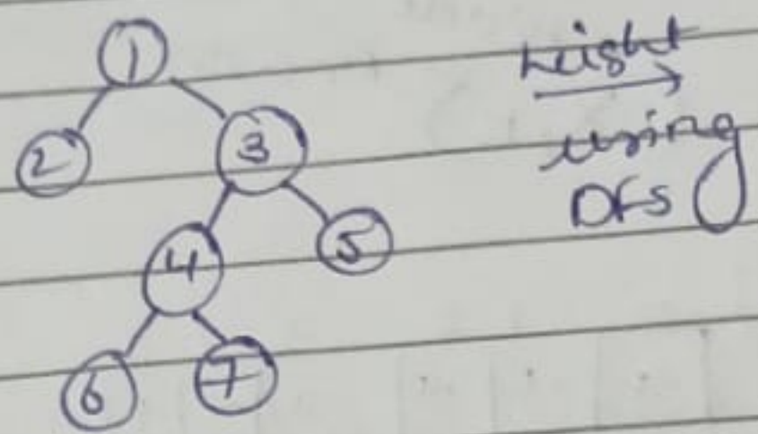
path score = nodes ki value count no. of vertical path reaching directly to root (jiska sum ≤ k).



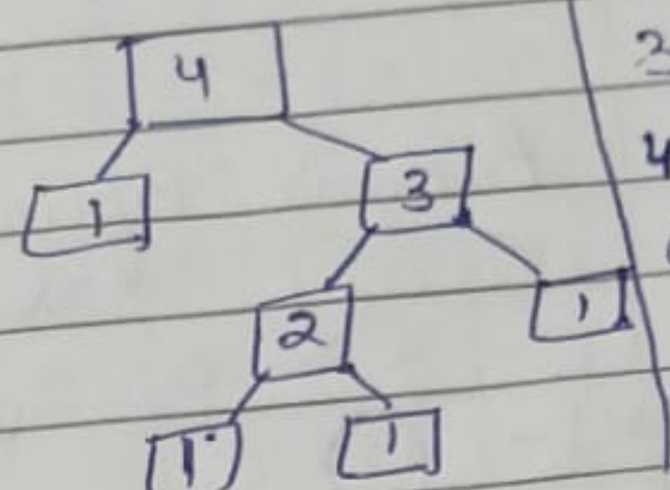
Page No.
Date 19, Feb, 2023

Sum of height of tree when rooted at each index.

Rerooting DP



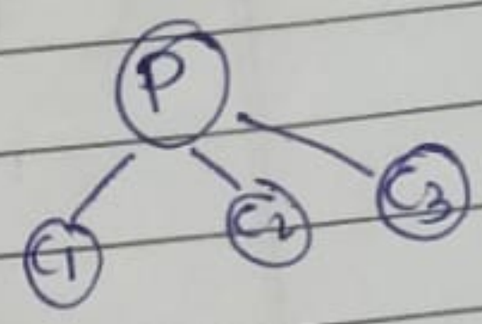
O(N) + O(N)
= O(2N)



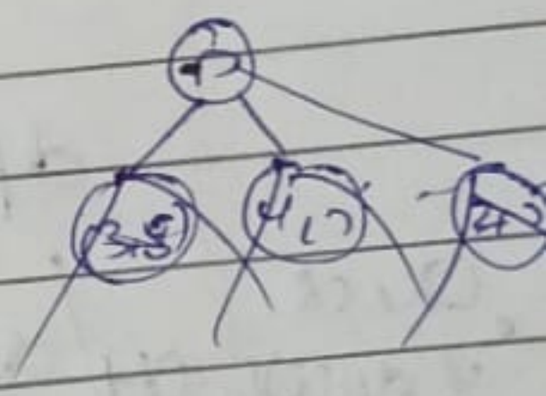
Root : height
1 : 4, 2
3 : 3, 3
4 : 4, 2
6 : 5, 1

Height (Parent) = Max (ht of children + 1);

Height (C_3) = Max (children of C_3,



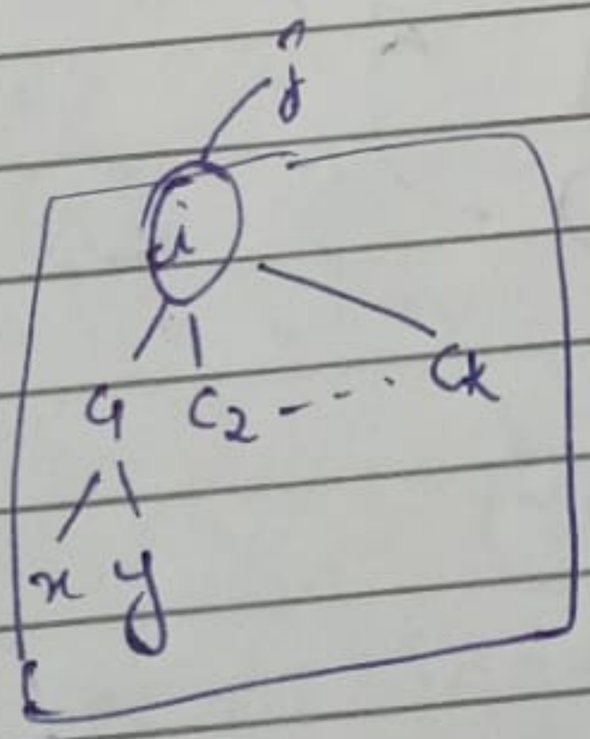
	1	2	3	4	5	6
ap	(3,2)	(1,1)	(3,2)	(2,2)	(1,1)	(1,1)
pardp	(4,2)		(3,3)	(4,2)		



(1) adjacency list

(2) addedge in it.

(3) dfs1 (finding height of subtrees).



Subordinates

$$subs(i) = \sum_{i=1}^{k} (1 + subs(C_i))$$

2458

* Centroid Decomposition] *interview*
HLD] CP/ACM ICPC

Page No.
Date: / /

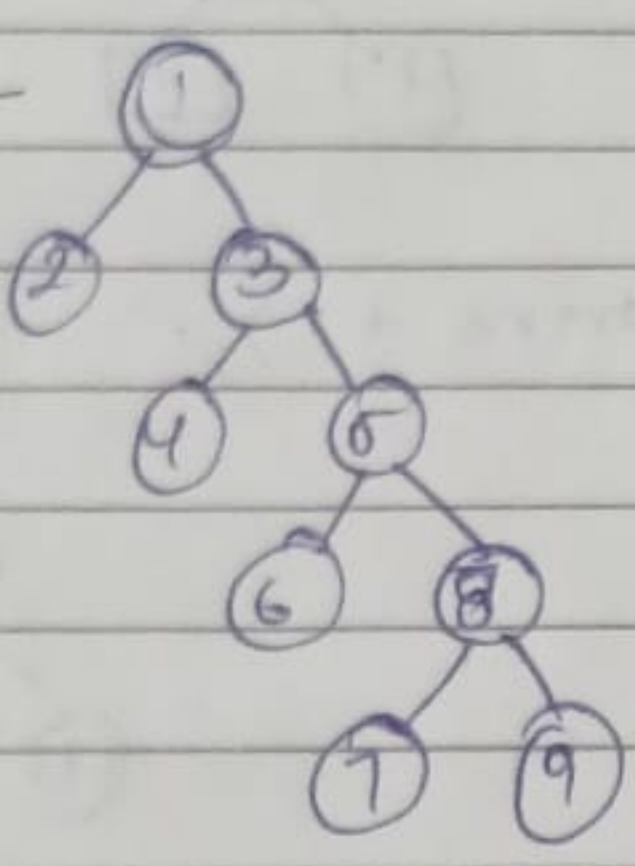
834 Sum of Distances in Tree

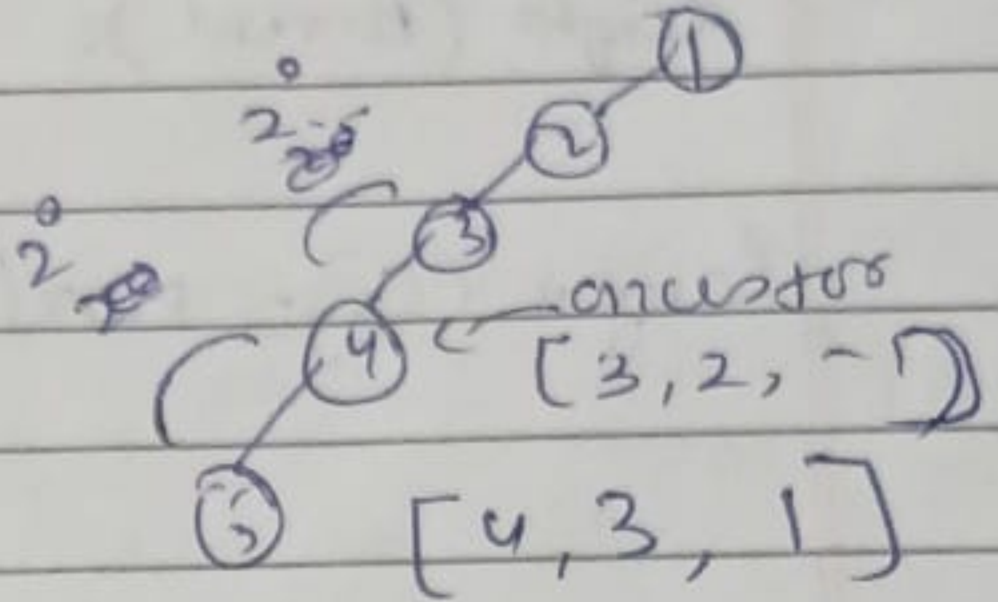
Binary lifting

(X, K)
node, Ancestor

0	1	2	3	4	5
-1	-1	-1	-1	-1	-1

ex:-

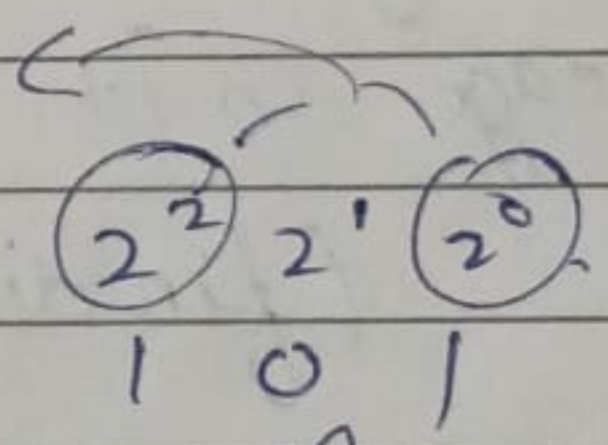




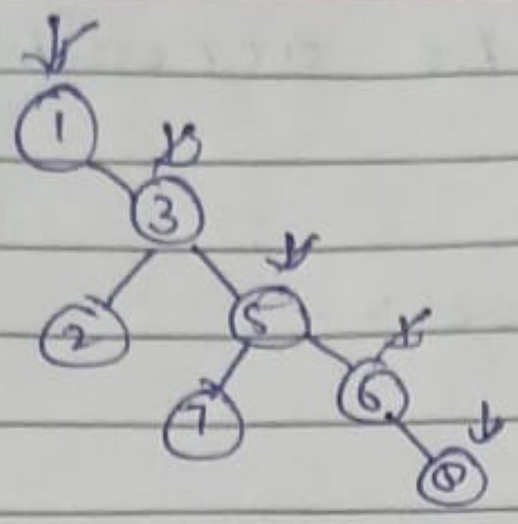
dp[i][j] = $2^{j\text{th}}$ ancestor of node i

Check if dp[i][j-1] != -1 / dp[i][j] = dp(dp(i)(j-1))(j-1)
else dp[i][j] = -1

k^{th} ancestor = 5



for sparse table O(N log N)
for binary rep. O(log N)



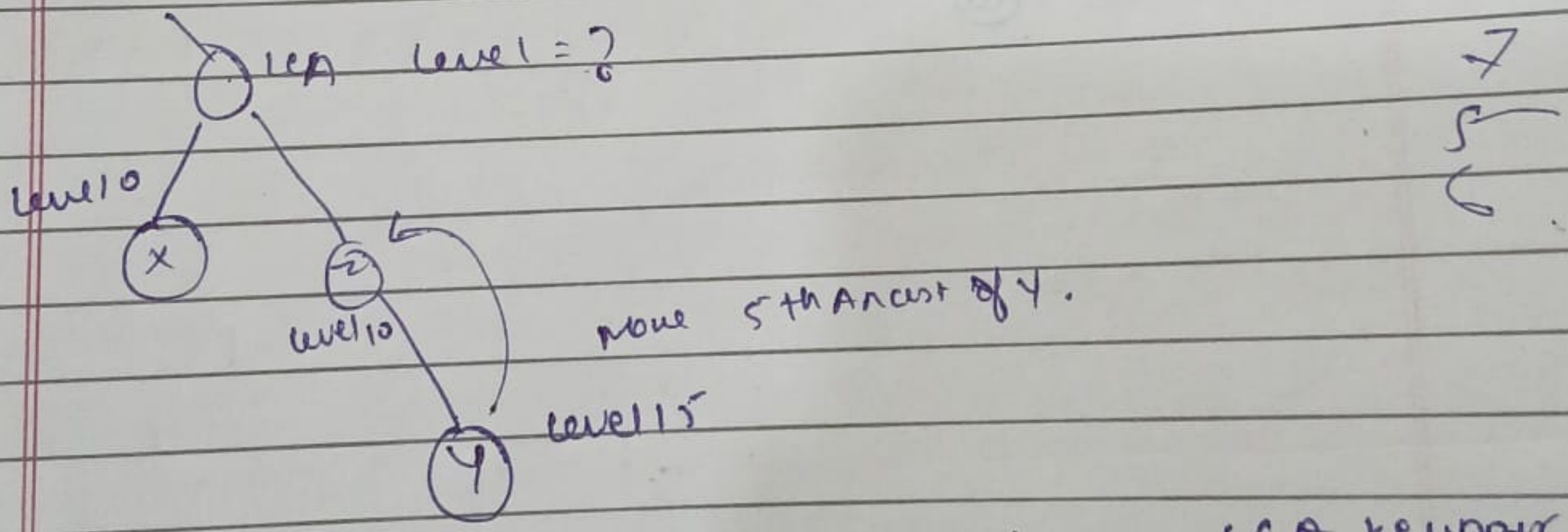
	0	1	2	3	4
1	-1	-1	-1	-1	-1
2					
3	1	-1	-1	-1	-1
4					
5	3	1	-1	-1	-1
6	5	3	-1	-1	-1
7					
8	6	5	1	-1	-1

ex for 8 ka 4th parent
i.e. 8 1 0 0
2^2 (in table)
2^2 is 1.
ans = 1.

1483 Kth ancestor of a tree node :-

Lowest Common Ancestor (LCA).

Normally O(NQ) Q → Queries.
With binary lifting O(logN)
Query. (x, y)



Now binary search use bcs. LCA ke upper level be x & y will have same ancestor.
that is monotonous search space.

* level of each node should be accessible in O(1) ~~(by spa.~~

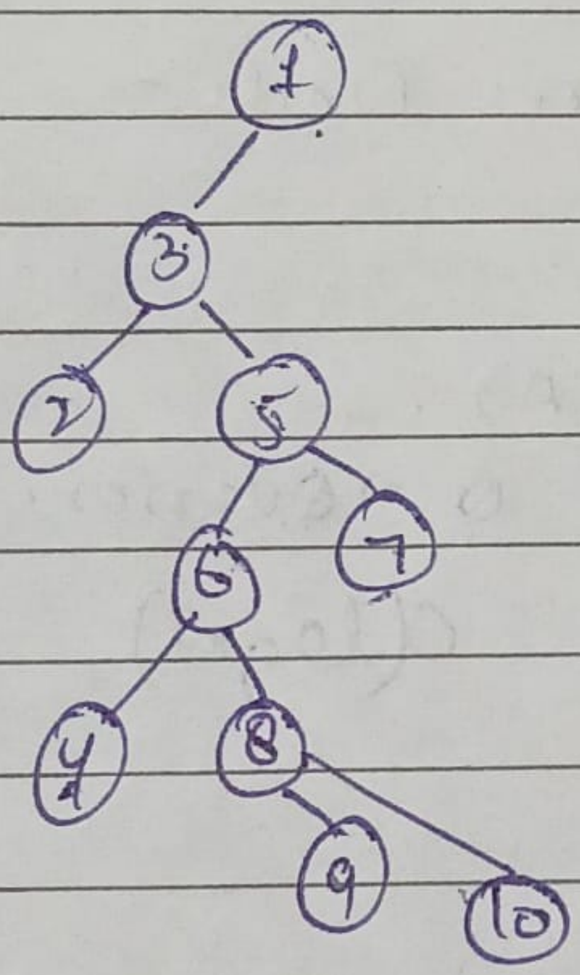
* Sparse table should be made.

* x y

* lvl[x] lvl[y]

if (lvl[x] > lvl[y]) ⇒ swap (x,y)
↓ this ensures lvl(x) will always be greater.

Z ⇒ lvl[y] - lvl[x]. Ancestor.



Query (7,4)
ans = 5

Query (3,9)
ans = 3

Query (4,10)
ans = 6

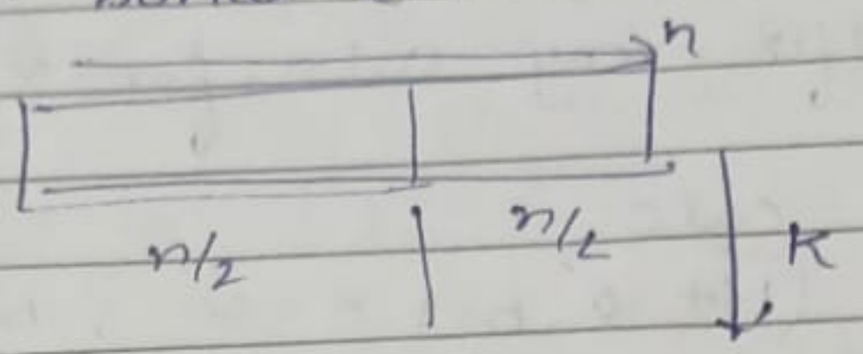
5-4
2-9

Page No.:
Date: 17/Feb/2016

Linked list
for Merging O(N+M)

Lecture 23

Merge 'K' sorted LL:-



$T(n) = 2T(n/2) + O(nk)$

Time Complexity $O(nk \log n)$

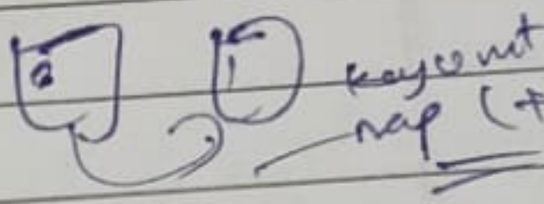
Take dummy tail & dummy head.

25. Reverse Nodes in K-Group

Time $O(N)$

Space $O(n/k)$

(432) All O' one Data Structure

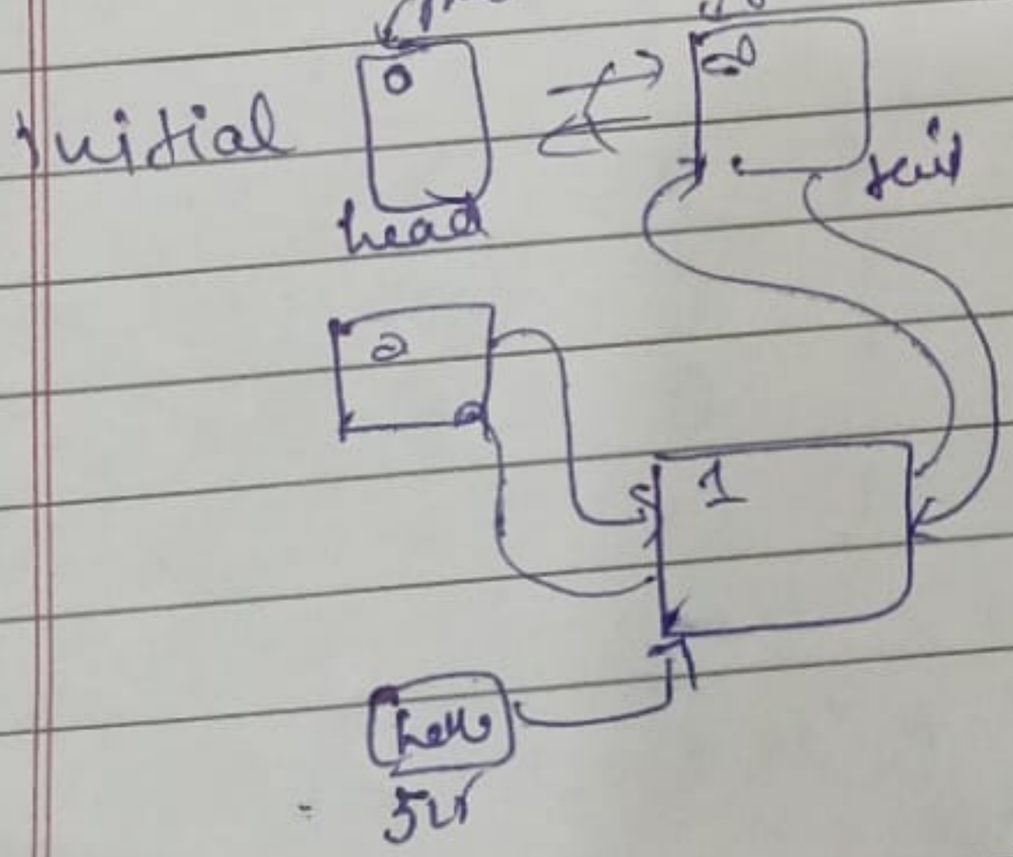


Map for addmers.

(i) freq val
(ii) next
(iii) previous
(iv) Unordered Set

ex:-

inc("hello")
inc("hello")



Map
hello: 5