

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

July 2022

Week 31

Day 209 - 156

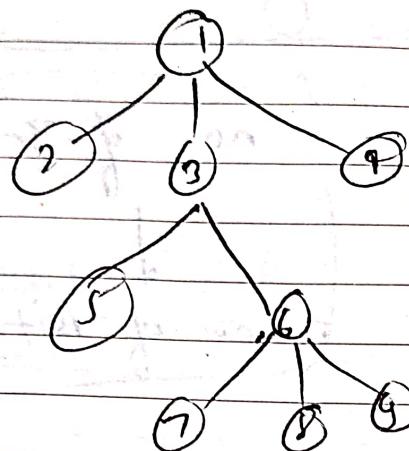
Date 28 - 07 - 2022

28

Thursday

Binary lifting & Euler tour :-

(Q) print k^{th} parent of a node



Brute f

Create a parent array

parent [0|1|1|1|1|2|3|6|2|c]

while ($k \neq -1$)

node = parent[node]

end node;

Q generic

(node, k)

Do it alone

$O(n) + O(q \times E)$

ref ref

Intuition →

K

written as sum of powers of 2.

29

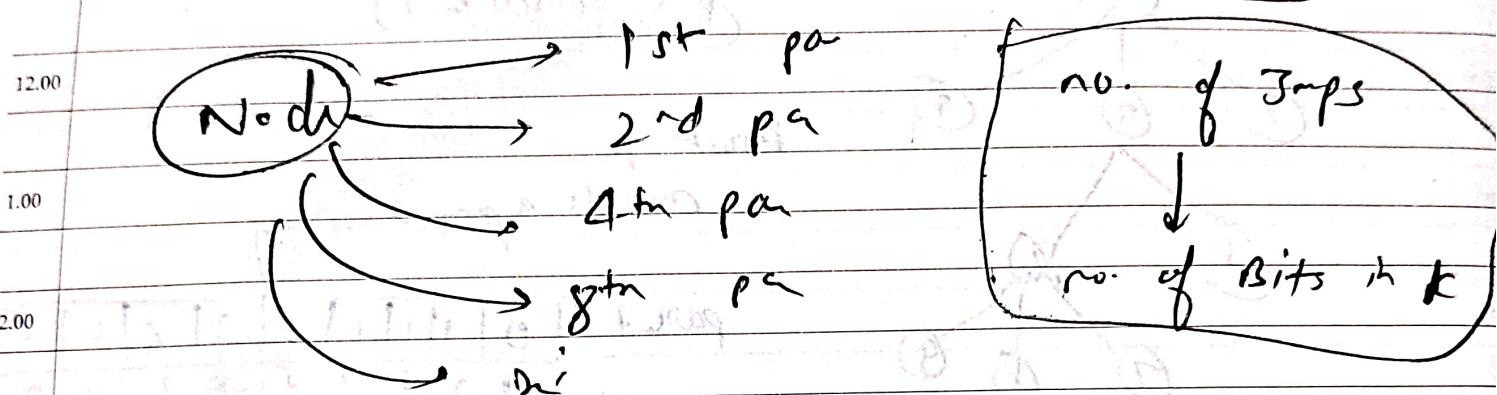
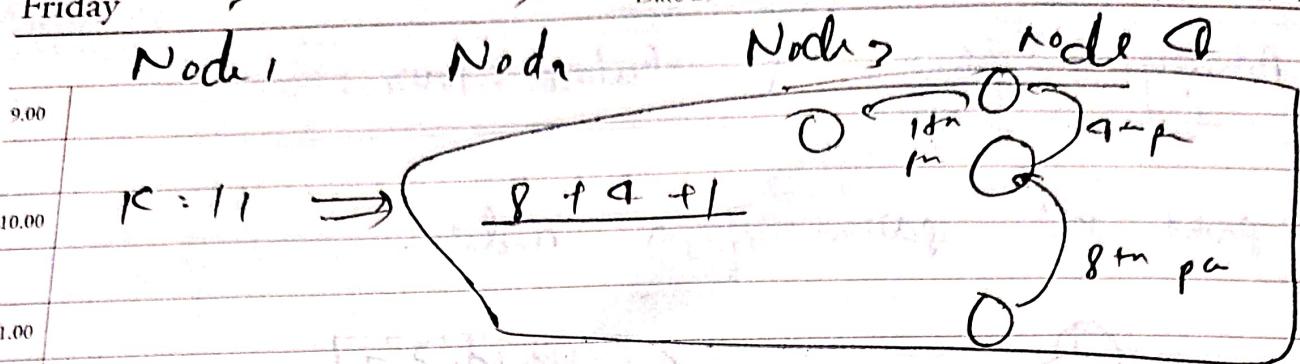
Friday

Binary Jumping

July 2022

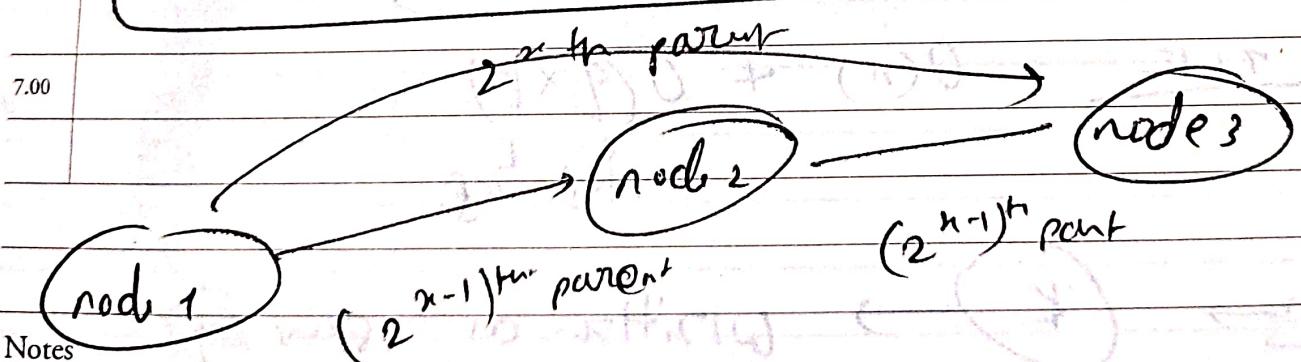
Week 31 June
Day 210 - 153
Date 29 - 07 - 2022

M	T	W	T	F	S	S
21	4	5	6	7	8	9
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31



a) How many parent per node will be stored?

Parent [node][n] $\rightarrow 2^{n-1}$ parent of Node



$$\text{parent } [\text{node } i][\text{P}_{n-1}] = \text{node } 2$$

$$\text{parent } [\text{node } 2][\text{P}_{n-1}] = \text{node } 3$$

When an individual or a company decides that success has been attained, progress stops.

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

AUG 22

July 2022

Week 31

Day 211 - 154

Date 30 - 07 - 2022

30
Saturday

$$\text{parent}[\text{node}_i][n] = \text{node}_j$$

$$\text{parent}[\text{node}_i][n] = \text{parent}[\text{node}_i][n-1]$$

$$\text{parent}[\text{node}_i][n] = \text{parent}[\text{parent}[\text{node}_i][n-1]][n-1]$$

↳ If you have $(n-1)^{\text{th}}$ level, you can know
in n^{th} level

$\text{parent}[\text{node}_i][0]$ ————— Bas case.

Q'17 ≈ 10⁻

Vector < vector < int >> dp((n+1) vector<int> (17, 0));

void dfs(nod, par, adj[], dp)

q

$$dp[\text{node}_i][0] = \text{par};$$

for (i=1 ; i<16 ; i++)

q

$$dp[\text{node}_i][i] = dp[dp[\text{node}_i][i-1]][i-1];$$

q

Anxiety should be used as a catapult to help you move forward to objectives.

31

Sunday

July 2022

Week 31

Day 212 - 153

Date 31 - 07 - 2022

JUL 22

	F	S	S
4	5	6	7
11	12	13	14
18	19	20	21
25	26	27	28
			29
			30
			31

for (auto it : adj[nod])

{ if (it == par) continue;

dfs(it, node, adj, dp);

{

{

// Now we have the DP Table

int q; ch >> q;

while (q--) {

int k; node;

ch >> k >> node;

// Kth part of node

if node == 1, int cat = 0.

while (k) {

if (k & 1)

nod = dp[node][cat];

{

cat + f;

k >> = 1;

The two hardest things to handle in life are failure & success.

} with node }

Notes

Action Plan

AUGUST

WEEK	32	33	34	35	36
MON	1	8	15	22	29
TUE	2	9	16	23	30
WED	3	10	17	24	31
THU	4	11	18	25	
FRI	5	12	19	26	
SAT	6	13	20	27	
SUN	7	14	21	28	

Diagram showing dependencies between tasks:

```

graph TD
    1((1)) --> 2((2))
    1((1)) --> 3((3))
    1((1)) --> 4((4))
    2((2)) --> 3((3))
    2((2)) --> 6((6))
    3((3)) --> 6((6))
  
```

Handwritten notes:

- $dp[1][0] = 17$
- $dp[2][0] = 7 \text{ rows} \times 17 \text{ columns}$
- $dp[1][0] = 0$
- $dp[2][0] = 0$
- $dp[3][0] = 0$
- $dp[4][0] = 0$
- $dp[5][0] = 0$
- $dp[6][0] = 0$
- $dp[7][0] = 0$
- $dp[1][1] = 3$
- $dp[2][1] = 0$
- $dp[3][1] = 0$
- $dp[4][1] = 0$
- $dp[5][1] = 0$
- $dp[6][1] = 0$
- $dp[7][1] = 0$
- $dp[1][2] = 0$
- $dp[2][2] = 0$
- $dp[3][2] = 0$
- $dp[4][2] = 0$
- $dp[5][2] = 0$
- $dp[6][2] = 0$
- $dp[7][2] = 0$
- $dp[1][3] = 0$
- $dp[2][3] = 0$
- $dp[3][3] = 0$
- $dp[4][3] = 0$
- $dp[5][3] = 0$
- $dp[6][3] = 0$
- $dp[7][3] = 0$
- $dp[1][4] = 0$
- $dp[2][4] = 0$
- $dp[3][4] = 0$
- $dp[4][4] = 0$
- $dp[5][4] = 0$
- $dp[6][4] = 0$
- $dp[7][4] = 0$
- $dp[1][5] = 0$
- $dp[2][5] = 0$
- $dp[3][5] = 0$
- $dp[4][5] = 0$
- $dp[5][5] = 0$
- $dp[6][5] = 0$
- $dp[7][5] = 0$
- $dp[1][6] = 0$
- $dp[2][6] = 0$
- $dp[3][6] = 0$
- $dp[4][6] = 0$
- $dp[5][6] = 0$
- $dp[6][6] = 0$
- $dp[7][6] = 0$
- $dp[1][7] = 0$
- $dp[2][7] = 0$
- $dp[3][7] = 0$
- $dp[4][7] = 0$
- $dp[5][7] = 0$
- $dp[6][7] = 0$
- $dp[7][7] = 0$

$$dp[2][1] = dp[dp[2][0]] [0]$$

$$dp[1][0] = 0$$

$$dp[5][1] = 3$$

$$dp[5][1] = dp[dp[5][0]] [0] \rightarrow dp[5][0] = 1$$

(Q)

Finding kth node's parent from d^e table?

int cnt = 0;
while (k > 0)

{ if (k & 1)

{ node = dt[node][cnt];

}

cnt++;

// bit no.

k = k >> 1;

}

return node;

OR

for (i = 16 ; i >= 0 ; i--)

{ if (((k >> i) & 1)

{ node = dt[node][i];

}

}

return node;

Do not take life too seriously, you will never get out of it alive.

M	T	W	T	F	S	S
1	2	3	4			
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

SEP 22

August 2022

Week 32

Day 214 - 151

Date 02 - 08 - 2022

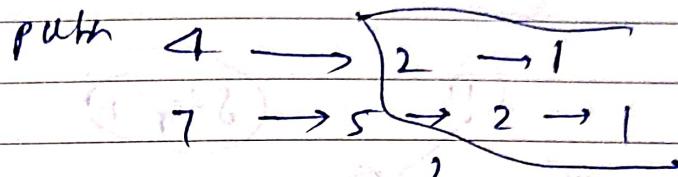
02

Tuesday

LCA

Using binary lifting

$$\text{lca}(4, 7) \Rightarrow 2$$



Bonita forced \rightarrow program parent array
point path

$\hookrightarrow O(N)$

\hookrightarrow , if given query \rightarrow TLE
using binary lifting

Step 1 \rightarrow Bonita forced to do the same level
using binary lifting

Step 2 \rightarrow move a & b to every
unequal node (16 to 0)

Notes

03

Wednesday

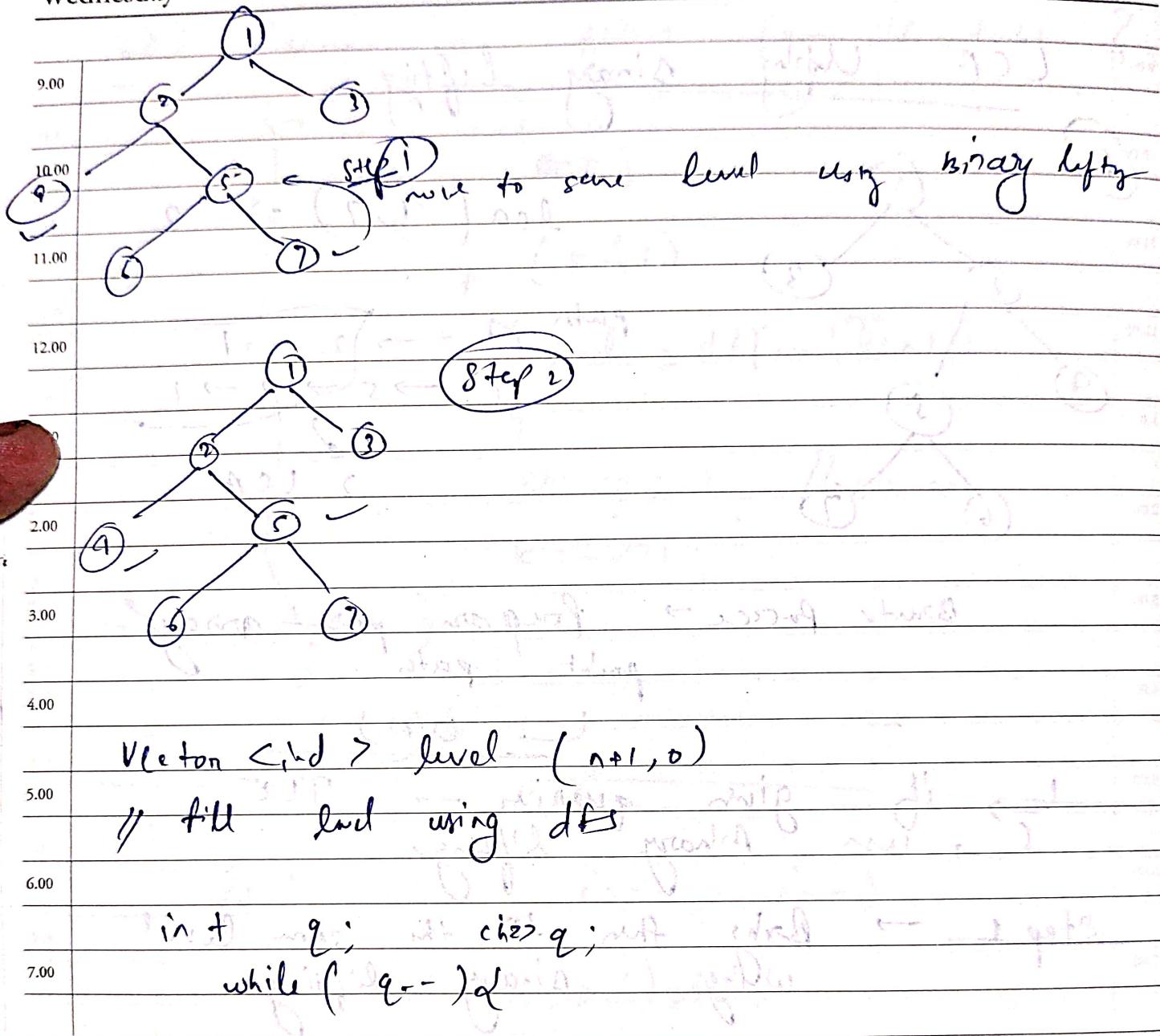
August 2022

Week 32

Day 215 - 150

Date 03 - 08 - 2022

AUG 22	M	T	W	T	F	S	S
	1	2	3	4	5	6	7
	8	9	10	11	12	13	14
	15	16	17	18	19	20	21
	22	23	24	25	26	27	28
	29	30	31				



Notes

```
if (level[a] > level[b])  
    swap(a, b);  
}
```

// a is on lesser level

If you don't drive your business, you will be driven out of business.

M	T	W	T	F	S	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

SEP 22

August 2022

Week 32

Day 216 - 149

Date 04 - 08 - 2022

04

Thursday

9.00 ht rc: lvel[b] - lvel[a];

10.00 b = getkpar(b, k); // move b to its kth parent using binary lifting

11.00 if(a == b) return a;

12.00 for (i = 16; i >= 0; i--)

13.00 if (dp[a][i] != dp[b][i])

14.00 a = dp[a][i];

15.00 b = dp[b][i];

16.00 // now this is emidial parent is LCA

17.00 count << dp[a][0] << endl;

18.00 }

5.00

6.00

7.00

Notes