Flip Array

Max sum Value

Ways to form Max Heap

1. Given an array A of +ve elements, flip sign of some elements such that resultant sum of array should be minimum non negative (0/+ve) (as close to 0 as possible)

Return min. no. of elements whose sign needs to be flipped such that resultant sum is minimum non negative

tsum = 20

A = [6 2 3 7 2]

ans
2

cnt	−	+	sum
	2	6, 3, 7, 2	16
	2, 6	3, 7, 2	4
3 →	2, 6, 2	3, 7	0
2 →	7, 3	2, 6, 2	0

A = [15, 10, 6]

ans
1

cnt	−	+	(0/+ve) sum
	15, 10	6	−19 ✗
1 →	15	10, 6	1
			0 ✗

A = [_ _ _ _ _]   tsum = 20

-ve	+ve	sum
sum → 10	sum → 10	0

A = [- . . - - - .]   tsum = N

-ve	+ve	sum
N/2	N/2	0

A = [_ _ _ _ _]   tsum = 20

-ve	+ve	sum
sum → 10	sum → 10	0 ×
9	11	2
8	12	4
7	13	6

Some -ve elements

sum → 10, 9, 8, 7 . . . .

sum of -ve ele as closest to 10 as possible

Pick -ve elements whose sum is as closest to $\frac{totalsum}{2}$

-ve
↓

Pick some ele such that sum $\leq \frac{totalsum}{2}$

↓

min cnt of ele

① Constraint → sum $\leq ts/2$

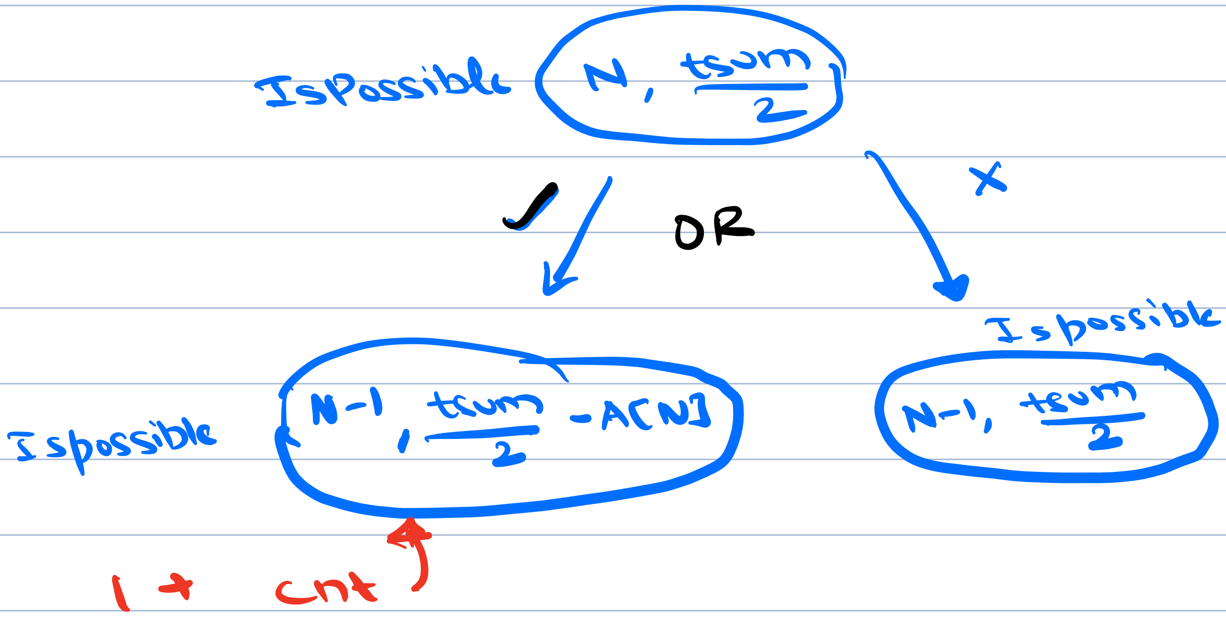
② Pick a subset

③ Min / Max → cnt of ele

A → [1 2 . . . . N] tsum

- ve

0



cnt of ele, possible to make sum or not

IsPossible $(N, \frac{ts}{2})$

ans
↓
2

$A = [\overset{0}{4} \quad \overset{1}{3} \quad \overset{2}{1} \quad \overset{3}{4}]$

$N = 4$

$\frac{ts}{2} = \frac{12}{2} = 6$

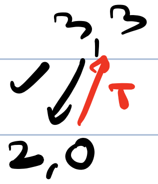
-ve	+ve	sum
6 x	6	0 x
5	7	2
<4,1>	<3,4>	

IsPossible $(4, 6)$

dp [N+1] [ts/2 +1]

dp [4 +1] [6 +1]

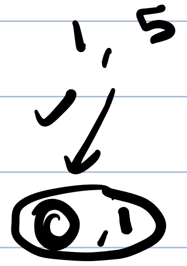
$\quad\quad\quad\quad \downarrow 5 \quad\quad \downarrow 7$

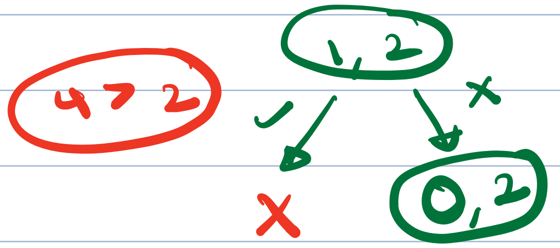


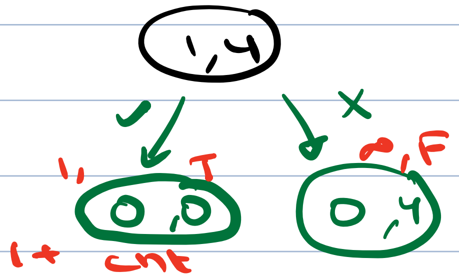
$A = [\overset{0}{4} \quad \overset{1}{4} \quad \overset{2}{3} \quad \overset{3}{1}]$

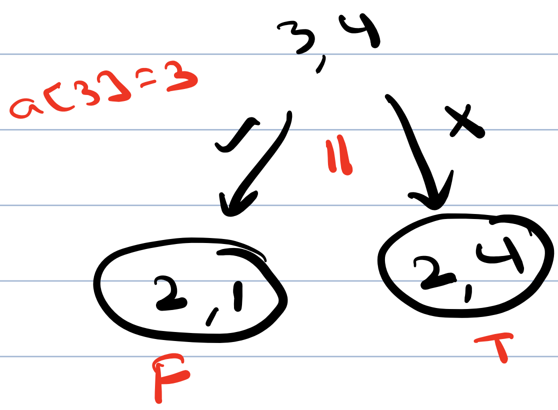
sum →

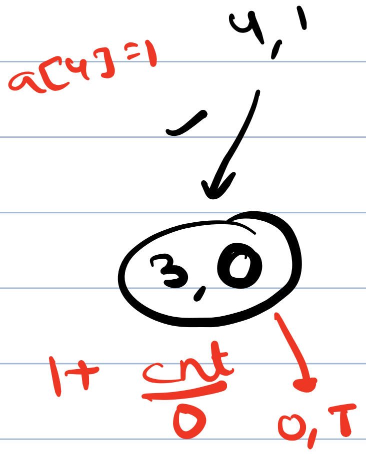
		0	1	2	3	4	5	6
	0	0 ,T	∞ ,F	∞ ,F	∞ ,F	∞ ,F	∞ ,F	∞ ,F
4	1	0 ,T	∞ ,F	∞ ,F	∞ ,F	1+0, T	∞, F	∞, F
4	2	0 ,T	∞, F	∞, F	∞, F	1, T	∞, F	∞, F
3	3	0 ,T	∞, F	∞, F	1, T	1, T	∞, F	∞, F
1	4	0 ,T	1, T	∞, F	1, T	1, T	2, T	∞, F

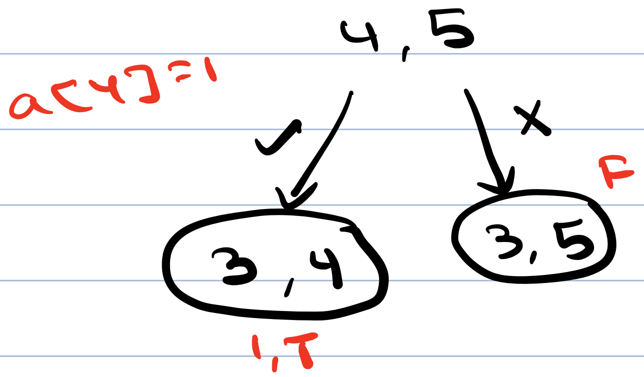


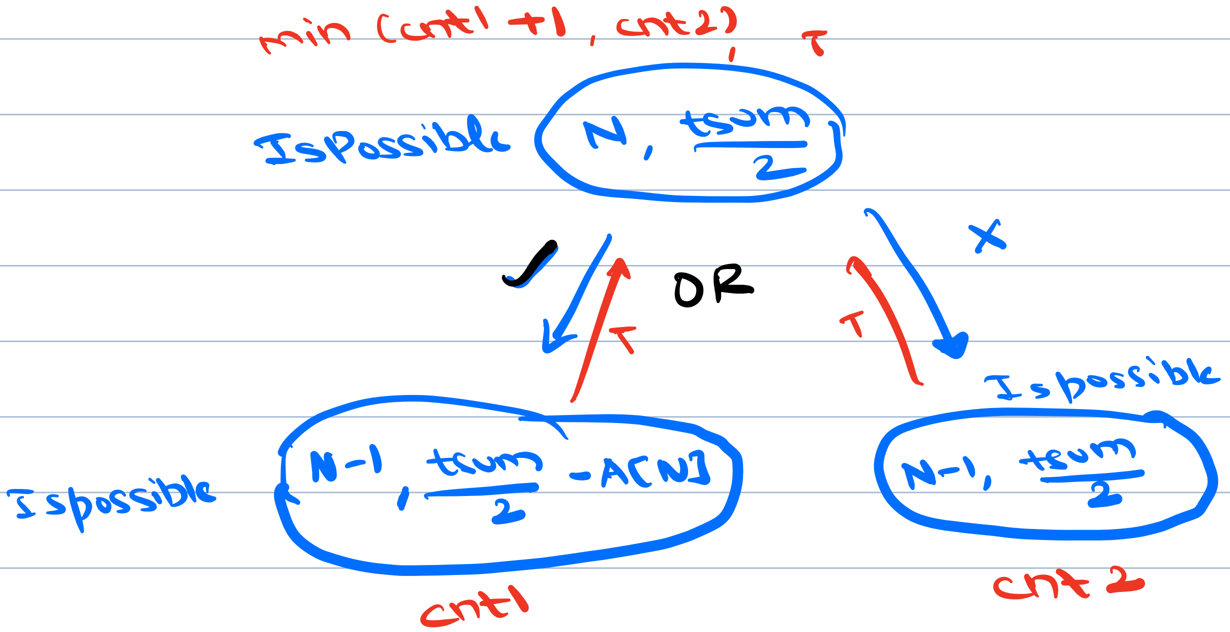


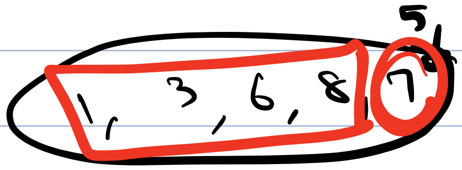


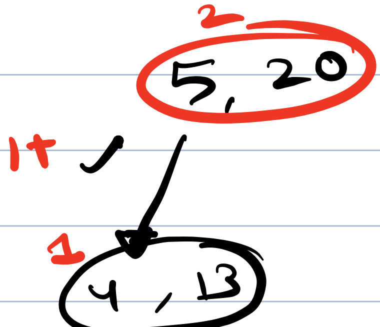

3,4
a[3]=3
✓
||
x
2,1
F
2,4
T


4,1
a[4]=1
✓
3,0
1+ cnt
0
0,T


4,5
a[4]=1
✓
x
3,4
1,T
3,5
F


min (cnt1 +1 , cnt2) , T
IsPossible
N, tsum/2
✓
OR
x
T
T
Ispossible
N-1 , tsum/2 - A[N]
cnt1
Ispossible
N-1, tsum/2
cnt 2


5,
1, 3, 6, 8, 7


2
5, 20
1+ ✓
1
4, 13

2. Given an array A of N integers and 3 integers B, C and D.

Maximize value of A[i] * B + A[j] * C + A[k] * D
where $1 <= i <= j <= k <= N$

$A = [\overset{0}{1}, \overset{1}{5}, \overset{2}{-3}, \overset{3}{4}, \overset{4}{-2}]$ $B = 2$ $C = 1$ $D = -1$

$i = 1$
$j = 1$
$k = 2$

$5 \times 2 + 5 \times 1 + (-3) \times (-1) = \boxed{18}$

i, j, k

B → i
C → j
D → k

```
for (i=0 ; i<N ; i++) {
    for(j=i ; j <N ; j++) {
        for(k=j ; k<N ; k++) {
        }
    }
}
```

TC: $O(N^3)$
SC: $O(1)$

$1 \le N \le 10^5$

N^3 works?

$\boxed{\text{NO}}$

Max

$[\ \underline{-10}\ \ \underline{1}\ \ \underline{4}\ \ \underline{-20}\ \ \underline{6}\ ]$, B $\qquad$ $A[i] * B$

$B = 5$ $\qquad$ $A[i] = 6$

$B = -5$ $\qquad$ $A[i] = -20$

$B > 0$ $\qquad$ $A[i] \rightarrow$ max

$B < 0$ $\qquad$ $A[i] \rightarrow$ min

0 $\quad$ 1 $\quad$ 2 $\quad$ 3 $\quad$ 4

$[\ \_\ \ \_\ \ 0\ \ \_\ \ \_\ ]$, Max $(A[i] * B + A[j] * C)$

j $\qquad$ j $\qquad$ $i \leq j$

↓ $\qquad$ ↓

$A[j] * C$ $\qquad$ $A[4] * C$

↓

Max $A[i] * B$ till ind j

0 $\quad$ 1 $\quad$ 2 $\quad$ 3 $\quad$ 4

[$\qquad$]

↓ $\qquad$ ↓

0 - 0 $\qquad$ 0 - 1

Max $\qquad$ Max

$A[i] * B$ $\qquad$ $A[i] * B$

0 - i $\rightarrow$ Max $A[i] * B$

B = 2   C = 1   D = -1

0 1 2 3 4

A = [1 5 -3 4 -2]

1st → A[i]   A[i] * B

val [i] = [2, 10, -6, 8, -4]

0 1 2 3 4

Premax [i] = [2, 10, 10, 10, 10]

↓

0 – i

2nd → A[j]   A[i] * B + A[j] * C

0 1 2 3 4

val [j] = [3, 15, 7, 14, 8]

2 + 1*1   10 + 5*1   10 + (-3)   10 + 4   10 + -2

premax [j] = [3, 15, 15, 15, 15]

↓

0 – i

3rd → A[K]   A[i] * B + A[j] * C + A[K] * D

Max

0 1 2 3 4

val [ ] → [2, 10, 18, 11, 17]

3 + -1   15 + -5   15 + 3   15 + -4   15 + 2

premax [ ] → [2, 10, 18, 18, 18]

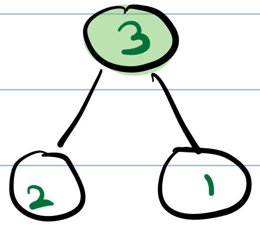
```
            // A[N], B,C,D
       val[N] = {0},  premax[N] = {0}

for (cnt=1 ; cnt ≤3 ; cnt++) {
      // val
    for (idx = 0 ; idx < N ; idx ++) {
      // contri of ith ele
        if (cnt ==1)          contri = A[idx] * B
            cnt ==2                    A[idx] * C
            cnt ==3                    A[idx]*D

        val[idx] = contri + premax[idx]
    }
      // premax
}

return premax[N-1]                  TC: O(N)
                                    SC: O(N)
```

10:54

3. Find the no. of distinct Max Heap that can be made from A distinct integers.

$1 \le A \le 100$

A = 3

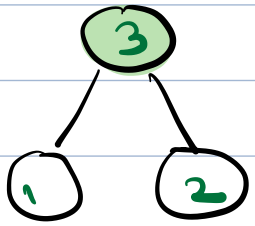


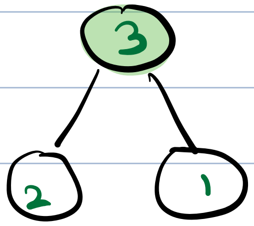
Max Heap

① CBT

② Par > child

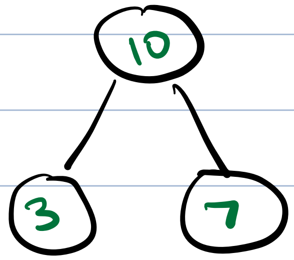
1, 2, 3

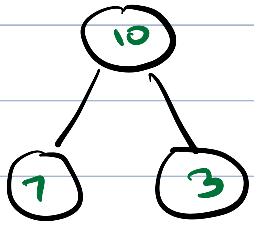




ans
2

3, 7, 10
1st 2nd 3rd

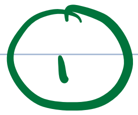




ans
2

A = 1

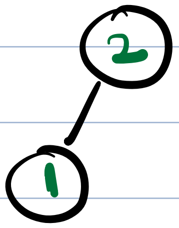
1



ans
1

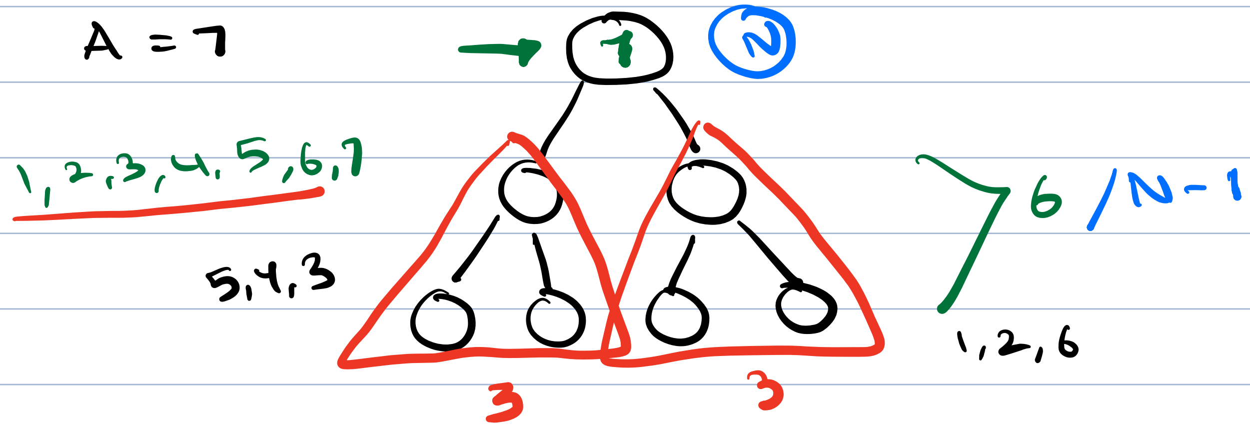
A = 2

1, 2

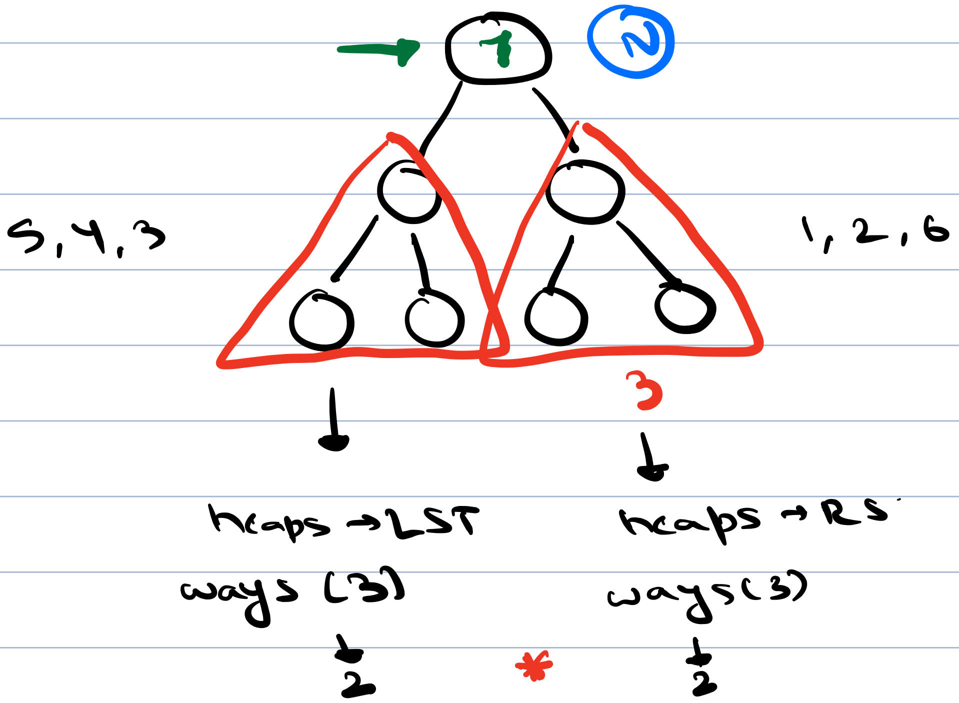


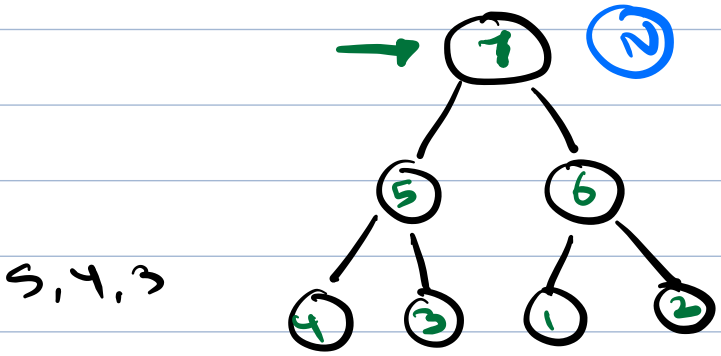
ans
1

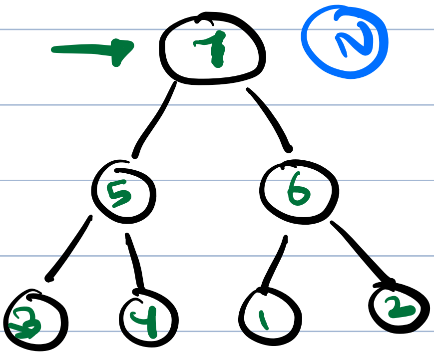
A = 7

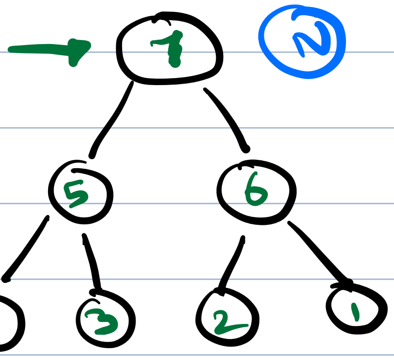


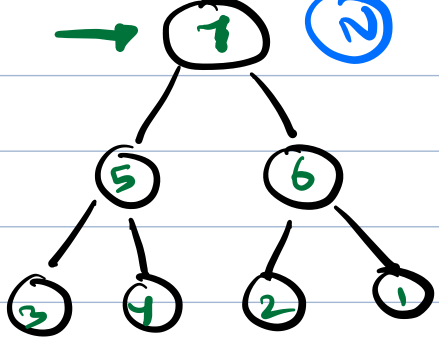
No relationship b/w elements in LST and RST



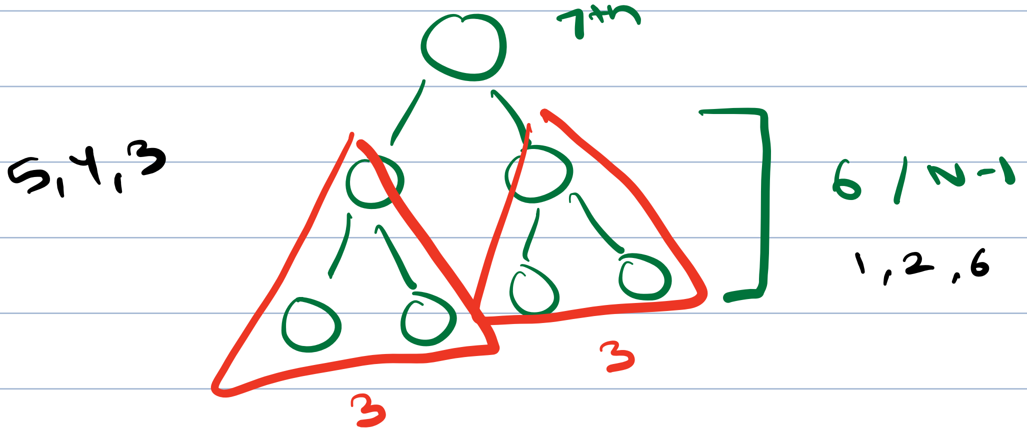








$$\boxed{\text{ways}(A) = {}^{N-1}C_L \times \text{ways}(L) * \text{ways}(R)}$$



$$\underset{\substack{\downarrow\\2}}{\text{ways}(3)} * \underset{\substack{\downarrow\\2}}{\text{ways}(3)}$$

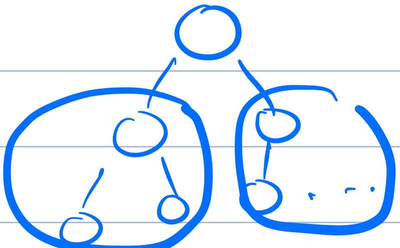
$$\boxed{A = N}$$

$$\boxed{\text{ways}(A) = {}^{A-1}C_L \times \text{ways}(L) * \text{ways}(R)}$$

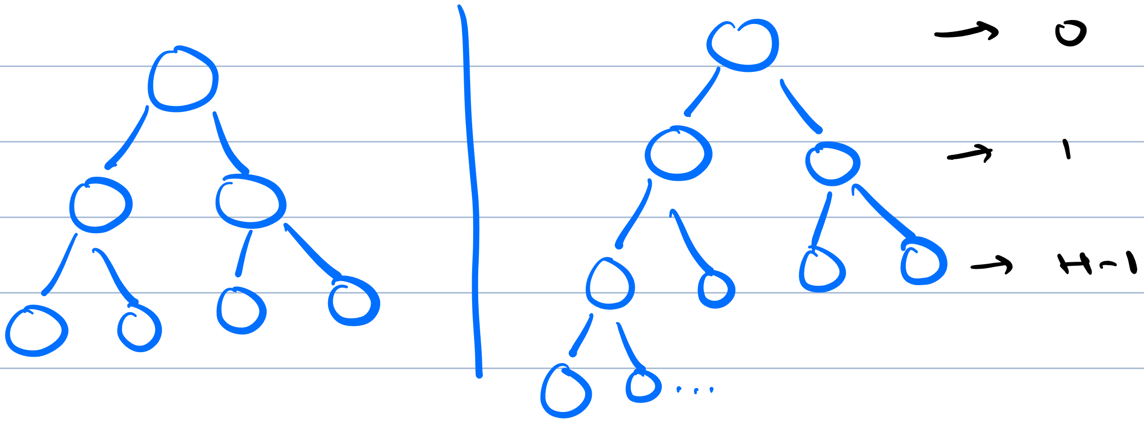
L = ? R = ?

① $A = L + R + 1$

② $L \geq R$ (bcoz of CBT)



③ $H = \lfloor \log_2 A \rfloor$ (bcoz of CBT)



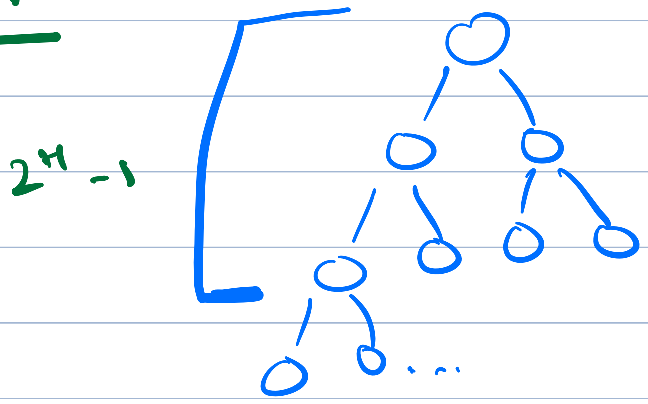
Last level might not be filled

All levels from 0 → H-1 will be filled.

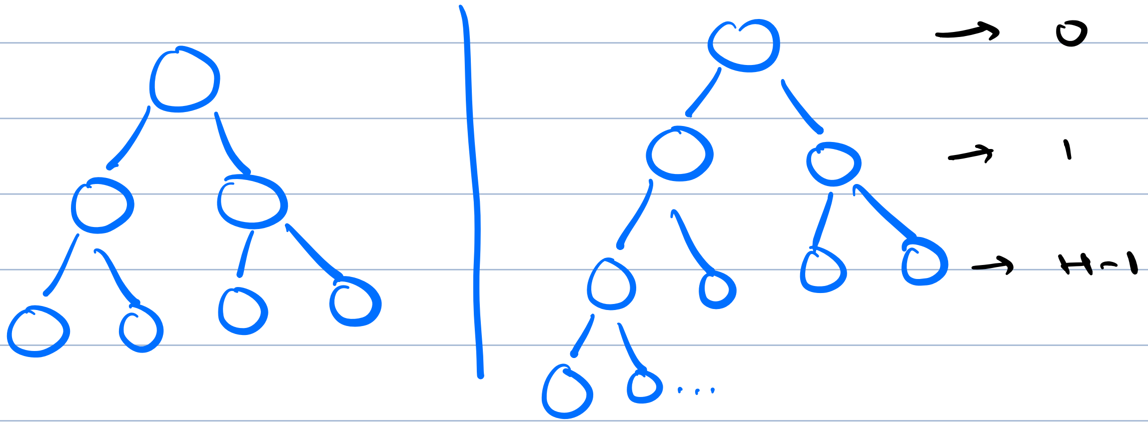
Total nodes from 0 → H-1 level $= 2^0 + 2^1 + 2^2 + \dots + 2^{H-1}$

$= 2^H - 1$

LST → 0 → H-1 level $= \dfrac{2^H - 1 - 1}{2}$



LST → Nodes in last level



H

↓

2^H

LST = $\frac{2^H}{2}$ (Best case

Actual nodes in last level = $A - (2^H - 1)$

LST → Nodes in last level = $\min\left(\frac{2^H}{2}, A-(2^H-1)\right)$

Total nodes in LST = $\frac{2^H-1-1}{2}$ +

(0 → H level)

$\min\left(\frac{2^H}{2}, A-(2^H-1)\right)$

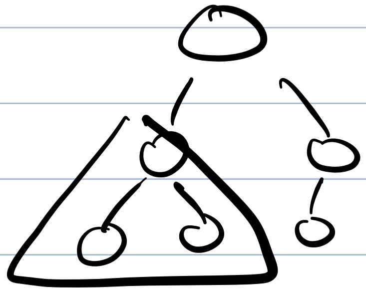
$A = L+R+1$

$\Rightarrow R = A-L-1$

A →

H → floor $(\log_2 A)$

L → using formula



R → A - L - 1

```
int dp[A+1] = {-1}
ways (A) {
    if (A =1 || A = 2)
                return 1
    if (dp[A] != -1) return dp[A]
      H = ⌊log_2 A⌋

      L = $\frac{2^H - 1 - 1}{2} + \min\left(\frac{2^H}{2}, A-(2^H-1)\right)$

      R = A-L-1

    dp[A] = $^{A-1}C_L$ × ways (L) × ways (R)

      return dp[A]
}
```

TC : O(A)

SC : O(A)