Agenda :

more 35% of problem Advanced DSA 1

49 problems

↓

> 17-18 problems

1. Tower of Hanoi
2. Subarray OR

Q2. Subarray OR

[7 4 8]

	OR Equivalent
7	7
7 4	7
7 4 8	15
4	4
4 8	12
8	8

Sum of this

53 ans

find sum of "OR of all subarrays"

2.1 given binary array, calculate the no of subarrays whose result OR value is 0.

{ 1, 1, 0, 0, 0, 1, 0 }
0 1 2 3 4 5 6

2 2
2 3
2 4
3 3
3 4
4 4
} 0

3 → $\frac{3 \times \cancel{4}^{2}}{\cancel{2}} = 6$

6 6 } 0

Observations:

BINARY ARRAY

OR value = 0, iff all elements are 0

{ 1, 1, 0, 0, 0, 1, 0 }
0 1 2 3 4 5 6

↓

c

$\left[\frac{c \times (c+1)}{2}\right]$

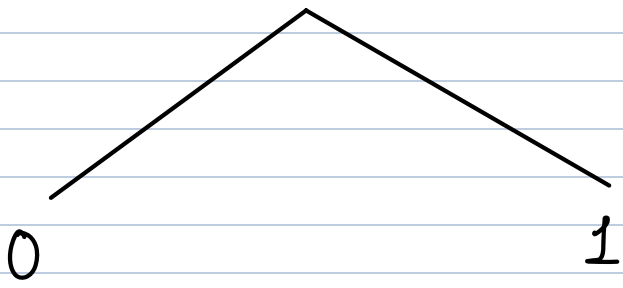
```
int c-subarray_OR_0 (int[] ba ) {
    int c = 0
    int ans = 0
    for ( i = 0        i < N    i++) {
        if ( ba[i] == 0) {
            c++
        }else {
            ans = ans + c x (c+1) / 2
            c = 0
        }
    }

    if ( c > 0) {
        ans = ans + c x (c+1) / 2
    }

    return ans
}
```

TC: O(N)

SC: O(1)

Q 2.2 given binary array, calculate the no of subarrays whose result OR value is 1.

{ 1, 1, 0, 0, 0, 1, 0 }
0 1 2 3 4 5 6



1 1 1 0 0

↓

1

No of Subarrays whose OR value is 1 = [Total Number of Subarrays] − [No of Subarrays whose OR value is 0]

Ques. 2.3 array of N integers.

Count subarrays whose 0th bit of OR result is 1

7 4 8

6 subarrays:

	OR Result	3 2 1 0
7	7	0 1 1 1
7 4	7	0 1 1 1
7 4 8	15	1 1 1 1
4	4	0 1 0 0
4 8	12	1 1 0 0
8	8	1 0 0 0

3

ar[0] 7 ···→ 1 1 1

ar[1] 4 1 0 0

ar[2] 8. 1 0 0 0

b[ ] = 1 0 0

0th bit of ar[0] | 0th bit of ar[1] | 0th bit of ar[2]

Count all subarrays whose 0th bit of OR Result is 1

7 4 = 7 | 4

1 1 1

1 0 0

1

ele involved → 0th bit of elements involved in subarray

OR result → 0th bit of OR result

arr [7 4 8] = how many subarrays of [7,4,8] → 0th bit of OR result is 1

0th bit [1 0 0] = how many subarrays of [1 0 0] will have OR result as 1

1

```
int c_sub_OR_0thbit_1 (int[] arr ){
    j = 0
    int[] ba = new int[arr.length]
    for( i=0  i<N  i++) {
        if (arr[i] & (1<<j) != 0) {
            ba[i] = 1
        }
    }
    ans = c_subarray_OR_1( ba );
    return ans
}
```

No of subarrays, whose OR result's 1st bit = 1

No of subarrays, whose OR result's 2nd bit = 1

⋮

31st bit is 1

7 4 8

7
7 4
7 4 8
4
4 8
8

OR result

7
7
15
4
12
8

3 2 1 0

1 1 1
1 1 1
1 1 1 1
1 0 0
1 1 0 0
1 0 0 0

3 5 3 3

$= 3 \times 2^3 + 5 \times 2^2 + 3 \times 2^1 + 3 \times 2^0$

$= 24 + 20 + 6 + 3$

$= 53$ ans

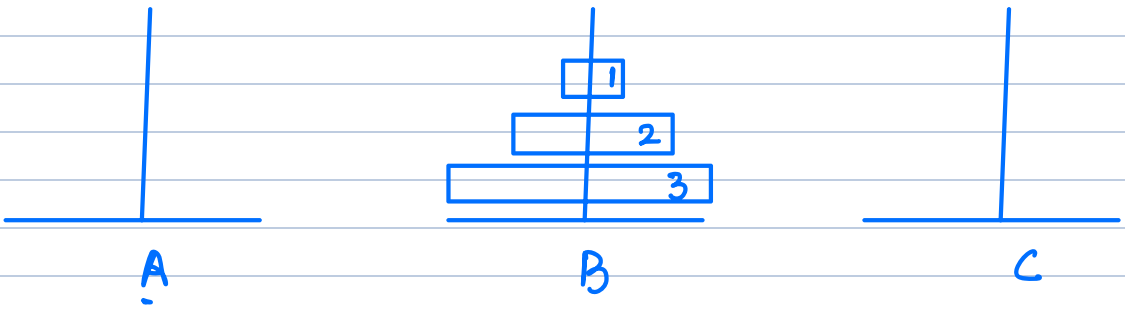
```
int sum_OR_subarray (int[] arr ){
    ans = 0
    for (j=0  j<31  j++) {
        int[] ba = new int[arr.length]
        for( i=0  i<N  i++) {
            if (arr[i] & (1<<j) != 0){
                ba[i] = 1
            }
        }
        C = c-subarray_OR_1( ba );
        ans = ans + c x 2^j;
    }
    return ans;
}
```

TC : $O(31N) = O(N)$

SC : $O(N)$

Q. Print instructions of moving N disks from src tower to dest tower, using hel tower,
A B C

following rules:
1. Move 1 disk at a time
2. Larger disk ~~X~~ over smaller disk



N=3

1,2 A to C
3 A to B
1,2 C to B

1	A	→ B
2	A	C
1	B	C
3	A	B ✓
1	C	A
2	C	B ✓
1	A	B ✓

7

1	A	B
2	A	C
1	B	C
3	A	B
1	C	A
2	C	B
1	A	B

P 1 2 3 A to B, using C

SP1 1 2 A to C, using B ✓

EW Move 3 A to B

SP2 1 2 C to B, using A

P

Print instructions of moving N disks from src tower to dest tower, using hel tower,
A B C
following rules

SP

SP1

Print instructions of moving N-1 disks from src tower to dest tower, using hel tower,
A C B
following rules

P = SP + EW

Print (Move N A → B)

SP2

Print instructions of moving N-1 disks from src tower to dest tower, using hel tower,
C B A
following rules

Dry Run

```
void toh ( N, A, B, C ) {
    toh ( N-1, A, C, B );
    Print ( Move N A → B );
    toh ( N-1, C, B, A );
}
```

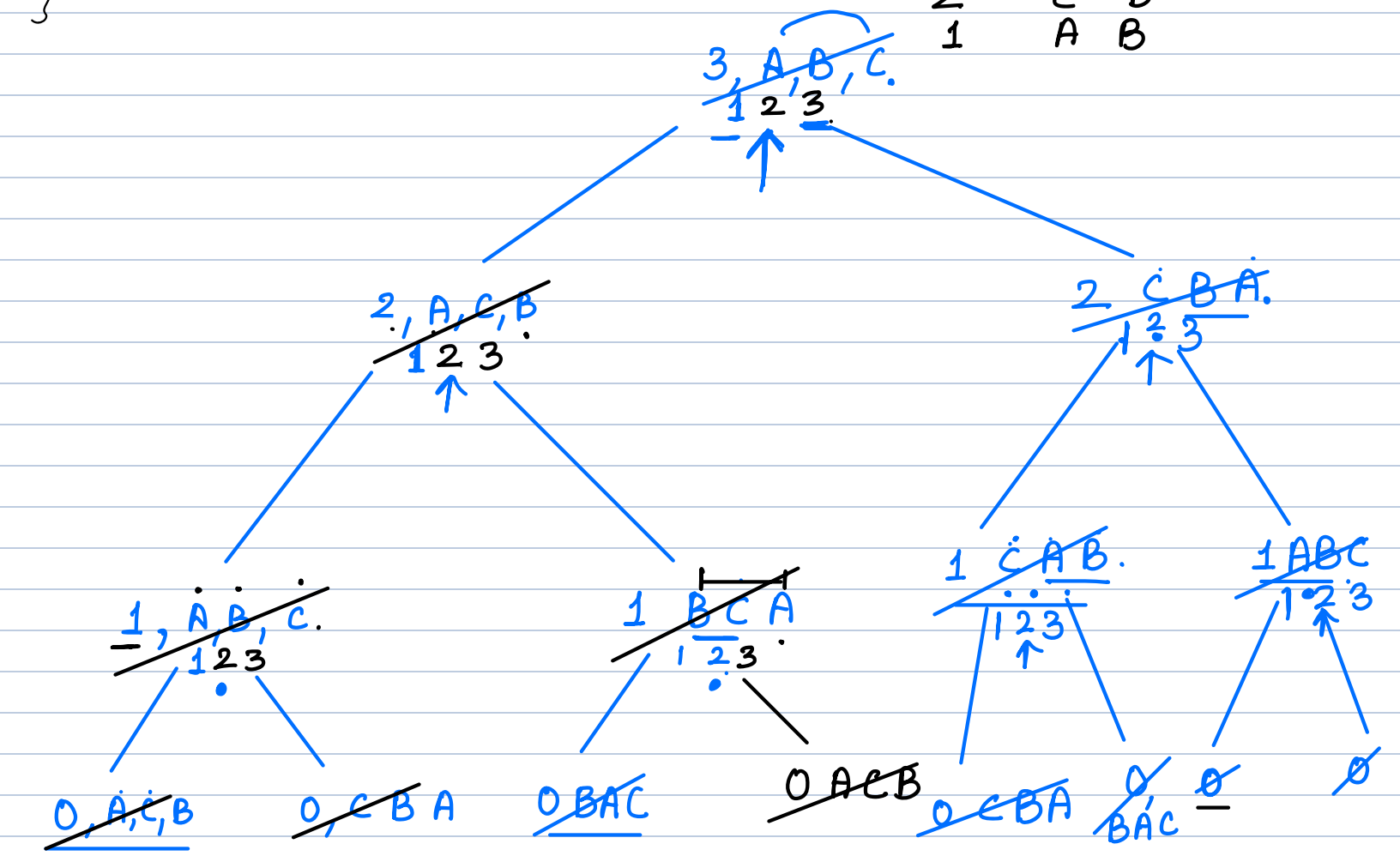
```
void toh ( N, A, B, C ) {
    if (N == 0) { return; }
    toh ( N-1, A, C, B );
    Print ( Move N A → B );
    toh ( N-1, C, B, A );
}
```

N = 3

1
2
3

Console :

1	A	B
2	A	C
1	B	C
3	A	B
1	C	A
2	C	B
1	A	B



1 ——————

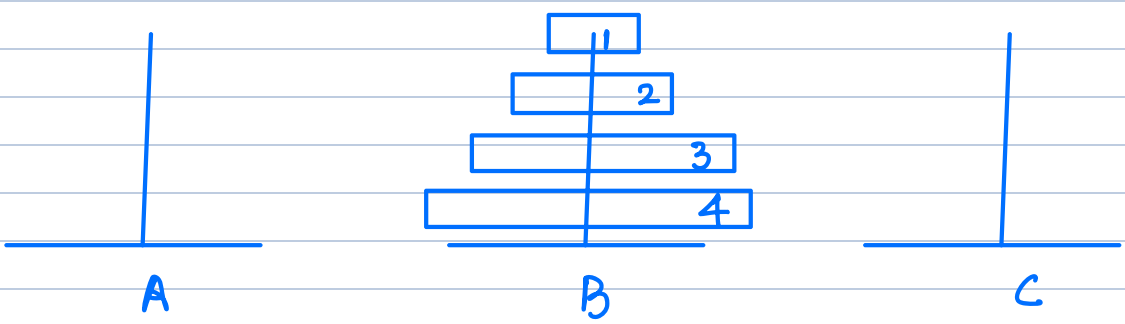
2 ↑ ↑

N = 4 4 disks A to B, C

3 A to C, B

→ 4 A to B

3 C to B, A.



SP1

123

A to C, B

1	A C ✓
2	A B
1	C B
3	A C ✓
1	B A
2	B C ✓
1	A C ✓

SP2

1	C B
2	C A
1	B A
3	C B
1	A C
2	A B
1	C B

Move 4th A to B