Alien's Dictionary

A : ["hello" , "scaler" , "interviewbit"]

B: "adhbcfegskjlponmirqtxwuvzy"

s1 = hello , s2 = hey

S1 < S2

```
bool  compare (s1 , s2) {
        i=0, j=0
        while( i< s1.size() && j< s2.size()){
            if (s1[i] < s2[j])
                    return true;
            else if (s1[i] > s2[j])
                    return false;
            else {
                    i++, j++;
            }
        }
        i == s1.size() ? true : false;
}
```

ASCII value's are compared.

fam , family

↑ ↑

i=l1 , j!=l2

S1<S2

true

goldfish , gold

↑ i ↑ j

i!=l1 , j=l2

False

abc , abc

↑ i ↑ j

i=l1 , j=l2

S1=S2

true.

B: "adhbcfegskjlponmirqtxwuvzy"

a → 1
d → 2
h → 3
b → 4
c → 5
f → 6
e → 7
g → 8
⋮
y → 26

HashMap< Char, int > map.

```
bool compare (s1, s2) {
    i=0, j=0
    while( i< s1.size() && j< s2.size() ){
        if (map[s1[i]] < map[s2[j]])
            return true;
        else if (map[s1[i]] > map[s2[j]])
            return false;
        else {
            i++, j++;
        }
    }
    i == s1.size() ? true : false;
}
```

A : ["hello", "scaler", "interviewbit"] ⇒ N

```
for ( i=0; i< N-1; i++){
    if (! compare( A[i], A[i+1])){
        return false;
    }
}
return true;
```

$$\boxed{\begin{array}{l} TC: O(N \cdot l) \\ SC: O(1) \end{array}}$$

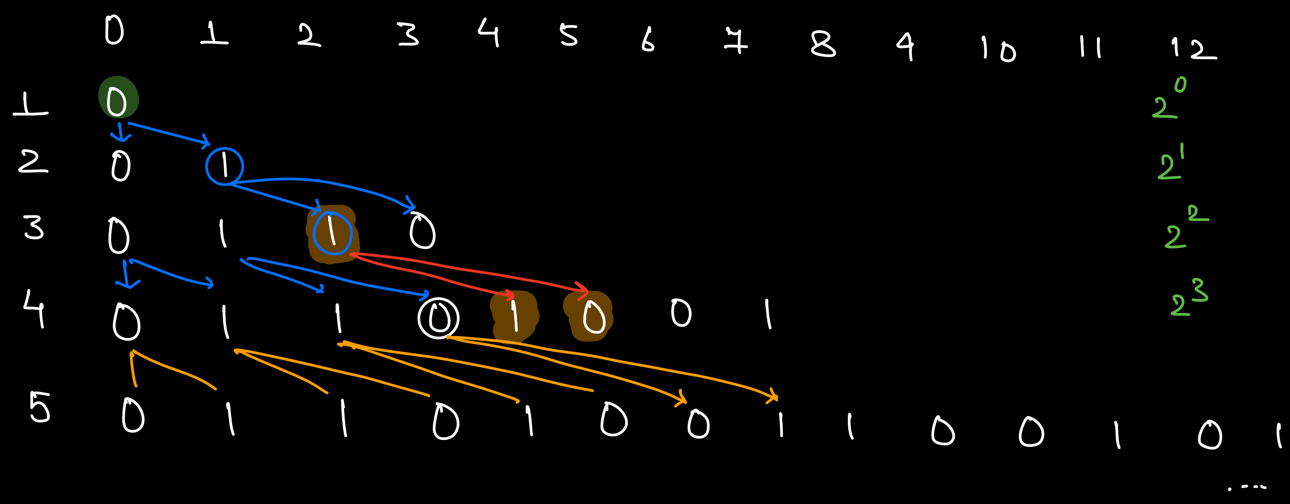
Q. K^th Char

→ Each row is generated by replacing all elements from previous row from

$0 \rightarrow \{0\ 1\}$

$1 \rightarrow \{1\ 0\}$

→ Given N & K, find K^th character in N^th row.



	0	1	2	3	4	5	6	7	8	4	10	11	12		
1	0														2^0
2	0	1													2^1
3	0	1	1	0											2^2
4	0	1	1	0	1	0	0	1							2^3
5	0	1	1	0	1	0	0	1	1	0	0	1	0	1	

$N = 5, K = 8 \Rightarrow 1$

$N = 5, K = 10 \Rightarrow 0$

$$N \Rightarrow \underbrace{2^0 + 2^1 + 2^2 + \cdots\cdots + 2^{N-1}}_{\text{G.P}}$$

$a = 1$

$r = 2$

$N = N$

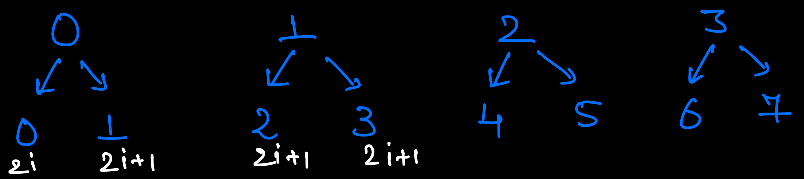
$$\text{Sum} = \frac{1\cdot(1-2^N)}{1-2} = 2^N - 1$$

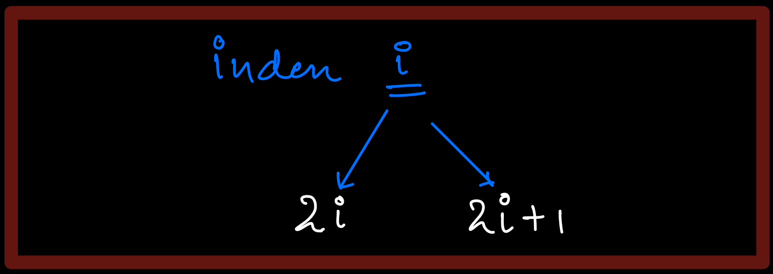
$$\boxed{TC: O(2^N)}$$

Constraints.

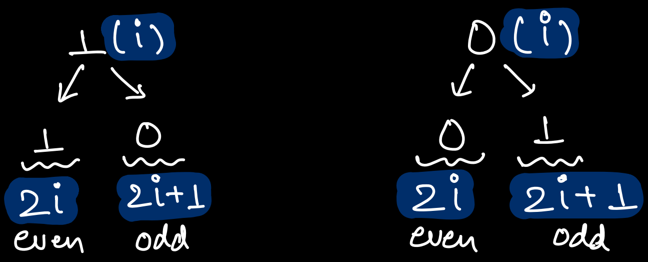
$N <= 10^5$

Index :





index $\underline{\underline{x}}$ ⇒ Parent $\frac{x}{2}$



Element at even index ⇒ same as parent

Element at odd index ⇒ $\sim$(Parent element)

N = 5, K = 8 ⇒ 1

↳ N = 4, K = 4 ⇒ 1

↳ N = 3, K = $\underline{\underline{2}}$ ⇒ 1

↳ N = 2, K = 1 ⇒ $\sim 0 = 1$

↳ N = 1, K = 0

$\underline{\underline{0}}$

```
int  KthChar ( N, K ) {
    if(K==0) return 0;
    parent = K/2
    par_value = KthChar(N-1, parent);
     if(K%2 == 0) {
          return par_value;
     }
     return   1-par_value;
}
```

Q. Given a Binary Array, Calculate the no. of subarrays whose OR = 0

↳ Bitwise OR of all subarray elements = 0.

$$A : \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 & 0 \end{array}$$

$$\left.\begin{array}{l} [1-1] \\ [3-3] \\ [4-4] \\ [3-4] \end{array}\right\} 4$$

Observation :

In subarray with bitwise OR = 0, there shouldn't be any 1.

$$A : \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \times & \times & & & & \times & & \times & & \end{array}$$

$$\frac{3(3+1)}{2} \qquad \frac{1(1+1)}{2} \qquad \frac{2(2+1)}{2}$$

$$6 \qquad 1 \qquad 3 \Rightarrow \underline{\underline{10}}$$

In an array of size N ⇒

$$\# \text{ of subarrays} = \frac{N(N+1)}{2}$$

```
ans = 0
c = 0
for( i = 0; i < n; i++ ) {
        if ( a[i] == 0 )
                c++
        else {
                ans += $\frac{c(c+1)}{2}$
                c = 0
        }
}
return ans;
```

TC : O(N)
SC : O(1)

No. of subarrays with Bitwise OR = 1

$$= \frac{n(n+1)}{2} - \underline{\underline{ans}}.$$