

Recursion: brute force approach is
start string to base condition
build it up, bottom to top
or largest string (start base case
up).

Problem is start string S_n is

previous S_{n-1} to build up S_n to

$\boxed{S_{n-1}}$ + "1" + $\boxed{\text{rev}(S_{n-1})}$ to start up.
left right

Similar to segment tree walk algorithm
first try to find on the left or right

Algorithm:

when k left & right, k same,

S_{n-1} or $\text{rec}(n-1, k)$ call up.

when k middle right, not start

bit ରେ 1.

k right 1 ରେ, length-right ପାଇଁ

call 275 ଯେଉଁ string reverse,

$S_i == S_{n-i}$. bit ଯଦି odd time flip

275, ୩୨୩ ରେ ଯେଉଁ ସିରିଜ୍ 32୭୫୫୫୫୫

୩୨୩ ୨୮ ସିରିଜ୍ ତିଆରି.

```
class Solution {
public:
    char findKthBit(int n, int k) {
        int invertCount = 0;
        int len = (1 << n) - 1; // Length of Sn is 2n - 1

        while (k > 1) {
            // If k is in the middle, return based on
            inversion count
            if (k == len / 2 + 1) {
                return invertCount % 2 == 0 ? '1' : '0';
            }

            // If k is in the second half, invert and
            mirror
            if (k > len / 2) {
                k = len + 1 - k; // Mirror position
                invertCount++; // Increment inversion
                count
            }

            len /= 2; // Reduce length for next iteration
        }

        // For the first position, return based on
        inversion count
        return invertCount % 2 == 0 ? '0' : '1';
    }
};
```

