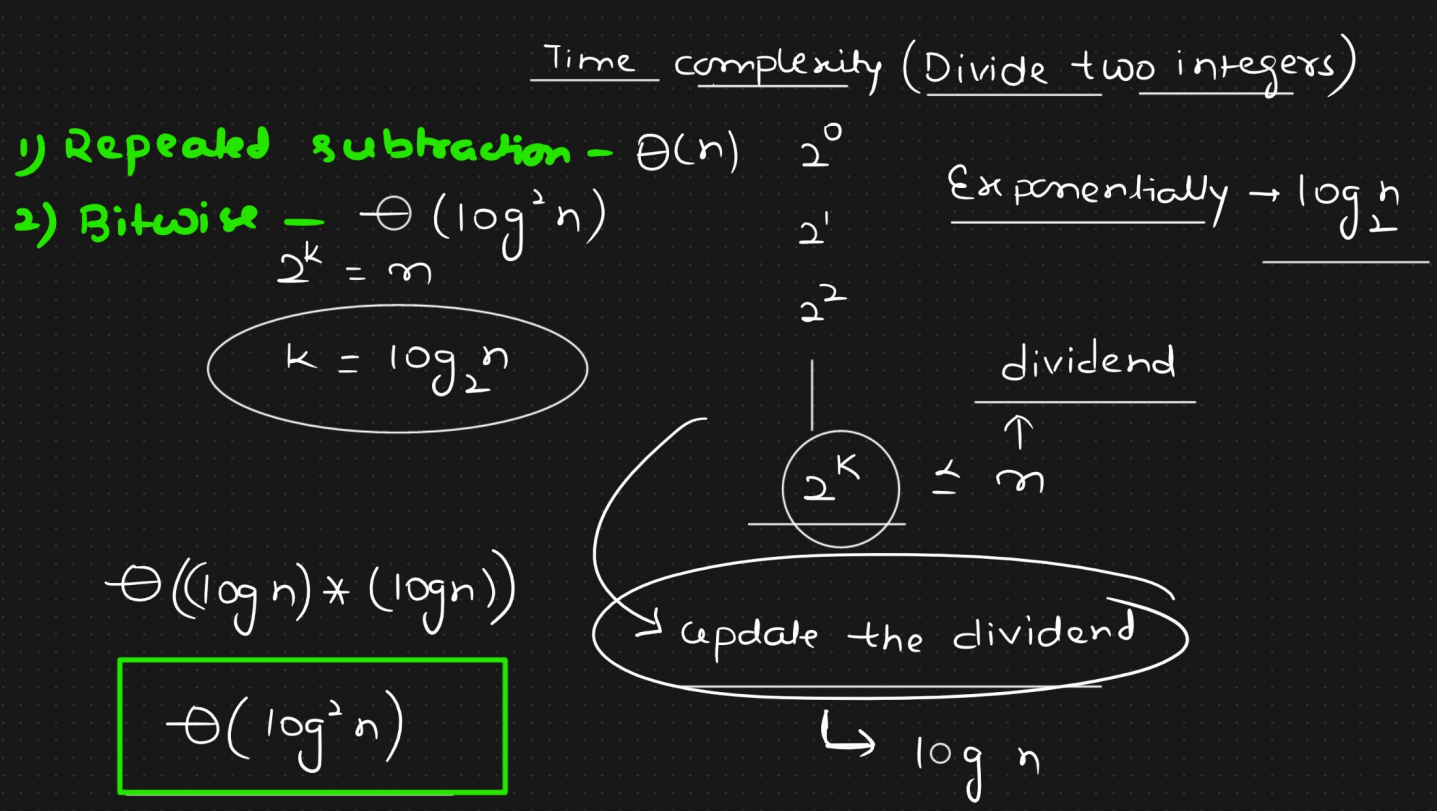

Space complexity = Θ(1)

Let n be the absolute value of dividend.

Time Complexity : O(log^2 n)

We started by performing an exponential search to find the biggest number that fits into the current dividend. So, for this time complexity is O(log n) operations

After doing this search, we updated the dividend by subtracting the number we found.

In the worst case, we were left with a dividend slightly less than half of the previous dividend (if it was more than half, then we couldn't have found the maximum number that fit in by doubling!).

So how many of these searches did we need to do?

Well, with the dividend at least halving after each one, there couldn't have been more than O(log n) of them.

So combined together, in the worst case, we have O(log n) searches with each search taking O(log n) time.

This gives us O((log n) * (log n)) = O(log^2 n)

Space Complexity : O(1)