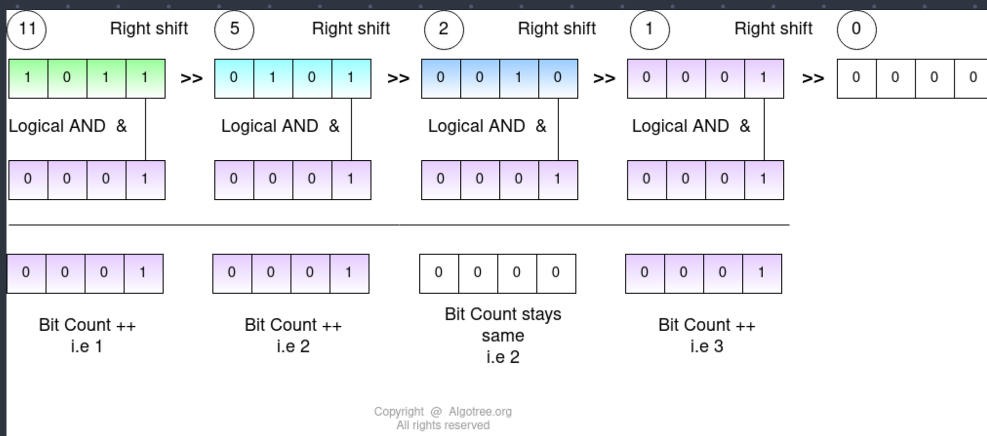
# Counting Set Bits

→ **Naive Approach :-**

84 → 1010100

We will use right shift operator with logical '&' to keep count of set bits untill the no. becomes 0.




$$\boxed{\begin{array}{l}\text{Time Complexity : } O(\log(N)) \\ \text{Space Complexity: } O(1)\end{array}}$$

→ **Brian Kernighan's Algorithm :-**

$n = 1010100$ $\qquad\qquad$ $n \,\&\, (n-1)$

$\quad - 1$

$(n-1) = \underbrace{1010}_{\text{same}}\ \underbrace{011}_{\text{flip}}$

$n \,\&\, (n-1) = 1010000$

Repeat this same step

$n = 1010000$

$\quad - 1$

$(n-1) = 1001111$ $\qquad\qquad$ $n\&(n-1) = 1000000$

$n = 1000000$

$n-1 = \underline{\quad\quad\quad\quad -1}$

$0111111$

$n \,\&\, n-1 = 0$

⇒ There are 3 set bits since the loop ran 3 times.

$\boxed{\begin{array}{l} \text{Time Complexity: } O(n) \\ \text{Space complexity: } O(1) \end{array}}$ no. of set bits

→ Builtin Method :-

```
__builtin_popcount(84);   // 3
```

$\boxed{\begin{array}{l} \text{Time Complexity: } O(1) \\ \text{Space Complexity: } O(1) \end{array}}$