Power of 2

231. Power of Two

Easy 5540 370 Add to List Share

Given an integer `n`, return `true` *if it is a power of two. Otherwise, return* `false`.

An integer `n` is a power of two, if there exists an integer `x` such that `n == 2^x`.

Example 1:

```
Input: n = 1
Output: true
Explanation: 2^0 = 1
```

Example 2:

```
Input: n = 16
Output: true
Explanation: 2^4 = 16
```

Example 3:

```
Input: n = 3
Output: false
```

1st approach (Basic)

```
twos = 2
while (twos != N)
    if (twos > N) return false;
    twos = twos * 2
```

3

2 != 16

if (2 > 16) False

two = 2*2 => 4

4 != 16

two = 4*2 = 8

8 != 16

two = 8*2 = 16 == 16

True

return True

Bit Manipulation

Binary of every power of two

1 = 1
2 = 10
4 = 100
8 = 1000
16 = 10000
- - - -

Binary of every power of $2^i - 1$ will be

1 = 1
3 = 11
7 = 111
15 = 1111

and if we perform & operation then

```
  1 0 0
& 0 1 1
-------
  0 0 0  = 0
```

so if we got n which is power of 2 then and if perform and operation with n-1 then we will get 0 always

example

```
7 → 0 1 1 1
8 → 1 0 0 0
  &
---------
    0 0 0 0  ⇒ 0
```

i.e power of 2 else not power of 2

```
class Solution {
public:
    bool isPowerOfTwo(int n) {
        if(n<=0) return false;
        return ((n & (n - 1))==0);
    }
};
```

T.C - O(1)
S.C - O(1)