**Approach: Looping**

Keep dividing it by 2 until 1 reached.

If 1 not reached, not a power of 2

Code:

if n<=0:

return False

while n%2 == 0:

n = n // 2

if n==1:

return True

return False

Time-> O(logn)

Space-> O(1) - Stack

**Approach: Recursion**

if n<=0:

return False

if n==1:

return True

return self.isPowerOfTwo(n/2)

Time-> O(logn)

Space-> O(logn) - Stack

**Approach 3: log function:**

logbase 2 of n must be an integer so that when 2 is raised to that integer it gives n

return n > 0 and log2(n) == trunc(log2(n))

Time Complexity : O(logn)

Space Complexity : O(1)