

Power of Two

Given an integer, write a function to determine if it is a power of two.

Credits:

Special thanks to [@jianchao.li.fighter](#) for adding this problem and creating all test cases.

Solution 1

Power of 2 means only one bit of n is '1', so use the trick $n \& (n-1) == 0$ to judge whether that is the case

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

written by [dong.wang.1694](#) original link [here](#)

Solution 2

This is kind of cheating, but the idea is that a power of two in binary form has and only has one "1".

```
public class Solution {  
    public boolean isPowerOfTwo(int n) {  
        return n > 0 && Integer.bitCount(n) == 1;  
    }  
}
```

written by [qxx](#) original link [here](#)

Solution 3

I used $n \& (n-1)$, but, attention the 0.

```
class Solution {  
public:  
    bool isPowerOfTwo(int n) {  
        return n > 0 && !(n & (n-1));  
    }  
};
```

written by [iamshanshanpt](#) original link [here](#)

From [LeetCoder](#).