**Bleak Numbers: -**

**Medium Accuracy: 49.12% Submissions: 43K+ Points: 4**

Given an integer, check whether it is **Bleak or not**.

A number n is called Bleak if it **cannot**be represented as sum of a positive number x and set bit count in x, i.e., x + [countSetBits(x)](http://www.geeksforgeeks.org/count-set-bits-in-an-integer/) is not equal to n for any non-negative number x.

**Example 1:**

**Input:**4

**Output:**1

**Explanation:**There is no x such that x + countSetbit(x) = 4

**Example 2:**

**Input:**   
3

**Output:**0

**Explanation:**   
3 is a Bleak number as 2 + countSetBit(2) = 3.

**Your Task:**  
You don't need to read or print anything. Your task is to complete the function **is\_bleak()** which takes n as input parameter and **returns 1** if n is **not**a Bleak number otherwise **returns 0**.  
 **Expected Time Complexity:**O(log(n) \* log(n))  
**Expected Space Complexity:**O(1)  
   
**Constraints:**  
1 <= n <= 109

**Code: -**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution

{

public:

int is\_bleak(int n)

{

// Code here.

for(int i = max(1, n-32); i <= n; ++i){

if(i + csb(i) == n)

return 0;

}

return 1;

}

int csb(int n){

int count = 0;

while(n > 0){

n = n & (n-1);

++count;

}

return count;

}

};

//{ Driver Code Starts.

int main(){

int T;

cin >> T;

while(T--)

{

int n;

cin >> n;

Solution ob;

int ans = ob.is\_bleak(n);

cout << ans << "\n";

}

return 0;

}

// } Driver Code Ends

**T.C: - O(log(N) \* log(N))**

**S.C: - O(1)**