Consecutive 1's in Binary Numbers

Sort 75 Discussions, By: votes

Search comments...

B I H | Preview

AbhishekVermaIIT 3 years ago

The problem can be solved most efficiently using bit-manipulation. Once again the ed disappointed me. "Discussions" has better solutions than the editorial. Anyways, the problem can be solved in **O(b)** complexity, where **b=consecutive 1's** :

```
n = int(input())
ans = 0
while n>0 :
    n &= (n<<1)
    ans += 1
print(ans)
```

Logic :

"Logical-and" accompanied with "left-shift" reduces consecutive set-bits(1's) by 1. Thus counting these operations gives the required answer.

14 | Add Comment Permalink

nitinshk1111 3 years ago

[deleted]

-1 | Add Comment Parent Permalink

srujan_madupu 3 years ago

good logic...even i didn't get initially..is there any resourse to become STRONG manipulation. thank you

-1 | Add Comment Parent Permalink

AbhishekVermaIIT 3 years ago

I can't comment about resource in particular. Though, HackerRank h
dedicated to Bit-manipulation. You can start practicing there. At time
awesome posts in discussions. Start reading them, a lot can be learn

3 | Add Comment Parent Permalink

werner 3 years ago

Abhishek your solutions makes me feel like i have lot to do in programming a
helping me to push a little bit in acquiring programming knowledge.thank you

2 | Add Comment Parent Permalink

AbhishekVermaIIT 3 years ago

Thanks a lot for appreciating the effort, makes it worth :)

0 | Add Comment Parent Permalink

Kj6995 3 years ago

i tried using the n &= (n-1) but this doesn't seem to work...can anyone please
does the same as mentioned in this

0 | Add Comment Parent Permalink

AbhishekVermaIIT 3 years ago

There's slight difference. The mentioned `n &= (n-1)` operation actu
bits. This problem requires **maximum consecutive** set-bits.

Example :

For binary `1110011`, the number of set-bits are 5 (this will be result
operation). But, the required answer would be 3, since maximum nu
set-bits are 3.

-2 | Add Comment Parent Permalink

Kj6995 3 years ago

oh thanks didn't read the question properly

0 | Add Comment Parent Permalink

anonymousGod 3 years ago

@AbhishekVermaIIT Hey, coming from java I'm not able to get what are you tr

n &= (n<<1) and your logic of "left-shift". Can you plese explain with a small ex

13? Thank you very much.

-1 | Add Comment Parent Permalink

AbhishekVermaIIT 3 years ago

Other than I/O, all operations stay as it is, in Java. Here's Java version

```
public static void main(String[] args) {
    Scanner scan = new Scanner(System.in) ;
    int n = scan.nextInt() ;
    scan.close() ;

    int ans = 0 ;
    while (n > 0) {
        n &= (n<<1) ;
        ans += 1 ;
    }
    System.out.println(ans) ;
}
```

Since you want explanation with an example, I'll explain it with `221` l

much. Now, binary represention of `221` is `11011101`. Visualizing ite

`loop` using binary clarifies the procedure :

```
First Iteration :   n           11011101
                  n<<1         110111010
              n & (n<<1)        10011000

Second Iteration :  n           10011000
                  n<<1         100110000
              n & (n<<1)           10000

Third Iteration :   n               10000
                  n<<1              100000
              n & (n<<1)                0
```

As the above example shows, number of consecutive `1`'s in **n** reduce

iteration of `while-loop`. Thus, counting the iterations, gives the rec

6 | Add Comment Parent Permalink

anonymousGod 3 years ago

I wasn't aware of such operation(n<<1) in java, have lot to le

2 | Add Comment Parent Permalink

Venkat30119 3 years ago

Bro small dobut but with out Dividing how we can get that b

(n%2) and etc...can u clarify it plz!!!

0 | Add Comment Parent Permalink

d_shubham777 1 year ago

very well explained...thank you AbhishekVermaIIT

1 | Add Comment Parent Permalink

sahilsabarwal50 2 years ago

[deleted]

0 | Add Comment Parent Permalink

NagaNaresh 2 years ago

superb logic man...

1 | Add Comment Parent Permalink

Sid_S 2 years ago

[deleted]

0 | Add Comment Parent Permalink

Kanahaiya 3 months ago

Here is the video explanation of the same-

Length Of The Longest Consecutive 1s In Binary R
A Number | BitManipulation

-2 | Add Comment Parent Permalink

pradeepyadava 3 years ago

```
public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int count=0;
        int result=0;
        while(n!=0)
            {
            if(n%2==1)
                {
               count++;
               result=fun(count,result);
            }
            else
                {
               result=fun(count,result);
                count=0;
            }
            n=n/2;
        }
        System.out.println(result);
}
    public static int fun(int a,int b)
        {
        if(b<a)
          b=a;
        return b;
    }
}
```

1 | Add Comment Permalink

raghav_rao 3 years ago

great ! how did you come with it? i mean the what was the thought process b
the logic?plz do reply!

0 | Add Comment Parent Permalink

bannurahul321 3 years ago

bro i didnt get the usage of function !

0 | Add Comment Parent Permalink

paul_hunter007 2 years ago

use of function is to store biggest no. consecutive ones in re

0 | Add Comment Parent Permalink

krobin_93 3 years ago

This one is more optimized solution. There is no need to compare count and my solution) for every iteration of while loop. Hope it helps.

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int max=0,count=0;
        while(n>0){
            if(n%2==1){
                count++;
            }else{
                max=Math.max(count,max);
                count=0;
            }
            n=n>>1; // same as n=n/2
        }
        System.out.println(Math.max(count,max));
    }
}
```

0 | Add Comment Parent Permalink

bhuvanms 3 years ago

complexity of the soulution?

0 | Add Comment Parent Permalink

krobin_93 3 years ago

Since we are checking every bit of the integer, the time complexity wi
number of bits/digits in the binary representation of the number n. H

0 | Add Comment Parent Permalink

AbhishekVermaIIT 3 years ago

It's good to see comments like these. I was reluctant to write
comment convinced me to write it finally. Have a look at this

-1 | Add Comment Parent Permalink

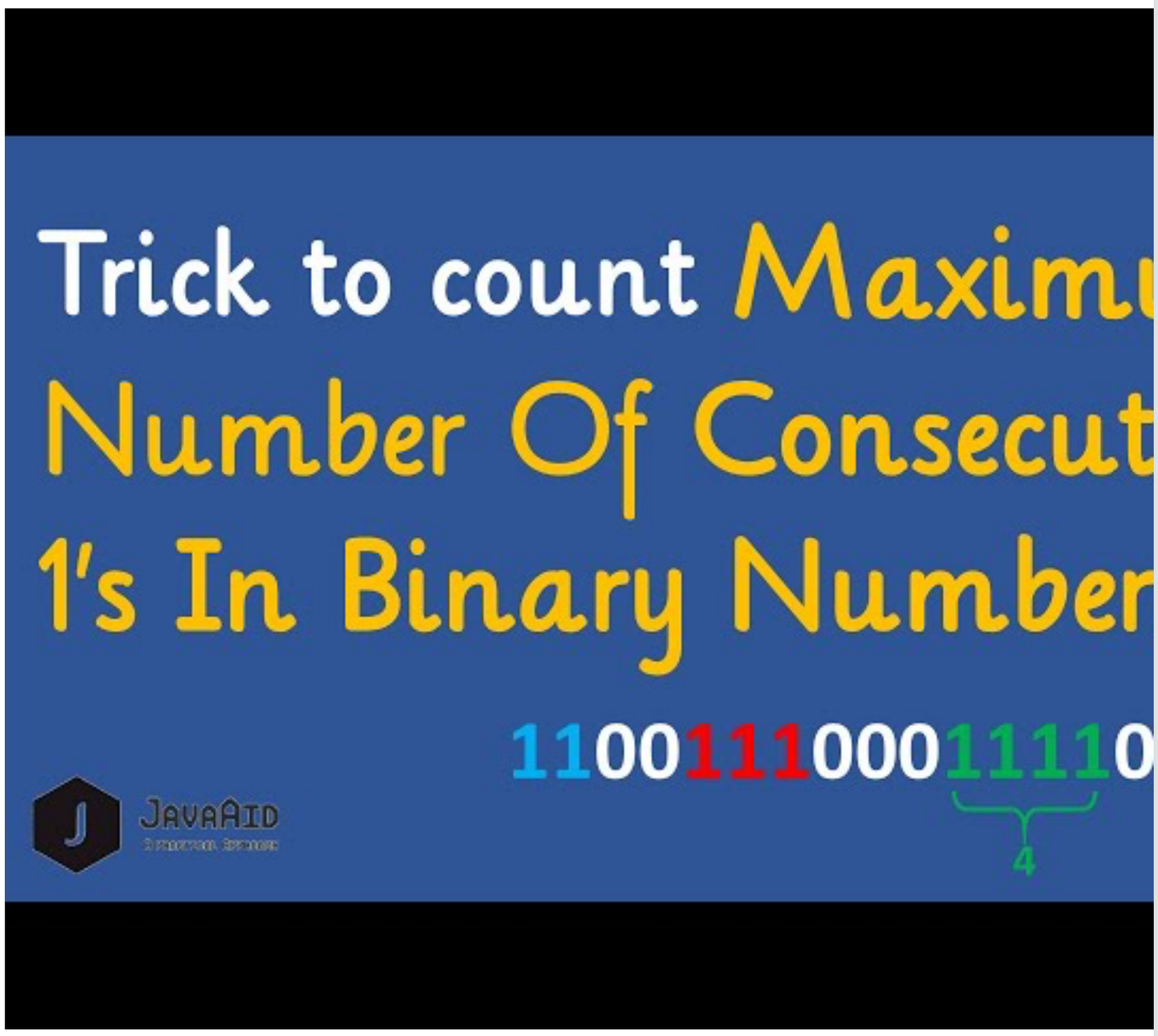
jayu24 3 years ago

explain the logic

0 | Add Comment Parent Permalink

Kanahaiya 3 months ago

Length Of The Longest Consecutive 1s In Binary Representatio Number | BitManipulation



Click here for video explanation

Would really appreciate your feedback like, dislike , comment etc. on my video.

0 | Add Comment Permalink

sahnigwl 2 years ago

```
binary=list(map(int,bin(int(input())).replace('b','')))
print(binary)
result = 0
maximum = 0
for i in binary:
    if i==1:
        result += 1
        if result > maximum:
            maximum = result

    else:
        result = 0
print(maximum)
```

0 | Add Comment Permalink

sahnigwl 2 years ago

python code for the question

0 | Add Comment Parent Permalink

himamshu14 2 years ago

num=13 s=str(bin(num)) c=0 for i in range(len(s)): if s[i]=='1' and str[i]==str[i+1]: c=c+1

print(c)

it is showing error though the logic is correct. can someone pls explain why

0 | Add Comment Permalink

himamshu14 2 years ago

num=13 s=str(bin(num)) print(s) c=0 for i in range(len(s)): if s[i]=='1' and str[i]==str[i+1]

c=c+1

print(c)

it is showing error though the logic is correct. can someone pls explain why

0 | Add Comment Permalink

muskan210497 2 years ago

```
int main(){
    int n, r,c;
    cin >> n;
    while(n!=0){
    r = n%2;
    n = n/2;
      if(r ==1){
          c +=1;
      }
        else if(r ==0){
            c = 0;
        }

    }
    cout<< c;
    return 0;
}
```

can anyone tell me what is the problem in my code. Test case 2 is not satisfying.

0 | Add Comment Permalink

Kanahaiya 3 months ago

Here you are making c to zero if sequenc is something 1000.

Here is the efficient way to solve this problem. https://youtu.be/bc7cxeDy308

0 | Add Comment Parent Permalink

udaykiran_kondr1 2 years ago

```
#!/bin/python3

import sys
import math

n = int(input().strip())
k=int(math.log(n,2))+1
j=0
c=0
l=[]
while(j<k):
    if(n & 1<<j):
        c+=1
    else:
        l.append(c)
        c=0

    j+=1
l.append(c)
print(max(l))
```

0 | Add Comment Permalink

abhishek008 2 years ago

passing all test case except #testcase2.

int main() {

```
int n;

int rem;

int count=0;

scanf("%d",&n);

while(n!=0)

{

rem=n%2;

n=n/2;

if(rem==1)

count=count+1;

else if(rem==0)

count=0;

}

printf("%d",count);

return 0;

}
```

0 | Add Comment Permalink

aakankshasahu09 2 years ago

not showing correct output for 439

0 | Add Comment Parent Permalink

[deleted] 2 years ago

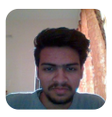
```
int n;
    cin >> n;
    int count=0,pre=0;

    while(n!=0)
        {
         if(n%2)
            count++;
         else
             {
             if(count>pre)
                 pre=count;
                 count=0;
             }
        n/=2;

    }

     if(count>pre)
     pre=count;
     cout<<pre;
```

0 | Add Comment Permalink



abhilashpani651 2 years ago

[deleted]

0 | Add Comment Parent Permalink

Kanahaiya 3 months ago

Here is the efficient way to solve this problem.

https://youtu.be/bc7cxeDy308

0 | Add Comment Parent Permalink

Load more conversations