GeeksforGeeks
A computer science portal for geeks

GeeksQuiz

Login

Home Algorithms DS GATE Interview Corner Q&A C C++ Java Books Contribute Ask a Q About
Array Bit Magic C/C++ Articles GFacts Linked List MCQ Misc Output String Tree Graph

Count numbers that don’t contain 3

Given a number n, write a function that returns count of numbers from 1 to n that don’t contain digit 3 in their decimal representation.

Examples:

```
Input: n = 10
Output: 9

Input: n = 45
Output: 31
// Numbers 3, 13, 23, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 43 contain digit 3.

Input: n = 578
Ouput: 385
```

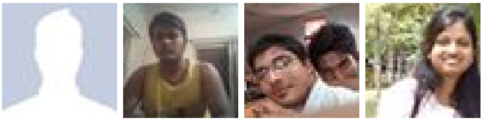
Solution:
We can solve it recursively. Let count(n) be the function that counts such numbers.

```
'msd' --> the most significant digit in n
'd'   --> number of digits in n.

count(n) = n if n < 3

count(n) = n - 1 if 3 <= n < 10

count(n) = count(msd) * count(10^(d-1) - 1) +
```



Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion

```
          count(msd) +
          count(n % (10^(d-1)))
          if n > 10 and msd is not 3

count(n) = count( msd * (10^(d-1)) - 1)
          if n > 10 and msd is 3
```

```
Let us understand the solution with n = 578.
count(578) = 4*count(99) + 4 + count(78)
The middle term 4 is added to include numbers 100, 200, 400 and 500.

Let us take n = 35 as another example.
count(35) = count (3*10 - 1) = count(29)
```

```c
#include <stdio.h>

/* returns count of numbers which are in range from 1 to n and don't c
   as a digit */
int count(int n)
{
    // Base cases (Assuming n is not negative)
    if (n < 3)
        return n;
    if (n >= 3 && n < 10)
       return n-1;

    // Calculate 10^(d-1) (10 raise to the power d-1) where d is
    // number of digits in n. po will be 100 for n = 578
    int po = 1;
    while (n/po > 9)
        po = po*10;

    // find the most significant digit (msd is 5 for 578)
    int msd = n/po;

    if (msd != 3)
      // For 578, total will be 4*count(10^2 - 1) + 4 + count(78)
      return count(msd)*count(po - 1) + count(msd) + count(n%po);
    else
      // For 35, total will be equal to count(29)
      return count(msd*po - 1);
}
```

Geometric Algorithms

Popular Posts

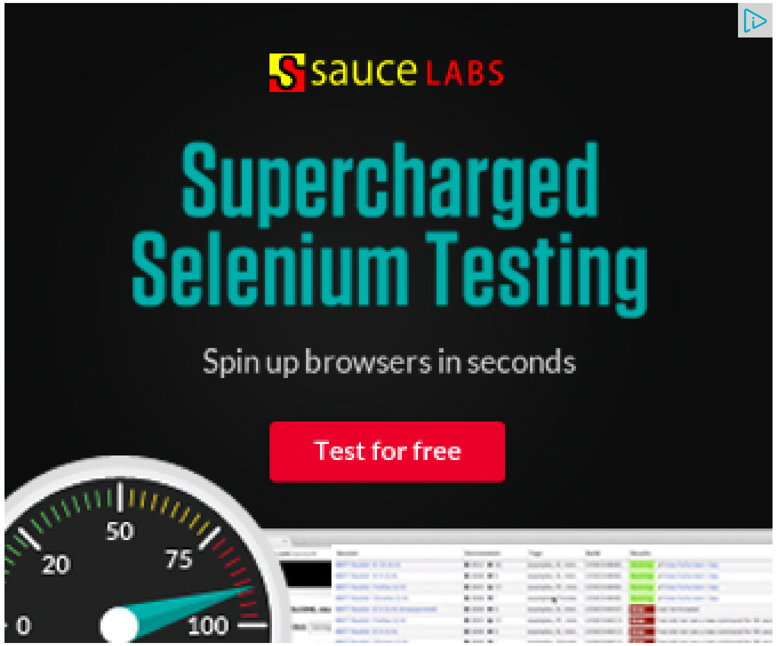
- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

```
// Driver program to test above function
int main()
{
    printf ("%d ", count(578));
    return 0;
}
```

Output:

```
385
```

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above



Related Tpoics:

- Backtracking | Set 8 (Solving Cryptarithmetic Puzzles)
- Tail Recursion
- Find if two rectangles overlap
- Analysis of Algorithm | Set 4 (Solving Recurrences)
- Print all possible paths from top left to bottom right of a mXn matrix

- Generate all unique partitions of an integer
- Russian Peasant Multiplication
- Closest Pair of Points | O(nlogn) Implementation

2 Tweet 0 0

Writing code in comment? Please use **ideone.com** and share the link here.

705 Subscribe

25 Comments GeeksforGeeks

Sort by Newest ▾

Join the discussion…

Hawk Eye • 9 months ago

I think this just an ordinary combinatorics problem....
let&#039s think for 578...
there are three gaps ==> _ _ _.
we can fill this item with any of the valid items.... {0,1,2,4,5,6,7,8,9}=9 items..
as we can repeat any item any number of time... so we can fill the gaps in 9^3
them result.... because , 0 0 0 =0 is not a valid number (counting starts from 1
now for 3rd gap 7,8,9 is not valid … cause they are more than 5(third digit of 5
Again 8,9 is not valid for 2nd gap.. cause they are more than 7(2nd digit of 578
And 9 is not valid for 1st gap.. cause it is more than 8(1st digit of 578).
So the actual result is = (9^3-9^2*3-9^1*2-1)-1=385 ….
Think a bit... you will get the math and also the algo ….

^ | ˅ •

সেুত বসাক ➦ Hawk Eye • 5 months ago

now for 3rd gap 7,8,9 is not valid … cause they are more than 5(third d

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 17 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 57 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago and what will be for number like 36??

raghson • 10 months ago

In the example mentioned above and in the comment of the code, it is mentior

I think it should be 5 * msd(99). However, the code is perfect but there is little r comment.

Rather than
// For 578, total will be 4*count(10^2 - 1) + 4 + count(78)
it should be
// For 578, total will be 5*count(10^2 - 1) + 4(which is msd(5)) + count(78).

Aashish ➦ raghson • 10 months ago

Please take a closer look. Mulplicating factor 4 is fine. We need to excl code count(5) returns 4. So in the calculation, it should be 4 and not 5.

Vignesh Venkatesan • 11 months ago

I think this would be an another optimal solution.

```
#include<iostream>
#include<conio.h>

using namespace std;.
int check(int num).
{
while(num>1)
{
int t=num%10;
return 0;
num=(num-t)/10;
}
return 1;
}
int count(int i).
```

see more

Rishabh · 2 years ago

here is my code nothing special but yes it works fine

```
[sourcecode language="C++"]
#include<iostream>
using namespace std;
int lets_count(int n)
{
int i,N=n,d;
for(i=1;i<=n;i++)
{
int j=i;
while(j>0)
{
d=j%10;
if(d==3)
{
N--;
break;
```

see more

Ankit Gupta · 2 years ago

Answer is : count3(0)-1 // Have included 0 also as a num not containing 3.

```
 int n;
string num;

inline int totdigits(int digit) { return (digit >= 3 ? digit : digit+

int count3(int pos)
{
    int digit = num[pos] - '0';

    if (pos == n-1) {
        return totdigits(digit);
    }
    return (digit != 3) * count3(pos+1) + totdigits(digit-1) * pow(9,
}
```

Ankit Gupta ➦ Ankit Gupta · 2 years ago

```
 This is O(n). And is less complex.
For 574
Possible digits at
one's place
4
ten's place
1*4 + (7-1)*9 = 1*count3(pos+1) + (digit-1)*pow(9, n-pos-1) //
hundred's place
1*1*4 + 1*(7-1)*9 + (5-1)*9*9 = 1*count3(pos+1) + (digit-1)*pow
```

```
and so on for others.
This gives 382, off by 1 from 1 to n.
```

Amit • 2 years ago

@Ashish, Can you please explain how did you conclude that the below foruma source of this ?

```
 count(n) = n if n < 3

count(n) = n - 1 if 3 <= n < 10

count(n) = count(msd) * count(10^(d-1) - 1) +
           count(msd) +
           count(n % (10^(d-1)))
           if n > 10 and msd is not 3

count(n) = count( msd * (10^(d-1)) - 1)
           if n > 10 and msd is 3
```

Aashish ➦ Amit • 2 years ago

I am explaining you with an example.

Suppose we need to count such numbers from 1 to 574.
If we know there are N such numbers from 1 to 99, then the problem c
P*N + foo(74) + Q

Where P=4 as we have to exclude 300 to 399.
Q=4 to take care of 100,200,400 & 500.

Now, the foo() can be called recursively to find count of such numbers

e.g.
n=574

calculate(574);
----

po=100
digit=5
rem=74

see more

Aashish ➦ Aashish • 2 years ago

Don't forget to handle corner cases.
As mentioned in the post, consider n=38.
If 3 is encountered, discard digits right to it.
So, call foo(29).

Aashish ➦ Amit • 2 years ago

I am explaining you with an example.

Suppose we need to count such numbers from 1 to 574.
If we know there are X such numbers from 1 to 99, then the problem c
P*X + foo(74) + Q

Where P=4 as we have to exclude 300 to 399.
Q=4 to take care of 100,200,400 & 500.

Now, the foo() can be called recursively to find count of such numbers
e.g.
n=574

calculate(574);
----

po=100
digit=5
rem=74

see more

Amit ➦ Aashish • 2 years ago

Thanks Aashish .. I understand now. good finding

Abusaleh Nayeem • 2 years ago

I just convert Aditya Pn code into C++

```
 #include <iostream>
#include <math.h>

using namespace std;

int main()
{
    int c = 0;
    int number = 50;
    for(int i=1; i<=number; i++) {
        bool flag = true;
        int num = i;
        for(int j=1; j<=i; j++) {
            int digit = (int) (num%(int)(pow(10, (double)j)));
            if(digit == 3) {
```

see more

googolplex • 2 years ago

```
#include <stdio.h>
#include <math.h>
int main()
{

        int i=0, n, t, sum=0, sub=0;
        scanf("%d", &n);

        while( n )
        {
                t = n%10;

                if( t>3 )
                        sum+=( (t-1)*pow(9, i) );
                else
                {
                        if(t==3){
```

see more

Aashish ➦ googolplex • 2 years ago

Nice approach.

If i am not mistaken, you are making use of the formula:

9^i such numbers exist from 1 to 10^i;

The pow(9,i) takes O(logi) time at best.

Aashish Aashish • 2 years ago

Based on the same approach, here is the simplified code.

```
int count(int num)
{
        int seen=0,i=0,dig,sum=0;
        for(;num;num/=10,++i)
        {
                dig=num%10;
                if(dig>3)
                        sum+=(dig-1)*pow(9,i);
                else
                {
                        sum+=dig*pow(9,i);
                        if(3==dig)
                                sum-=seen;
                }
                seen=seen*10+dig;
        }
        return sum;
}
```

http://ideone.com/uaNRd

googolplex Aashish • 2 years ago

sorry, but i didn't get what is simplified in this code.

rakesh · 2 years ago

```
#include<stdio.h>



int count(int);





int main(){
        int n, i, j=0;
        int c=0;
        scanf("%d", &n);
        for(i=0;i<=n;i++){

                j = j + count(i);
        }
        j = n - j;
        printf("%d\n", j);
```

see more

venkat · 2 years ago

```
#include <stdio.h>
#include <conio.h>

main()
```

> ```
> {
>
>    int  n=0, count=0, i=0 , k=0, r=0;
>
>
>
>     printf("enter the number \n");
>    scanf("%d", &n);
>    if(n<=2){
>
>    printf("Output =  %d \n", n  );
>    return 0;
>
>    }
> ```

see more

venkat • 2 years ago

#include
#include

main()
{

int n=0, count=0, i=0 , k=0, r=0;

printf("enter the number \n");
scanf("%d", &n);
if(n2 && n<=10){
printf("Output = %d \n", n-1);

return 0;
}

count=1;

for(i = 1;i0){

r=k%10;

if(r==3){

see more

Aditya Pn • 2 years ago

```
import java.util.*;

public class Skip3 {

        /**
         * Skip the numbers that have 3 in it
         *
         */

        public static void main(String[] args) {
                 int c = 0;
                 int number = 50;
                 for(int i=1;i<=number;i++){
                         boolean flag = true;
                         int num = i;
                         for(int j=1;j<=i;j++){
```

see more

mayank agarwal • 2 years ago

can anyone tell the time complexity of code written by aashish

duke · 2 years ago

we van do it this way too!!!!!!!!!!!!

```
#include
using namespace std;
void numbercount(int num)
{
int r;
static int count=0; //count number of 3 in a number
if(num<3)
return;
else
{
int temp=num;
while(temp)
{
r=temp%10;
if(r==3)
{
```

see more

lost ➜ duke · 2 months ago

How do I write the code for counting all numbers with 5 from 1 to 10,0C