

[Courses](#)[Login](#)[Suggest an Article](#)

How can one become good at Data structures and Algorithms easily?

Quick Sort vs Merge Sort

Minimum number of items to be delivered

Find maximum in a stack in $O(1)$ time and $O(1)$ extra space

Reach the numbers by making jumps of two given lengths

Cracking Technical Interviews

Samsung Semiconductor Institute of Research(SSIR)

Software) intern/FTE |
Set-3

Print the nodes of
binary tree as they
become the leaf node

Find the number of
distinct islands in a 2D
matrix

Shannon-Fano
Algorithm for Data
Compression

Sum of similarities of
string with all of its
suffixes

Maximize the total
profit of all the persons

Rearrange Odd and
Even values in Alternate
Fashion in Ascending
Order

Smallest power of 2
which is greater than or
equal to sum of array
elements

Dijkstra's shortest path
with minimum edges

Sort the array of strings
according to
alphabetical order
defined by another
string

Program for SSTF disk
scheduling algorithm

Check if it is possible to
reach a number by
making jumps of two
given length

Code Optimization
Technique (logical AND
and logical OR)

Program to print the
Zigzag pattern

Number of array
elements derivable
from D after performing
certain operations

Largest perfect square
number in an Array

Dividing a Large file into
Separate Modules in
C/C++, Java and
Python

Minimum cost to reach
the top of the floor by
climbing stairs

Lower and Upper Bound
Theory

Smallest Pair Sum in an
array

Find elements of array
using XOR of
consecutive elements

Maximize the value of A
by replacing some of its
digits with digits of B

Count distinct
substrings that contain
some characters at
most k times

Cristian's Algorithm

Ad ▾

experienceEducation
that skyrockets your potential





Count of Numbers in Range where first digit is equal to last digit of the number

Given a range represented by two positive integers L and R. Find the count of numbers in the range where the first digit is equal to the last digit of the number.

Examples:

Input : L = 2, R = 60

Output : 13

Explanation : Required numbers are

2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44 and 55

Input : L = 1, R = 1000

Output : 108

Recommended: Please try your approach on *{IDE}* first, before moving on to the solution.



Free Writing Tool

Improve grammar, word choice, and sentence structure writing. It's free!

Prerequisites : Digit DP

There can be two approaches to solve this type of problem, one can be a combinatorial solution and other can be a dynamic programming based solution. Below is a detailed approach of solving this problem using a digit dynamic programming.

Dynamic Programming Solution : Firstly, if we are able to count the required numbers upto R i.e. in the range $[0, R]$, we can easily reach our answer in the range $[L, R]$ by solving for from zero to R and then subtracting the answer we get after solving for from zero to $L - 1$. Now, we need to define the DP states.

DP States:

- Since we can consider our number as a sequence of digits, one state is the **position** at which we are currently in. This position can have values from 0 to 18 if we are dealing with the numbers upto 10^{18} . In each recursive call, we try to build the sequence from left to right by placing a digit from 0 to 9.
- Second state is the **firstD** which defines the first digit of the number we are trying to build and can have values from 0 to 9.
- Third state is the **lastD** which defines the last digit of the number we are trying to build and can have values from 0 to 9.
- Another state is the boolean variable **tight** which tells the number we are trying to build has already become smaller than R so that in the upcoming recursive calls we can place any digit from 0 to 9. If the number has not become smaller, maximum limit of digit we can place is digit at current position in R.

In each recursive call, we set last digit as the digit we placed in the last position and we set first digit as the first non zero digit of the number. In the final recursive call, when we are at the last position if the first digit is equal to the last digit, return 1, otherwise 0.

Below is the implementation of the above approach.



```
// CPP Program to find the count of  
// numbers in a range where the number  
// does not contain more than K non  
// zero digits
```



```
#include <bits/stdc++.h>

using namespace std;

const int M = 20;

// states - position, first digit,
// last digit, tight
int dp[M][M][M][2];

// This function returns the count of
// required numbers from 0 to num
int count(int pos, int firstD, int lastD,
          int tight, vector<int> num)
{
    // Last position
    if (pos == num.size()) {
        // If first digit is equal to
        // last digit
        if (firstD == lastD)
            return 1;
        return 0;
    }

    // If this result is already computed
    // simply return it
    if (dp[pos][firstD][lastD][tight] != -1)
        return dp[pos][firstD][lastD][tight];

    int ans = 0;

    // Maximum limit upto which we can place
    // digit. If tight is 1, means number has
    // already become smaller so we can place
    // any digit, otherwise num[pos]
    int limit = (tight ? 9 : num[pos]);

    for (int dig = 0; dig <= limit; dig++) {
        int currFirst = firstD;
```



```

        // If the position is 0, current
        // digit can be first digit
        if (pos == 0)
            currFirst = dig;

        // In current call, if the first
        // digit is zero and current digit
        // is nonzero, update currFirst
        if (!currFirst && dig)
            currFirst = dig;

        int currTight = tight;

        // At this position, number becomes
        // smaller
        if (dig < num[pos])
            currTight = 1;

        // Next recursive call, set last
        // digit as dig
        ans += count(pos + 1, currFirst,
                    dig, currTight, num);
    }
    return dp[pos][firstD][lastD][tight] = ans;
}

// This function converts a number into its
// digit vector and uses above function to compute
// the answer
int solve(int x)
{
    vector<int> num;
    while (x) {
        num.push_back(x % 10);
        x /= 10;
    }
    reverse(num.begin(), num.end());

    // Initialize dp

```

```

    memset(dp, -1, sizeof(dp));
    return count(0, 0, 0, 0, num);
}

// Driver Code
int main()
{
    int L = 2, R = 60;
    cout << solve(R) - solve(L - 1) << endl;

    L = 1, R = 1000;
    cout << solve(R) - solve(L - 1) << endl;

    return 0;
}

```

Output:

13
108

Time Complexity : $O(18 * 10 * 10 * 2 * 10)$, if we are dealing with the numbers upto 10^{18}



Recommended Posts:

[Count numbers with unit digit k in given range](#)

[Count numbers \(smaller than or equal to N\) with given digit sum](#)

[Count of Numbers in a Range where digit d occurs exactly K times](#)

[Count of Numbers in a Range divisible by m and having digit d in even positions](#)

[Count n digit numbers not having a particular digit](#)

Count n digit numbers divisible by given number

Count total number of N digit numbers such that the difference between sum of even and odd digits is 1

Count numbers with difference between number and its digit sum greater than specific value

Count 'd' digit positive integers with 0 as a digit

Find the highest occurring digit in prime numbers in a range

Count numbers from 1 to n that have 4 as a digit

Count numbers having 0 as a digit

Count numbers having 0 as a digit

Count all possible N digit numbers that satisfy the given condition

Count of Binary Digit numbers smaller than N



Nishant Tanwar

Check out this Author's [contributed articles](#).

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.



Article Tags : Algorithms Combinatorial Competitive Programming Dynamic Programming Mathematical digit-DP number-digits Technical Scriptor 2018

Practice Tags : Dynamic Programming Mathematical Combinatorial Algorithms



11

☐ To-do ☐ Done

3

Based on 9 vote(s)

Feedback

Add Notes

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Share this post!

A computer science portal for geeks

710-B, Advant Navis Business Park,
Sector-142, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE

Company-wise
Topic-wise
Contests
Subjective Questions

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks, Some rights reserved