



# Sam and substrings ☆

Problem Submissions Leaderboard Discussions Editorial



Editorial by dheeraj

A brute force approach to solve this problem is to generate all possible sub-strings of the given number. In the worst case scenario, for generating all possible sub-strings for  $N$  of size  $2 * 10^5$ , it would take

$$\text{len}(N) * (\text{len}(N) + 1) / 2$$

which is about  $10^{10}$  operations.

There is an efficient dynamic programming approach which can solve this problem in linear time with respect to the given number.

Let's look at the algorithm for  $N = 5312$ ,

all possible substrings are

5 3 1 2 53 31 12 531 312 5312

with a little bit of rearrangement,

5 | 3 53 | 1 31 531 | 2 12 312 5312

The above rearrangement lists all numbers that ends with 5, 3, 1 and 2 respectively.

Let  $sd[i]$  hold the sum of all digits that ends at  $N[i]$  i.e.,

$sd[0]$  holds the sum of all digits that ends at  $N[0] = 5$

$sd[1]$  holds the sum of all digits that ends at  $N[1] = 3$

$sd[2]$  holds the sum of all digits that ends at  $N[2] = 1$

$sd[3]$  holds the sum of all digits that ends at  $N[3] = 2$

If  $S$  is the final answer, then we have

$$S = sd[0] + sd[1] + sd[2] + sd[3]$$

ie.,  $S = \sum sd[i] \forall 0 \leq i < \text{len}(N)$

Now, given  $sd[i]$ , let us see if we can calculate  $sd[i+1]$ .

$$sd[2] = 1 + 31 + 531 = 563$$

$$sd[3] = 2 + 12 + 312 + 5312$$

$$sd[3] = 2 + 10 + 2 + 310 + 2 + 5310 + 2$$

$$sd[3] = 4 * 2 + 10 * (1 + 31 + 531)$$

$$sd[3] = (3 + 1) * N[3] + 10 * sd[2]$$

on generalising, we get

$$sd[i+1] = (i + 2) * N[i] + 10 * sd[i]$$

$$sd[0] = N[0]$$

As  $S$  can be large, we can use [modulo equivalence](#) at every step ( Why? ) and print  $S \% 10^9 + 7$ . This process takes  $O(\text{len}(N))$  where  $\text{len}(N)$  is the size of the input.



Set by amititkgp

Problem Setter's code:

## STATISTICS

Difficulty: **Medium**

Success Rate: 52.94%

Time Complexity:  $O(N)$

Required Knowledge: Dynamic Programming

Publish Date: May 16 2014

Originally featured in [Weekly Challenges - Week 3](#)

Of the 1475 contest participants, 595 (40.34%) submitted code for this challenge.

## NEED HELP?

[View discussions](#)

[View top submissions](#)

**C++**

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

using namespace std;
typedef unsigned long long LL;

#define MOD7 1000000007

string str;

int main()
{
    cin>>str;
    int l = str.length();
    LL outp = str[0] - '0'; // output stored in it.
    LL temp = str[0] - '0'; // sum of all strings ending at index i is stored in this string
    for(int i=1 ; i< l ; i++)
    {
        temp = temp*10 + (i+1)*(str[i] - '0');
        temp = temp%MOD7;
        outp = (outp +temp)%MOD7;
    }
    cout<<outp<<endl;
}
return 0;
```



Tested by shashank21j

Problem Tester's code:

**Python 2**

```
s=raw_input()
len1=len(s)
mul_ten=1
sum1=1
total_sum=len1*int(s[len1-1])
for i in range(len1-2,-1,-1):
    mul_ten=(mul_ten*10)%1000000007
    sum1=sum1+mul_ten
    total_sum=(total_sum+(sum1*(i+1))*int(s[i]))%1000000007
print total_sum%1000000007
```

**Feedback**

Was this editorial helpful?

Yes

No

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)