

DIGIT DP



Tejas Khairnar

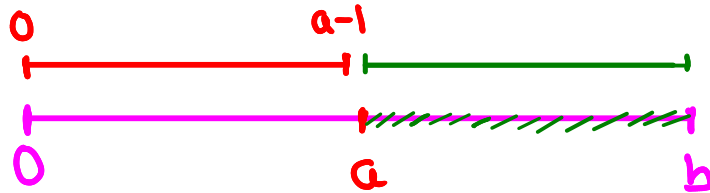
CSE 3rd Year

ICPC Executive, CSEA

IIT Guwahati

Where it is used??

- There are many problems that ask to **count the number of integers x** between two integers say **a and b** such that x satisfies a specific property that can be related to its digits.
- $G(x)$ tells the number of such integers between 1 to x (inclusively), then the number of such integers between a and b can be given by **$G(b) - G(a-1)$** .



Digit DP

- Convert the given number to a string.

$s_0 s_1 \dots s_{n-1}$

as the decimal representation where s_i tells the i -th digit from the left.

- Now, after representing the given number this way we generate the numbers **less than or equal to the given number** and simultaneously calculate using DP, if the number satisfies the given property.
- We start generating integers having number of digits = 1 and then till number of digits = n .
- Integers having **number of digits less than n** can be analyzed by setting the leftmost digits to be zero.
- Declare the string globally to avoid creation of unnecessary copies of the string.

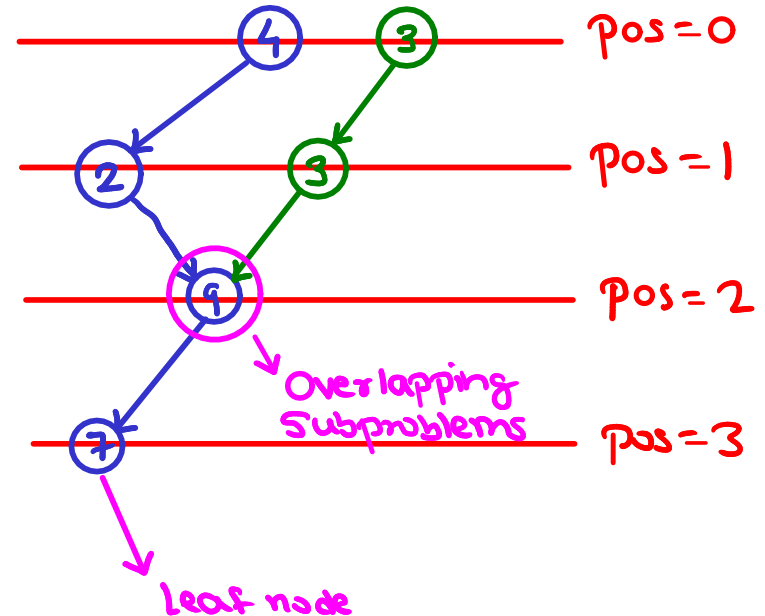
Digit Dp Explained

- Tight = 1 -----> restricted range
Tight = 0 -----> unrestricted range

➤ Consider the number : **5** 8 3 1

↓

4 - - -



Basic Code Structure

- **N:** upper limit in number of digits in number.
- **M:** upper limit of Max value of function $G(x)$
- **2:** tight specifies whether we are free to choose any number as the current digit.

```
string s;  
int dp[N][M][2];  
  
int solve(int pos, int cnt, int t)  
{  
    if (pos == int(s.size()))  
    {  
        // return value  
    }  
  
    if (dp[pos][cnt][t] != -1)  
        return dp[pos][cnt][t];  
  
    int lim = t ? s[pos] - '0' : 9;  
    int ans = 0;  
  
    for (int i = 0; i <= lim; i++)  
    {  
        ans += solve(pos + 1, cnt + (*condition*), t & (i == lim));  
    }  
  
    return dp[pos][cnt][t] = ans;  
}
```

Question 1

Problem Statement

Find the number of integers between 1 and K (inclusive) satisfying the following condition, modulo $10^9 + 7$:

- The sum of the digits in base ten is a multiple of D .

Constraints

- All values in input are integers.
- $1 \leq K < 10^{10000}$
- $1 \leq D \leq 100$

Sample Input 1

[Copy](#)

```
30
4
```

[Copy](#)

Sample Output 1

[Copy](#)

```
6
```

[Copy](#)

Those six integers are: 4, 8, 13, 17, 22 and 26.

Question 1

```
const int N = 1e4 + 5;
string s;
lli dp[N][105][2];
int d;
lli solve(int pos, int cnt, int t)
{
    if (pos == int(s.size()))
    {
        return cnt == 0;
    }

    if (dp[pos][cnt][t] != -1)
        return dp[pos][cnt][t];

    int lim = t ? s[pos] - '0' : 9;
    lli ans=0;

    for(int i=0;i<=lim;i++)
    {
        ans+=solve(pos+1,(cnt+i)%d,t&(i==lim));
        ans%=mod;
    }

    return dp[pos][cnt][t]=ans;
}

int main()
{
    kira;
    memset(dp, -1, sizeof(dp));
    cin>>s>>d;
    cout<<(solve(0,0,1)-1+mod)%mod;
    return 0;
}
```

Question 2

A number is said to be a 369 number if

1. The count of 3s is equal to count of 6s and the count of 6s is equal to count of 9s.
2. The count of 3s is at least 1.

For Example 12369, 383676989, 396 all are 369 numbers whereas 213, 342143, 111 are not.

Given A and B find how many 369 numbers are there in the interval [A, B]. Print the answer modulo 1000000007.

Input

The first line contains the number of test cases (T) followed by T lines each containing 2 integers A and B.

Output

For each test case output the number of 369 numbers between A and B inclusive.

Constraints

$$T \leq 100$$

$$1 \leq A \leq B \leq 10^{50}$$

Question 2

```
string s;
lli dp[51][18][18][18][2];
lli solve(int pos, int s3, int s6, int s9, int t)
{
    if (s3 >= 17 || s6 >= 17 || s9 >= 17)
        return 0;
    int k = int(s.size());
    if (pos == k)
    {
        if (s3 == s6 && s6 == s9 && s3 >= 1)
            return 1;
        return 0;
    }

    if (dp[pos][s3][s6][s9][t] != -1)
        return dp[pos][s3][s6][s9][t];

    int lim = t ? s[pos] - '0' : 9;
    int nt = t;
    lli ans = 0;
    for (int i = 0; i <= lim; i++)
    {
        if (i != lim)
            nt = 0;
        else
            nt = t;
        ans = (ans + solve(pos + 1, s3+(i==3), s6+(i==6), s9+(i==9), nt) % mod) % mod;
    }
    return dp[pos][s3][s6][s9][t] = ans;
}
```

```
int main()
{
    kira;
    int nc;
    string a, b;
    cin >> nc;
    while (nc--)
    {
        cin >> a >> b;
        int i = int(a.size()) - 1;
        while (a[i] == '0')
        {
            a[i] = '9';
            i--;
        }
        a[i]--;
        memset(dp, -1, sizeof(dp));
        s = a;
        lli y = solve(0, 0, 0, 0, 1);
        memset(dp, -1, sizeof(dp));
        s = b;
        lli x = solve(0, 0, 0, 0, 1);

        cout << (x - y + mod) % mod << endl;
    }
    return 0;
}
```

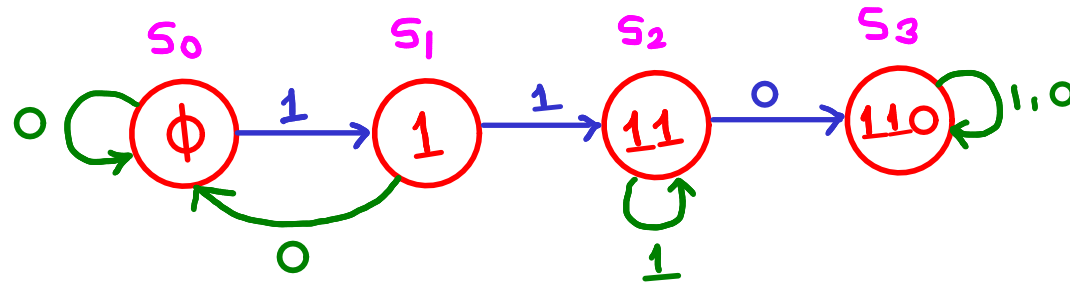
Question 3

➤ Find number of numbers between L to R wherein the binary representation of the number has “110” as its substring.

➤ Constraints :

$$1 \leq L \leq R \leq 10^{18}$$

➤ Key Idea:



Input	0	1
S ₀	S ₀	S ₁
S ₁	S ₀	S ₂
S ₂	S ₃	S ₂
S ₃	S ₃	S ₃

* Transition Table

Question 3

```

string s;
lli dp[65][4][2];
lli solve(int pos, int st, int t)
{
    if (pos == int(s.si))
        return (st == 3);

    if (dp[pos][st][t] != -1)
        return dp[pos][st][t];

    int lim = t ? s[pos] - '0' : 1;
    int nt = t;

    lli ans = 0;
    for (int i = 0; i <= lim; i++)
    {
        if (i != lim)
            nt = 0;
        else
            nt = t;
        int nst;
        if (i == 0)
        {
            if (st == 0)nst = 0;
            else if (st == 1)nst = 0;
            else if (st == 2)nst = 3;
            else if (st == 3)nst = 3;
        }
        else
        {
            if (st == 0)nst = 1;
            else if (st == 1)nst = 2;
            else if (st == 2)nst = 2;
            else if (st == 3)nst = 3;
        }
        ans += solve(pos+1,nst,nt);
    }
    return dp[pos][st][t]=ans;
}

```

Input	0	1	
S ₀	S ₀	S ₁	
S ₁	S ₀	S ₂	
S ₂	S ₃	S ₂	
S ₃	S ₃	S ₃	

```

string tobin(lli n)
{
    string temp="";
    for(lli i=61;i>=0;i--)
    {
        if((1ll<<i)&n)temp+='1';
        else temp+='0';
    }
    return temp;
}

int main()
{
    kira;
    int nc;
    lli l,r;
    cin>>nc;
    while(nc-->0)
    {
        cin>>l>>r;
        l--;
        memset(dp,-1,sizeof(dp));
        s=tobin(l);
        lli y=solve(0,0,1);
        memset(dp,-1,sizeof(dp));
        s=tobin(r);
        lli x=solve(0,0,1);
        cout<<x-y<<endl;
    }
    return 0;
}

```

Question 4

Problem Statement

You are given a positive integer L in base two. How many pairs of non-negative integers (a, b) satisfy the following conditions?

- $a + b \leq L$
- $a + b = a \text{ XOR } b$

Since there can be extremely many such pairs, print the count modulo $10^9 + 7$.

► What is XOR?

Constraints

- L is given in base two, without leading zeros.
- $1 \leq L < 2^{100\,001}$

Sample Input 1

Copy

10

Copy

Sample Output 1

Copy

5

Copy

Five pairs (a, b) satisfy the conditions: $(0, 0)$, $(0, 1)$, $(1, 0)$, $(0, 2)$ and $(2, 0)$.

Question 4

```
lli n, k;
string s;
const int LEN = 1e5 + 5;
lli dp[LEN][2];

lli solve(int pos, int tight)
{
    if (pos == int(s.size()))
    {
        return 1;
    }

    if (dp[pos][tight] != -1)
        return dp[pos][tight];

    int lim = tight ? s[pos] - '0' : 1;

    lli ans = 0;

    for (int i = 0; i <= lim; i++)
    {
        ans += (1 + (i == 1)) * solve(pos + 1, (i == lim) & tight);
        ans %= mod2;
    }

    return dp[pos][tight] = ans;
}
```

Let $a+b=k$
Hence $k \leq L$, apply digit dp on digits of k

$a+b = a^b + 2(a \& b)$
Hence $a \& b = 0$

Question 5

Find the number from the range $[a, b]$ which has the maximum product of the digits.

Input

The first line contains two positive integers a and b ($1 \leq a \leq b \leq 10^{18}$): the left and the right ends of the range.

Output

Print the number with the maximum product of the digits from the range $[a, b]$. If there are several possible answers, print any one of them.

Examples

input

Copy

1 10

output

Copy

9

input

Copy

51 62

output

Copy

59

Question 5

```
string s, t;
pair<lli, string> dp[20][2][2][2];
pair<lli, string> solve(int pos, int t1, int t2, int srt)
{
```

```
    if (pos == int(s.si))
        return {1, ""};
```

```
    if (dp[pos][t1][t2][srt].F != -1)
        return dp[pos][t1][t2][srt];
```

```
    lli st = t1 ? t[pos] - '0' : 0;
    lli et = t2 ? s[pos] - '0' : 9;
```

```
    lli res = -1;
    string ans = "";
```

```
    for (lli i = st; i <= et; i++){
        lli val;
        if (srt == 0 && i == 0) val = 1;
        else val = i;
```

```
        pair<lli, string> k;
        k=solve(pos + 1, t1 & (i == st), t2 & (i == et), srt | (i > 0));
```

```
        if ((val * k.F) > res){
            res = (k.F * val);
            if (srt == 0 && i == 0)
                ans = k.S;
            else{
                reverse(all(k.S));
                k.S.pb('0' + i);
                reverse(all(k.S));
                ans = k.S;
            }
        }
    }
```

```
    return dp[pos][t1][t2][srt] = {res, ans};
}
```

```
int main()
```

```
{
```

```
    kira;
    string lf, rt;
    cin >> lf >> rt;
```

```
    forz(i, 20)
```

```
        forz(j, 2)
```

```
            forz(k, 2)
```

```
                forz(l, 2)
```

```
                    dp[i][j][k][l] = {-1, ""};
```

```
    reverse(all(lf));
```

```
    while (int(lf.si) < int(rt.si))
```

```
    {
```

```
        lf.pb('0');
```

```
    }
```

```
    reverse(all(lf));
```

```
    t = lf;
```

```
    s = rt;
```

```
    string ret = solve(0, 1, 1, 0).S;
```

```
    cout << ret;
```

```
    return 0;
```

```
}
```

Links to the Questions

Sr. No.	Question
Question 1	https://atcoder.jp/contests/dp/tasks/dp_s
Question 2	https://www.spoj.com/problems/NUMTSN/
Question 3	N.A.
Question 4	https://atcoder.jp/contests/abc129/tasks/abc129_e
Question 5	https://codeforces.com/problemset/gymProblem/100886/G

Practice Questions

Difficulty	Question
Easy	https://codeforces.com/problemset/problem/1036/C
Easy	https://www.spoj.com/problems/GONE/
Easy-Medium	https://www.spoj.com/problems/RAONE/
Medium	https://www.spoj.com/problems/LUCIFER/
Medium	https://atcoder.jp/contests/abc135/tasks/abc135_d
Medium	https://codeforces.com/problemset/problem/628/D
Medium-Hard	https://www.codechef.com/problems/AOPN
Hard	https://www.codechef.com/AUG19A/problems/ENCODING
Hard	https://www.codechef.com/problems/MPROD MUL

Acknowledgements

SPECIAL THANKS TO

