Codeforces 296 B

Combinatorics and Reversing Problem

Description

Given two strings with only digits '0'-'9' and '?', define two strings are incomparable if they satisfy the condition:

$$\text{Exist an } (i, j), \text{that } s_i > w_i \wedge s_j < w_j.$$

'?' means that you can choose an arbitrary digit from '0' to '9' at this position.

Calculate how many ways you can fill these strings to make them imcomparable.

Solution

At first I've thought about DP with status definition $f[i][0/1]$ but we will see it cannot work well.

Time past, we would find it's hard to solve the problem, so we may try to reverse it.

We know that all possible ways is 10^x, where x is the number of '?'s.

So just work on how to make $S \leq W$ or $S \geq W$.

Take $S \leq W$ for example, another situation is the same. At first the counter is 1.

$$\begin{cases} counter \leftarrow counter * 55 & \text{if } s_i ='?' \wedge w_i ='?', \\ counter \leftarrow counter * (w_i + 1) & \text{if } s_i ='?' \wedge w_i \neq '?' \\ counter \leftarrow counter * (10 - s_i) & \text{if } s_i \neq '?' \wedge w_i ='?' \end{cases}$$

Additionally, if there exist a position that $s_i > w_i$, we can set the counter to 0 and break the algorithm.

$w_i + 1$ means we can choose any digit from 0 to w_i, and $10 - s_i$ means s_i to 9.

Now we can subtract them from 10^x.

Notice that when all elements are equal, they will be counted in both case $\leq$ and case $\geq$, so we need to count ways to make the string equal, then add to the final answer.

Code

```cpp
#include <iostream>
#include <cstring>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cstdio>
using namespace std;
typedef long long ll;
const int maxn = 1e+5 + 5;
const ll mod = 1e+9 + 7;
int n;
ll tot, sle, wle, equ;
char s[maxn], w[maxn];
inline void solve1() {
  for (int i = 1; i <= n; ++i) {
    if (s[i] == '?' && w[i] == '?') sle = sle * 55 % mod;
    else if (s[i] == '?' && w[i] != '?') sle = sle * (w[i] - '0' + 1) % mod;
    else if (s[i] != '?' && w[i] == '?') sle = sle * (10 - (s[i] - '0')) % mod;
    else if (s[i] - '0' > w[i] - '0') {
      sle = 0; break;
    }
  }
}
inline void solve2() {
  for (int i = 1; i <= n; ++i) {
    if (s[i] == '?' && w[i] == '?') wle = wle * 55 % mod;
    else if (s[i] == '?' && w[i] != '?') wle = wle * (10 - (w[i] - '0')) % mod;
    else if (s[i] != '?' && w[i] == '?') wle = wle * (s[i] - '0' + 1) % mod;
    else if (s[i] - '0' < w[i] - '0') {
      wle = 0; break;
    }
  }
}
inline void solve3() {
  for (int i = 1; i <= n; ++i) {
    if (s[i] == '?' && w[i] == '?') equ = equ * 10 % mod;
    else if (s[i] != '?' && w[i] == '?') continue;
    else if (s[i] == '?' && w[i] != '?') continue;
    else if (s[i] - '0' != w[i] - '0')  {
      equ = 0; break;
    }
  }
}
int main() {
  scanf("%d", &n);
  scanf("%s", s + 1); scanf("%s", w + 1);
  tot = 1;
  for (int i = 1; i <= n; ++i) {
    if (s[i] == '?') tot = tot * 10ll % mod;
    if (w[i] == '?') tot = tot * 10ll % mod;
  }
  sle = 1; wle = 1; equ = 1;
  solve1(); solve2(); solve3();
  cout << (tot - sle - wle + equ + mod) % mod << endl;
  return 0;
}
```