知识精炼（二）

主讲人：邓哲也

HDU 4507

如果一个整数符合下面3个条件之一，那么我们说这个整数和7有关：

整数中某一位是7

整数的每一位加起来的和是7的整数倍

这个整数是7的整数倍

询问[L,R]中与7无关的数字的平方和，模10^9+7.

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

HDU 4507

我们可以先算出与 7 有关的数的平方和，再用总和去减。

有三种情况：

1. 整数中某一位是7 =》 一维状态 a 记录是否有 7 出现
2. 整数的每一位加起来的和是7的整数倍 =〉 一维状态 b 记录当前的数每一位加起来的和模 7 的值
3. 这个整数是7的整数倍 =》一维状态 c 记录当前这个数模 7 的值

状态即为 f[dep][a][b][c]

HDU 4507

但是这里要求的是平方和。

为了转移需要维护三个值：

cnt　和 7 有关的数的个数

sum　和 7 有关的数的和

sqr　和 7 有关的数的平方和

HDU 4507

设当前是第 dep 位，填了数字 k。

dfs 回溯上来的数为 x。

令 $d = k * 10^{dep-1}$ ，那么这个数就是 $d + x$

它的平方对答案的贡献是 $(d+x)^2 = d^2+2dx+x^2$

因此对于回溯上来的数，你得知道与7有关的数的个数（对应d^2），

与7有关的数之和（对应$2dx$），与7有关的数的平方和（对应x^2）

HDU 4507

对答案的贡献是 $(d+x)^2 = d^2+2dx+x^2$

假设当前状态是 f，从 g 状态转移过来（dfs回溯上来）

f.cnt += g.cnt

f.sum += g.sum + d * g.cnt

f.sqr += d * d * g.cnt + 2 * d * g.sum + g.sqr

计算1～n的平方和：n * (n + 1) * (2n + 1) / 6

HDU 4507

```
long long p10[21]; // 10 的幂次
struct node{
      long long cnt, sum, sqr;
}f[20][2][7][7];
```

HDU 4507

```
node dp(int eq, int dep, int a, int b, int c){
    if(!eq && ~f[dep][a][b][c].cnt) return f[dep][a][b][c];
    if(!dep){
        int k=!(!a && b && c);
        f[dep][a][b][c]=(node){k,0,0};
        return f[dep][a][b][c];
    }
    int ed=(eq)?A[dep]:9;
    node t=(node){0,0,0};
    …
```

HDU 4507

```
…
for(int i=0;i<=ed;i++){
        int f=1ll*p10[dep-1]*i%mo;
        node p=dp(eq&&(i==ed),dep-1,a||(i==7),(b+i)%7,(c*10+i)%7);
        t.cnt=(t.cnt+p.cnt)%mo;
        t.sum=(t.sum+p.sum+1ll*p.cnt*f%mo)%mo;

t.sqr=(t.sqr+p.sqr+1ll*f*f%mo*p.cnt+2ll*f*p.sum%mo)%mo;
}
if(!eq) f[dep][a][b][c]=t;
return t;
}
```

下节课再见