

# 知识精炼（三）



主讲人：邓哲也



# Codeforces 55D Beautiful Numbers

定义一个数是美丽的，当且仅当它能整除自身的每个非零数位。

例如250是美丽的，2333就是不美丽的。

询问  $[L, R]$  中有多少个数是美丽的。

$$1 \leq L \leq R \leq 9 \cdot 10^{18}$$

## Codeforces 55D Beautiful Numbers

考虑一个数能被它的每个非零数位整除。

等价于一个数能被它的每个非零数位的LCM整除。

而1~9的LCM是 2520。

因此一个数的所有非零数位的LCM一定是 2520的因子。

## Codeforces 55D Beautiful Numbers

我们设状态为  $f[dep][lcm][mo]$

表示当前枚举到了第  $dep$  位，当前所有非零数位的 LCM 是  $lcm$ ，当前这个数模 2520 的值是  $mo$ 。

因为中途  $lcm$  会变化，所以不能用  $mo$  表示模  $lcm$  的值。

最后检查  $mo \% lcm == 0$  即可。

枚举当前位  $k$ ，哪些状态可以转移到当前状态呢？

$f[dep-1][LCM(lcm, k)][(mo*10+k) \% 2520]$

## Codeforces 55D Beautiful Numbers

但是  $lcm$  和  $mo$  都是 2520 的大小，内存不够

注意到 2520 的因子只有 48 个。

因此把  $lcm$  这一维优化到 48 就行了。

数组的大小就变成了  $20 * 48 * 2520$

# Codeforces 55D Beautiful Numbers

```
long long dp(int less, int dep, int l, int mo) {  
    if(!dep) return (mo%b[1]==0);  
    if(!less) if(~f[dep][1][mo]) return f[dep][1][mo];  
    if(less) if(~g[dep][1][mo]) return g[dep][1][mo];  
    long long ret=0;  
    int ed=(less)?9:a[dep];  
    for(int i=0;i<=ed;i++) ret+=dp(less||(i<ed), dep-1, (!i)?1:d[1][i], (mo*10+i)%2520);  
    if(!less) f[dep][1][mo]=ret; else g[dep][1][mo]=ret;  
    return ret;  
}
```

# Codeforces 55D Beautiful Numbers

```
long long work(long long x) {  
    a[0]=0;  
    memset(f, -1, sizeof(f));  
    memset(g, -1, sizeof(g));  
    for (; x; x/=10) a[++a[0]] = x%10;  
    return dp(0, a[0], 0, 0);  
}  
  
void solve() {  
    long long L, R;  
    cin >> L >> R;  
    cout << work(R) - work(L-1) << endl;  
}
```

# Codeforces 55D Beautiful Numbers

```
int main() {  
    int t=0;  
    for(int i=1;i<=2520;i++)    if(2520%i==0) {    b[t]=i; c[i]=t++;}  
    for(int i=0;i<t;i++)    for(int j=1;j<10;j++)    d[i][j]=c[lcm(b[i], j)];  
    int tc;  
    scanf("%d",&tc);  
    while(tc--) solve();  
    return 0;  
}
```



下节课再见