数位DP通用解法

主讲人：邓哲也

数位DP

问题的一般形式是这样的：

定义一个条件 A，比如：被 7 整除、数位中含有 3 等等。

询问区间 [L, R] 中有几个数满足条件 A

L 和 R 的范围一般非常大，比如10^{18}

通过数位 DP，我们会发现这些问题的规模实际上是 $\log_{10} R$

数位DP

数位 DP 就是考虑数字的每一位。

问题的规模变为 $\log_{10} N$

每一位作为不同的阶段，设计状态。

我们从高位往低位依次枚举。

（为什么不选择从低位到高位？）

每一位的数选择的范围是不同的，依据前面选的数决定。

数位DP

比如 N 是 1230.

假设前两位枚举的数是 1 和 2，也就是计算了12开头的贡献。

此时枚举第三位，可选的范围只有0～3

如果前两位枚举的数是 1 和 0

此时枚举第三位，可选的范围就是0～9

因此我们用一个变量 eq 或者 less，表示在枚举当前位之前，

每一位是不是都和 N 选的一样。

数位DP

当前设为第 dep 位，N的第dep位为 A[dep]，假设填上 k

如果采用 eq 变量：

```
eq = eq && (A[dep] == k)
```

可选的最大值是 eq ? A[dep] : 9

如果采用 less 变量：

```
less = less || (k < A[dep])
```

可选的最大值是 less ? 9 : A[dep]

数位DP

为了直观的理解数位 DP，我们介绍记忆化搜索的形式。

高位的答案由低位的答案转移而来。

下面一起来看一道例题。

BZOJ 1799 同类分布

询问 [L,R] 中各位数字之和能整除原数的个数。

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

BZOJ 1799 同类分布

可以发现各位数之和最大只能是 9 * 18 = 162

我们可以枚举这个和 sum

然后去统计可以被 sum 整除，且数位和是 sum 的数。

我们把状态定义为 f[dep][cur][mod]

表示当前枚举第 dep 位，目前这个数的数位和是 cur，对 sum 取模是 mod.

cur = sum 且 mod = 0 的个数要统计进答案。

BZOJ 1799 同类分布

```
void solve(){
    long long L, R;
    scanf("%lld%lld",&L,&R);
    printf("%lld\n",work(R)-work(L-1));
}
int main(){
    int tc=1;
    while(tc--) solve();
    return 0;
}
```

BZOJ 1799 同类分布

```
long long work(long long n){
    a[0]=0;
    for(;n;n/=10)    a[++a[0]]=n%10;
    long long ret=0;
    for(int i=1;i<=a[0]*9;i++){
        sum=i;
        memset(f,-1,sizeof(f));
        ret+=dp(1,a[0],0,0);
    }
    return ret;
}
```

BZOJ 1799 同类分布

```
long long f[21][170][170];
int a[22],tt=0,sum;

long long dp(int eq,int dep,int cur,int mod){
    if(cur>sum)  return 0;
    if(!dep)    return mod==0 && cur==sum;
    if(!eq && ~f[dep][cur][mod])    return f[dep][cur][mod];
    int ed=(eq)?a[dep]:9;
    long long ret=0;
    for(int i=0;i<=ed;i++)   ret+=dp(eq&&(i==ed),dep-
1,cur+i,(mod*10+i)%sum);
    if(!eq) f[dep][cur][mod]=ret;
    return ret;
}
```

下节课再见