

动态规划—数位DP 学习笔记

定义

引入

数位 DP 往往都是这样的题型：给定一个区间 $[l, r]$ ，求这个区间中满足某种条件的数的总数。

简单的暴力代码如下：

```
1 | int ans = 0;
2 | for(int i = l; i <= r; ++i)
3 |     if(check(i)) ++ans;
```

而当数据规模过大，暴力枚举就 T 飞了，因此引入数位 DP：

概念

数位 (digit)：对于十进制，即把一个数字按照个位、十位、百位等，一位一位地拆开，它每一位上的数字，也就是 $0 \sim 9$ ；其他进制可类比十进制。

数位 DP：一种按照数位暴力枚举的方式，用来解决一类特定问题；这种问题比较好辨认，一般具有这几个特征：

1. 提供一个数字区间（有时也只提供上界）来作为统计的限制；
2. 统计满足某种条件的数的数量，有时也有统计总和、平方和等的；
3. 上界很大，甚至会有 10^{18} 这么大，暴力枚举验证会超时；
4. 这些条件经过转化后可以使用「数位」的思想去理解和判断。

原理

例如，当我们在数数的过程中， $100 \sim 199$ 和 $200 \sim 299$ 这两部分，后两位是完全相同的，这种重复计算可以通过 DP 的方式进行优化。

实现

计数原理

数位 DP 中通常会利用常规计数问题技巧，比如把一个区间内的答案拆成两部分相减，即查分的思路：

$$ans_{[l,r]} = s_r - s_{l-1}.$$

一般根据是否计入 0 的贡献，将 s_k 定义为： $\sum[0, k]$ 或 $\sum[1, k]$ 。

数的存储

一般将数字中较低位存在数组的低位之中，即：

```
1 | typedef long long ll;
2 | ll solve(ll x) {
3 |     int len = 0;
4 |     while (x) a[++len] = x % 10, x /= 10;
5 |     return dfs(...); //记忆化搜索
6 | }
```

常用形参

统计答案可以选择记忆化搜索，也可以选择循环迭代递推；因为数位 DP 的预处理一般比较变态，所以我一般使用记忆化搜索。

常用的形式参数如下：

1. `pos` (int)：表示当前枚举的位置，一般从 `len` 开始，到 0 为止。
2. `limit` (bool)：表示当前枚举到的位置，可以填的数是否收到限制；若为 `true`，则该位最大填 a_{pos} ；否则最大填 $R - 1$ ，其中 R 表示枚举的进制数。
3. `sum` (int)：表示从 `len` 到 `pos + 1` 位的贡献，常用的有求和等。
4. `last` (int)：表示上一位填的数，当题目限制连续的两个（或多个）数位有条件限制的话常用。
5. `lead0` (bool)：表示从 `len` 到 `pos + 1` 是否都为 0（前导零）。
6. `r` (int)：表示从 `len` 到 `pos + 1` 这个前缀模一个数 `mod` 的结果，也可以表示数位和取模的结果。
7. `st` (bool)：常用与状态压缩，其二进制表示某一位是否满足某一条件等。

如何复用结果

简单分析可知，一定是已经求解过中，状态与当前状态相同的，可以复用，如 `pos`、`sum`、`last` 相同等；特殊的，当 `limit == 1` 或 `lead0 == 1` 时，即当前位受到限制时，无需记录状态，因为这一状态不会频繁的复用，这种空间换时间价值不大。

即：

```
1  typedef long long ll;
2  ll f[N][M]; // DP 数组，第一维表示枚举到的数位，第二维表示当前的状态；默认为
3  -1
4  ll dfs(int pos, bool limit, int sum) {
5      if (!pos) return sum;
6      if (!limit && f[pos][sum] != -1) return f[pos][sum];
7      int up = limit ? a[pos] : 9;
8      ll res = 0; for (int i = 0; i <= up; ++i)
9          res = (res + dfs(pos - 1, limit && i == up, sum + i)) % MOD;
10     if (!limit) f[pos][sum] = res;
11     return res;
12 }
```

习题

见：<https://www.luogu.com.cn/training/384691>

Reference

- [1] <https://oi-wiki.org/dp/number/>
- [2] <https://blog.csdn.net/hzf0701/article/details/116717851>
- [3] https://blog.csdn.net/m0_63726942/article/details/127060217

本文来自博客园，作者：RainPPR，转载请注明原文链接：<https://www.cnblogs.com/RainPPR/p/number-dp.html>

合集：学习笔记

标签：算法，学习笔记