初探数位类统计问题

Keywords: 数位DP，二进制，异或。

——Lazycal

基本知识

- 动规
- [l,r] 意为 l<=且<=r的数
- [l,r) 意为 l<=且< r的数
- (l,r] 意为 l< 且<=r的数
- (l,r) 意为 l< 且< r的数
- 其实方括号意味着取等，小括号意味着不取等

Content

- 引入
- 基本思想与方法
- Hdu2089
- Hdu3652
- ural1057
- test-09-07-p1
- 总结
- 参考文献

引入

- “在信息学竞赛中，有一类与数位有关的区间统计问题。这类问题往往具有比较浓厚的数学味道，无法暴力求解，需要在数位上进行递推等操作。”——刘聪《浅谈数位类统计问题》
- 这类问题往往需要一些预处理，这就用到了数位DP。

基本思想与方法

- OI中经常需要统计区间[l,r]的满足题意的数的个数，这往往可以转换成求[0,r]-[0,l)
- 对于求区间[0,n)有一个通用的方法。
- 对于一个小于n的数，肯定是从高位到低位出现某一位<n的那一位。
- 如 n = 58 n为十进制数。
- x = 49 此时x的十位<n
- x = 51 此时x的个位<n

基本思想与方法

- 有了上述性质，我们就可以从高到低枚举第一次<n对应位是哪一位。
- 这样之前的位确定了，之后的位就不受n的限制即从00...0~99...9，可以先预处理，然后这时就可以直接统计答案。

基本思想与方法

- 预处理f数组。
- F[i,st] 代表 位数为i（可能允许前导0。如00058也是个5位数），状态为st的方案数。这里st根据题目需要确定。
- 如i=4,f[i,st]也就是0000~9999的符合条件的数的个数（十进制）
- 决策第i位是多少(such as 0~9)
- F[i,st] = F[i,st] + f[i–1,st']
- st'为相对应的状态

Hdu2089

- 题目链接：http://acm.hdu.edu.cn/showproblem.php?pid=2089
- 题目大意：给定区间[n,m]，求在n到m中没有“62“或“4“的数的个数。
- 如62315包含62，88914包含4，这两个数都是不合法的。
- 0<n<=m<1000000

Hdu2089

- 参照刚刚所说的基本思路。预处理f数组，然后统计[0,m] - [0,n)。
- f[i,j]代表开头是j的i位数中不含"62"或"4"的数有几个。
- 如f[2,6]包含60,61,63,65,66,67,68,69
- f[0,0] = 1;
- for i = 1 ~ 7
-   for j = 0 ~ 9 //枚举第i位
-     for k = 0 ~ 9 //枚举第i - 1位
-       if j <> 4 and not(j = 6 and k = 2)
-         f[i,j] = f[i - 1,k] + f[i,j];

Hdu2089

- 如f[2,6]的转移
- 6?   ? = 0,1,~~2~~,3,~~4~~,5,6,7,8,9
- f[2,6] = sum(f[1,j]) j = ?

Hdu2089

- 统计区间[0,n]
- 从高到低枚举哪一位比n小
- 如n = 4 5 6

4	5	6
↑	0	0
0/1…	0	1
…	…	…
	9	9

ans = ans + f[3,0]
ans = ans + f[3,1]
ans = ans + f[3,…]

Hdu2089

- 统计区间[0,n]
- 从高到低枚举哪一位比n小
- 如n = 4 5 6

$$\begin{array}{ccc} 4 & 5 & 6 \\ & \uparrow & \\ 4 & 0..4 & 0..9 \end{array}$$

ans = ans + f[2,0..4]

Hdu2089

- 伪代码:
- //digit[i] 代表 n 从右到左第i位是多少,len是n有几位。
- //如 n = 58 digit[1] = 8 digit[2] = 5

- for i = len ~ 1 //枚举哪一位<n的对应位
-   for j = 0 ~ digit[i] - 1 //枚举这一位的取值
-     if j <> 4 and not (j = 2 and digit[i + 1] = 6)
-       ans = ans + f[i,j]; //情况合法
-   if digit[i] = 4 or (digit[i] = 2 and digit[i + 1] = 6) break; //已经出现4或62

Hdu3652

- 题目链接：http://acm.hdu.edu.cn/showproblem.php?pid=3652
- 题目大意：求小于n是13的倍数且含有'13'的数的个数。

Hdu3652

- 同样参照前面的思想，先预处理，再统计。
- 题目需要包含13，且被13整除，我们就设计状态f[i,j,k,l]代表
- i位数中第一位是j的，
- 是否有包含13(k == 1 or 0)，
- 模13余数是l的数有几个。

Hdu3652

- 决策第i位：

```
for x = 0 ~ 9
  if k = 1 //要求要包含13
    f[i,j,k,l] = f[i - 1,x,1,(l - j*10^(i-1))%13];
    if j = 1 and x = 3 //已经有13了。
      f[i,j,k,l] = f[i,j,k,l] +
                   f[i - 1,x,0,(l - j*10^(i-1))%13];
  else //不要求包含13
    if not (j = 1 and x = 3)
      f[i,j,k,l] = f[i - 1,x,0,(l - j*10^(i-1))%13];
```

Hdu3652

- 统计小于n的合法的数有几个与上一题类似，只需要记录当前位之前的余数是多少，和是否已经出现了13

```
bit[0] = 1;
for (ll i = 1; i <= 12; ++i) bit[i] = bit[i - 1]*10;
for (ll i = digit[0],mod = 0; i; --i) {
    for (ll j = 0; j < digit[i]; ++j) {
        ans += f[i][j][1][(13 - mod*bit[i]%13)%13];
        if (t || (j == 3 && digit[i + 1] == 1))
           ans += f[i][j][0][(13 - mod*bit[i]%13)%13];
     }
     if (digit[i + 1] == 1 && digit[i] == 3) t = 1;
     mod = (mod*10 + digit[i])%13;
}
```

ural1057

- 题目链接：http://acm.hust.edu.cn/vjudge/problem/viewProblem.action?id=18851或http://acm.timus.ru/problem.aspx?space=1&num=1057
- 题目大意：求给定区间[X,Y]中满足下列条件的整数个数：这个数恰好等于K个互不相等的B的整数次幂之和。例如，设X=15，Y=20，K=2，B=2，则有且仅有下列三个数满足题意：
- 17 = 2^4+2^0,
- 18 = 2^4+2^1,
- 20 = 2^4+2^2.
- 1 ≤ X ≤ Y ≤ 2^31-1，1 ≤ K ≤ 20， 2 ≤ B ≤ 10。

ural1057

- 所求的数为互不相等的幂之和，亦即其B进制表示的各位数字都只能是0和1。
- 因此，我们只需讨论二进制的情况，其他进制都可以转化为二进制求解。
- 本题区间满足区间减法，因此可以进一步简化问题：令count[i..j]表示[i..j]区间内合法数的个数，则count[i..j]=count[0..j]-count[0..i-1]。
- 换句话说，给定n，我们只需求出从0到n有多少个符合条件的数。

ural1057

- 首先预处理f
- f[i,j]代表i位二进制数中恰好有j个1的数的个数。
- f[i,j]=f[i-1,j]+f[i-1,j-1]

- 计算count[0..n]
- 像前几题一样，一位一位枚举，只需要多记录后面需要的1的个数即可。
- if digit[i] = 1 then ans = ans + f[i,need]
- need就是后面需要的1的个数。

ural1057

- 最后的问题就是如何处理非二进制。
- 对于询问n，我们需要求出不超过n的最大B进制表示只含0、1的数：找到n的左起第一位非0、1的数位，将它变为1，并将右面所有数位设为1。
- 将得到的B进制表示视为二进制进行询问即可。
- 如n = (10204)9进制
- n = (10111)2进制

test-09-07-p1

- 题目大意：给定长度为n的序列A[i]，求所有A[i] xor A[j] (i<j)的值之和。

test-09-07-p1

- 还记得这题吧……
- 现在看是不是很水……
- 一位一位的处理。
- 统计这个数之前这一位有几个是0
- 然后根据当前位来处理。

拓展：spoj Sorted bit squence

- 题目链接：http://www.spoj.pl/problems/SORTBIT
  - or http://acm.hust.edu.cn/vjudge/problem/viewProblem.action?id=18852
- 题目大意：参照论文。
- 分析：参照论文。
- ……

Conclusion

- "解决问题的核心思想就是“逐位确定”思想。"
- “由于基本操作的复杂度是O(log(n))级别的，因此在处理一些较繁琐问题时，可以适当牺牲时间复杂度，对一些子问题采用二分、穷举等方法以降低思考和编程复杂度。”
- 对于求区间[l,r]的符合题目的数的个数，往往可以用[0,r] - [0,l)

参考文献

- 算法合集之《浅谈数位类统计问题》——刘聪
- http://hi.baidu.com/billdu/item/c749952ab2ab50c2ef10f137