简单模拟赛题解

samjia2000

中山纪念中学
Sun Yat-sen Memorial Secondary School

One?One!

$\frac{x}{\underbrace{11..11}_{d个1}} = \frac{9x}{10^d-1}$

答案即求：$\sum\limits_{d=2}^{l+1} \lfloor \frac{9n}{10^d-1} \rfloor$

记$N = 9n$

对于d，设$N = 10^d x + y (0 \leq y < 10^d)$，那么$\frac{N}{10^d-1} = x + \frac{x}{10^d-1} + \frac{y}{10^d-1}$，也就是说，我们计算$\frac{N}{10^d-1}$的过程可以不断地变成上面的形式之后再去计算$\frac{x}{10^d-1}$。

可以发现，如果不管每次计算出的y，那么第i位对第j位的贡献就是$i-j$的大于1的约数个数，这是个卷积的形式，可以用FFT算出这个贡献。

然后y的部分，实际上是，将N分成若干段长度为d的数字，然后将这些数字加起来再计算一次，由于生成的n可以看成是随机的，所以只需要保留每一段里的前15位做高精度加法即可。

时间复杂度$O(l \log l)$

Two?Two!

设DP状态$f_{i,j}$表示安排完前i个人，$1..i$中的最大值（记为M_i）不在的那条队里的最大值是j的最小不愉悦值。
如果$x_i \neq M_i$，那么

$$f_{i,j} = \begin{cases} f_{i-1,j} + M_i - x_i & j \in [0, x_i) \\ \min\{f_{i-1,k} | k \leq j\} & j = x_i \\ f_{i-1,j} + j - x_i & j \in (x_i, M_i] \end{cases} \quad (1)$$

否则，若$x_i = M_i$，那么$f_{i,j} = f_{i-1,j}$
那么就得到了$O(n^2)$的DP了.

Two?Two!

设DP状态$f_{i,j}$表示安排完前i个人，$1..i$中的最大值（记为M_i）不在的那条队里的最大值是j的最小不愉悦值。
如果$x_i \neq M_i$，那么

$$f_{i,j} = \begin{cases} f_{i-1,j} + M_i - x_i & j \in [0, x_i) \\ \min\{f_{i-1,k} | k \leq j\} & j = x_i \\ f_{i-1,j} + j - x_i & j \in (x_i, M_i] \end{cases} \quad (1)$$

否则，若$x_i = M_i$，那么$f_{i,j} = f_{i-1,j}$
那么就得到了$O(n^2)$的DP了.
容易发现，可以用分块维护每一块的凸壳，这样就可以得到时间复杂度为$O(n\sqrt{n})$的做法。

Two?Two!

考虑$f_{i,j}$和$f_{i,j+1}$，当$f_{i,j} \le f_{i,j+1}$时，显然$f_{i,j+1}$是没有意义的。

Two?Two!

考虑$f_{i,j}$和$f_{i,j+1}$，当$f_{i,j} \leq f_{i,j+1}$时，显然$f_{i,j+1}$是没有意义的。
那么，对于f_i，我们只维护k个有用的状态，即$t_1 \leq t_2 \leq \cdots \leq t_k$，他们对应的DP值为$d_1 > d_2 > \cdots > d_k$。
加入x_i的影响的时候：

1. 对于$t_j < x_i$，$d'_j = d_j + M_i - x_i$
2. 对于$t_j = x_i$，$d'_j = d_j$
3. 对于$t_j > x_i$，$d'_j = d_j + t_j - x_i$

对于情况1，只需要给一段前缀做加法即可。
对于情况2，不会产生影响。

Two?Two!

对于情况3，假设d_j和d_{j+1}分别被加了k次t_j以及t_{j+1}，那么，需要满足等式：$d_j + kt_j > d_{j+1} + kt_{j+1}$，得到：$k < \frac{d_j - d_{j+1}}{t_{j+1} - t_j}$
也就是说，当k大于等于$\frac{d_j - d_{j+1}}{t_{j+1} - t_j}$时，$(t_{j+1}, d_{j+1})$就会变成无用状态，可以删去。
于是可以用平衡树维护这些关键点，并且维护$\frac{d_j - d_{j+1}}{t_{j+1} - t_j}$的最小值，当这个最小值小于0的时候说明出现了无用的状态，去掉对应的点即可。
由于加入状态以及去掉状态的总次数是$O(n)$的，所以总的时间复杂度是$O(n \log n)$.

More?More!

记$f_{n,i}$表示有n个人，集合内有i个人的概率。
那么分别考虑在$1..n$加入$n+1$和在$2..n+1$加入1得到：

$$\begin{aligned} f_{n+1,i} = & \quad f_{n,i} \times p^i + f_{n,i-1} \times (1-p)^{n-i+1} \\ = & \quad f_{n,i} \times (1-p)^i + f_{n,i-1} \times p^{n-i+1} \end{aligned}$$

得到：

$$f_{n,i}(p^i - (1-p)^i) = f_{n,i-1}(p^{n-i+1} - (1-p)^{n-i+1})$$

当$p \neq \frac{1}{2}$时，直接扫一遍过去就好了。
当$p = \frac{1}{2}$的时候，$f_{n,i} = \binom{n}{i} \times (\frac{1}{2})^{i(n-i)}$
时间复杂度$O(n \log Mod)$