EASYEX解题报告

宁波市镇海蛟川书院 施舟行

1 题目描述

1.1 题目来源

EASYEX from Codechef July Challenge 2015.

1.2 题目大意

有一个有$K(K \leq 10^9)$面的骰子，每个面上的数字分别是1到K。同时给出两个参数L和$F(0 < L \leq K, 0 < F \leq 1000, 0 < L \times F \leq 50000)$。

现在将这个骰子掷$N(N \leq 10^9)$次。令a_i为掷出数字i的次数。求$a_1^F * a_2^F * ... * a_L^F$的期望值。

1.3 子任务

- 子任务1：$K = L = 1$。
- 子任务2：$K = L = 2, N \leq 100000$。
- 子任务3：$K \leq 10, N \leq 20$。
- 子任务4：$F = 1, L \times F \leq 2000$。
- 子任务5：$L \times F \leq 2000$。

2 算法讨论

2.1 算法一

对于第一个子任务，满足$K = 1$，也就是这个骰子只有一面，每次掷得的数字都为1。则有$a_1 = N$，故$a_1^F = N^F$。所以答案也就是N^F。算法复杂度为$O(1)$。

2.2 算法二

对于第二个子任务，骰子只有两面，且掷骰子的总次数不大。不妨考虑枚举两种数字各自出现的次数，满足$a_1 + a_2 = N(0 \le a_1, a_2 \le N)$。将每种情况下的$a_1^F * a_2^F$的值乘上出现的方案数相加，最后再除上总方案数。答案也就是：

$$\frac{\sum_{k=1}^{N} k^F (N-k)^F * \binom{N}{k}}{2^N}$$

算法复杂度为$O(N)$。

2.3 算法三

对于第三个子任务，N和K都不大。这时，可以枚举每种数字出现的次数，求得$a_1^F * a_2^F * \cdots * a_L^F$，再求得各自出现的概率，相加即可。算法复杂度为$O(\binom{N+K-1}{K-1})$。

2.4 算法四

a_i可以看做是a_i个1的和，其中的每一个1，都代表数字i在某次掷骰子时被掷出。倘若将$a_1^F * a_2^F * \cdots * a_L^F$展开，可以发现，这个值就是若干1的乘积的和。根据期望的线性性，我们尝试把每一部分，即每个若干1的乘积分开来计算，求和即可。

当$F = 1$时，每一各个数字在投掷过程中出现的时刻形成的组合，都会对答案的分子产生贡献，其贡献值也就是其他时刻掷骰子所得到点数的方案数。若贡献不为0，则满足L个数字在过程中都出现过，方案数就是$\binom{N}{L} \times L!$，其他时间掷骰子所得点数的方案数为K^{N-L}，因此，答案的分母也就是$\binom{N}{L} \times L! \times K^{N-L}$。算法复杂度为$O(1)$。

2.5 算法五

当$F \neq 1$时，我们先来看一个例子。假如某数字i在3个时刻出现，且$F=2$。为了便于描述，我们将这个数字出现的三个时刻用$x,y,z(x=y=z=1)$来表示。则$a_i^F=a_i^2=(x+y+z)^2=x^2+y^2+z^2+2xy+2xz+2yz$。其中，$x^2,y^2,z^2$产生贡献时只需要占用1个时刻，而$xy,xz,yz$产生贡献时需要占用2个时刻。这提示我们，可以尝试关于占用时刻数目进行DP。令$dp_{i,j}(0 \le i \le L)$为当前已经考虑了前i个数字，它们对答案产生贡献时，需要占用j个时刻的方案数。枚举当前第i个数字占用的时刻数目k，可以得到转移方程：

$$dp_{i,j} = \sum_{k=1}^{\min(j,F)} dp_{i-1,j-k} \times f_k \times \binom{j}{k}$$

其中f_k代表一个数字要占用k个时刻的方案数。比如，在上一个例子中，占用一个时刻的有x^2,y^2,z^2，方案数都为1；而占用两个时刻的有$2xy,2xz,2yz$，方案数为2。容易得到，可以用下式来求得$f_i(1 \le i \le F)$：

$$f_i = i^F - \sum_{j=1}^{i-1} f_j \times \binom{i}{j}$$

根据dp的结果，就可以得到答案的分子，如下：

$$\sum_{i=1}^{L\times F} dp_{L,i}\binom{N}{i} \times K^{N-i}$$

算法复杂度为$O(F^2+L^2F^2)$。

2.6 算法六

注意到，上述的dp的转移过程与求多项式的卷积极为相似。事实上，$dp_{L,i}$的值，也就是一个生成函数L次幂的i次项的系数。我们注意到，不同的数字出现的顺序不同，是被算作多种方案的。因此，这里可以使用指数生成函数。通过构造，得到如下指数生成函数：

$$G(x) = \sum_{i \ge 0} f_i \frac{x^i}{i!}$$

则：

$$dp_{L,i} = [x^i]G^L(x)$$

通过快速幂和FFT，可以在$O(LFlog(LF)logL)$的复杂度内求得$dp_{L,i}(1 \le i \le LF)$。总复杂度降至$O(F^2 + LFlog(LF)logL)$。

至此，问题解决。