Code Jam 2009 - World Finals

Analysis: Year of More Code Jam

The setting of this problem is no doubt discrete probability. From the definition, the space consists of N^T equally likely possible outcomes. That can be, under our limits, as huge as 10^{450}. Clearly, a naive approach is not feasible.

But let us do a little exercise in probability. Define the random variable X_i to be the number of contests on the i-th day, the quantity we want to compute is the average of X_i^2, i.e., the expectation $E(\sum X_i^2)$. By the linearity of the expectation, we have

$$E(\sum_{1\le i\le N} X_i^2) = \sum_{1\le i\le N} E(X_i^2).$$

So let us focus on the computation of the variable for a fixed day for the moment. Pick any i, and let $X := X_i$. Let us define more random variables. Define Y_j to be the *indicator* of whether the j-th tournament will have a contest on the i-th day. Clearly, $X = \sum_j Y_j$. So,

$$(*) \quad E(X^2) = E((\sum_{1\le j\le T} Y_j)^2) = \sum_{1\le j\le T} E(Y_j^2) + 2 \sum_{1\le j<k\le T} E(Y_jY_k).$$

We observe that each terms in the last expression is easy to compute. Being the indicator random variables, the *Y*'s take value 0 or 1. So

- Y_j^2 always has the same value as Y_j, and its expectation is just the probability that Y_j is 1, i.e., tournament j has a contest on day i.
- Y_jY_k is 1 if and only if both Y_j and Y_k are 1. The expectation is the probability that both the j-th and the k-th tournament has a contest on day i.

Let the input for the tournament j, i.e., the contest pattern be $d_1=1$, d_2, ..., d_m. Denote D(i,j) be the number of the d's for tournament j that are less than or equal to i. There are *N* choices for the starting date of a particular tournament. It is easy to see that the first probability above is D(i,j) / N; while the second probability is D(i,j)D(i,k) / N^2.

So far we addressed the problem just for a single day i. We need to do this for every i. There are 10^9 of them. But notice that, as long as there is no input d_t = i, D(i-1, j) = D(i, j) for all j. This means that the expectation for the i-th day is the same as the expectation for the (i-1)-th. There are at most T max(M) ≤ 2500 such d's in the input, so we need to compute (*) for at most 2500 days. For our problem, it is good enough to realize that there is no d > 10000. So all the expectations after the 10000-th day are the same. We can just do the computation for the first 10000 days, and for the rest, a simple multiplication.

The last problem is the need for big integers. At the first glance we might have both numerators and denominators as big as 10^{450}. But that is not the truth. Simply observe the above answer, which is a sum of various D(i,j) / N and D(i,j)D(i,k) / N^2. We actually proved that the denominator is never bigger than N^2. A careful implementation with 64-bit integers will be good enough.

For a further speed-up. The formula in (*) involves computing $O(T^2)$ terms. But if we do it from day 1, keep D(i,j) for each j and two more variables -- S_1 for the sum of all the D(i,j)'s, and S_2 for the sum of $D(i,j)^2$'s, then we just need constant update time when we see an input d, and also constant computation time for each day we want to compute (*).