Codeforces Problem 1916H1

> **Problem.** For each r where $0 \leq r \leq m$, count the number of $n \times n$ matrices over $\mathbb{F}_p$ with rank r.

Solution. After quite a bit of thinking and structuring the problem, we arrive at the following realization, which allows us to somewhat easily solve for the quantities we're looking for: for each $0 \leq r \leq m$,

$$\sum_{k=0}^{r} \mathsf{sub}[r, k] \cdot \mathsf{ranks}[k] = p^{nr},$$

where $\mathsf{sub}[r, k]$ denotes the number of dimension k subspaces that a dimension r vector space over $\mathbb{F}_p$ has, and $\mathsf{ranks}[k]$ denotes the number of $n \times n$ matrices with rank k formed using vectors from a fixed k dimensional vector space. Our answer for each r is then $\mathsf{sub}[n, r] \cdot \mathsf{ranks}[r]$.

In other words, we have a system of $n + 1$ equations in our $n + 1$ unknowns that we wish to solve for, namely each $\mathsf{ranks}[k]$. Thus, we've reduced the problem down to counting the number of subspaces of a vector space over this finite field.

Claim. We have that

$$\mathsf{sub}[r, k] = \frac{(p^r - 1)(p^r - p) \cdots (p^r - p^{k-1})}{(p^k - 1)(p^k - p) \cdots (p^k - p^{k-1})}.$$

Proof. We would like to be able to pick out k-dimensional subspaces of a r-dimensional vector space, so we do so by picking k linearly independent vectors to form a basis. However, this overcounts by quite a bit, so we must divide by the number of ways to choose k linearly independent vectors from the subspace that we found. ∎

Since the denominator of $\mathsf{sub}[r, k]$ can be precomputed in reasonable time and the numerator can be computed with an easy recursion, we can solve for $\mathsf{ranks}[k]$ and thus find the answer in something like $O(k^2 \log n)$ time, where the logarithm factor appears because of calculating the exponentials modulo the fixed prime P.