Spring 2021

MATH 1025: Introduction to Cryptography

Bonus 1

unstable cipher

Problem 1.

(a) [2 **pts**] Let $1 \leq k \leq 26$ and find how many simple substitution ciphers have at least k letters fixed.

(b) [3 **pts**] Find how many simple substitution ciphers have no letters fixed. [1]

(c) [3 **pts**] Find the answer to (b) for a general n (instead of 26). Let us denote your answer by $\mathcal{D}_n$. Compute the limit $\lim\limits_{n\to\infty} \frac{\mathcal{D}_n}{n!}$. [2]

[1] **Hint:** look up and use the inclusion-exclusion principle, starting with all simple substitution ciphers, subtracting simple substitution ciphers that fix at least one letter, etc.

[2] **Hint:** you may use that $\lim\limits_{n\to\infty} \left(1+\frac{x}{n}\right)^n = e^x$.

Remark. As $n!$ is the number of all possible simple substitution ciphers, the number you have found in (c) above is the probability that a randomly chosen simple substitution cipher will not fix any elements. In other words, this number represents the share of permutations that fix no elements. Such permutations are known as *derangements*.

Problem 2 [2 **pts**] Write a program that computes $\mathrm{GCD}(a, b)$. The program should take two positive integers $a, b \in \mathbb{Z}_{>0}$ as an input and return a single number $\mathrm{GCD}(a, b)$. Acceptable formats: pseudocode, Python or C#.