- This problemset has *four* questions.
- For other questions, either send the solutions to `gawry1+aos@gmail.com`, or leave them in the envelope attached to the doors of my office (room 321).

1. Show that deciding whether a *system* of word equations is satisfiable is NP-hard, even if one side of each equation is constant.

Solution: We reduce one-in-three SAT. Given a conjunction of clauses $(x_{1,1} \vee x_{1,2} \vee x_{1,3}) \wedge \ldots \wedge (x_{k,1} \vee x_{k,2} \vee x_{k,3})$, where each $x_{i,j}$ is either a variable or its negation, we want to assign a true/false value to each variable so that in each $x_{i,1} \vee x_{i,2} \vee x_{i,3}$ exactly one element is true. For each variable p we create two variables X_p, Y_p in our system of equations. We add an equation $X_p Y_p = 1$ to enforce that either $X_p = 1$ and $Y_p = \epsilon$ or $X_p = \epsilon$ and $Y_p = 1$. The former corresponds to making p true, and the latter to making it false. Then for each $x_{i,1} \vee x_{i,2} \vee x_{i,3}$ we add one equation. For example, for $p \vee \neg q \vee r$ we add an equation $X_p Y_q X_r = 1$. Then the system of equations has a solution iff the conjuction is satisfiable.

2. Consider a word w over an alphabet Σ such that none of its two consecutive letters are identical. Show that there exist a partition of Σ into two disjoint sets Σ_ℓ and Σ_r such that there are at least $\frac{|w|-1}{4}$ appearances of pairs from $\Sigma_\ell \Sigma_r$ in w.

Solution: For each letter $a \in \Sigma$ put it in Σ_ℓ with probability $\frac{1}{2}$ and in Σ_r with probability $\frac{1}{2}$. The choice is independent for each letter. Then let S (as score) be the number of appearance of pairs from $\Sigma_\ell \Sigma_r$ in w. We have:

$$S = \sum_{i=1}^{|w|-1} [w_i \in \Sigma_\ell \vee w_{i+1} \in \Sigma_r] = \sum_{i=1}^{|w|-1} [w_i \in \Sigma_\ell] * [w_{i+1} \in \Sigma_r]$$

where $[b]$ is 1 when b is true and 0 otherwise. Because our choice of Σ_ℓ and Σ_r was random, S is a random variable. Let's calculate its expected value:

$$E[S] = E[\sum_{i=1}^{|w|-1} [w_i \in \Sigma_\ell] * [w_{i+1} \in \Sigma_r]] = \sum_{i=1}^{|w|-1} E[[w_i \in \Sigma_\ell] * [w_{i+1} \in \Sigma_r]]] =$$

We used the linearity of expectation. Now let's use the fact that w_i and w_{i+1} are different, so the corresponding events are independent:

$$\sum_{i=1}^{|w|-1} E[[w_i \in \Sigma_\ell] * [w_{i+1} \in \Sigma_r]]] = \sum_{i=1}^{|w|-1} E[[w_i \in \Sigma_\ell]] * E[[w_{i+1} \in \Sigma_r]]] = \sum_{i=1}^{|w|-1} \frac{1}{2} * \frac{1}{2} = \frac{|w|-1}{4}$$

But if the expected score is $\frac{|w|-1}{4}$, then for at least one choice of random values the resulting score must be at least $\frac{|w|-1}{4}$.

This only shows that a good partition exists. Actually, we can also find one efficiently and deterministically. If you want to know how, read about the method of conditional probabilities.

3. Consider a word-equation over one-variable (but perhaps many appearances of it) and its solution in a^*.
   (a) show that for each a: either there are no solutions from a^*, there is a unique such solution or each a^k is a solution of this equation;
   (b) devise a linear-time algorithm which, given a letter a and an equation, decides which of those three cases holds.

Solution: First consider the case $\Sigma = \{a\}$. Then the equation looks like

$$aaXaaXXaaa = XXaaXaa$$

The only thing that matters is the number of a's and X's on both sides. If there is an a on both sides, we can erase it. Similarly, if there is a X on both sides, we can erase it. After repeating this we will get a reduced equation of the form $a^i = X^j$. Now we consider two cases:

- $j = 0$, then if $i = 0$ any $X = a^k$ is a valid solution, if $i > 0$ there are no solutions,
- $j > 0$, then we should have $i = jk$, so either j doesn't divide i and we have no solutions, or the unique solution is $X = a^{i/j}$.

Now assume that the alphabet can contain other letters, so the equation looks like

$$aabXaaXXcabaa = bXXacaXaba$$

We erase all a's and all X's and check if the resulting word are the same. If not, there is no solution. If yes, so what's left looks like bcb, we construct one equation of the simpler type for each pair of consecutive letters, and for the first and last letter. Each equation is created by taking the removed part (consisting of a's and X's) on both sides, so for instance for bc we construct $XaaXX = XXa$. Then we check if the resulting system of equation has a solution, which can be done by considering each equation separately. If any of them has no solution, the whole system has no solution. If any of them has exactly one solution, we have at most one candidate. Finally, if for all of them any $X = a^k$ is a valid solution, then any such $X = a^k$ is a valid solution to the whole system.

This actually gives a linear time algorithm if you're careful.

(SPOJ) 4. For extra credit: solve Morphing is fun problem. A longer version of the statement with a story is available there, and a short version is: given an alphabet Σ of size at most 26 and a morphism $f : \Sigma \to \Sigma^+$, consider the sequence of words a, $f(a)$, $f(f(a))$, $f(f(f(a)))$, ... Such a sequence **converges** if for any $k = 0, 1, 2, \ldots$ the k-th letter of $f^i(a)$ eventually stabilizes, which means that the k-th letter of all $f^i(a)$ is the same, for $i = i_0, i_0 + 1, i_0 + 2, \ldots$ (or there is no k-th letter in all these words). Given a description of f, check if the corresponding sequence stabilizes.

Solution: This is just a hint: for each letter c check if $f^i(c)$ becomes a fixed finite word after a number of iterations. So, check if for some i we have $f^{i+1}(c) = f^i(c)$, and if so call the letter finite. Then for each non-finite letter d look at $f(d)$. First few letters of $f(d)$ could be finite, and then we have the first non-finite letter. Create an edge from d to this first non-finite letter. Then look at your starting letter a. If in the defined graph it lies on a cycle of size > 1, the sequence doesn't stabilize.