- This problemset has *three* questions.
- To get the credit for questions marked as SPOJ, you must get them accepted on `http://www.spoj.com/AOS`, but you **don't** have to send any explanation!
- 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. Prove that the lexicographical order is a total order:
   (a) for any string a: $a \leq a$,
   (b) for any strings a and b: if $a \leq b$ and $b \leq a$ then $a = b$,
   (c) for any strings a, b, and c: if $a \leq b$ and $b \leq c$ then $a \leq c$,
   (d) for any strings a and b: either $a \leq b$ or $b \leq a$.

2. The goal of this question is to make sure that you are familiar with binary search. The first part is straightforward, while the second requires some additional insight. Try to solve both!
   (a) You are given a sorted sequence of numbers $a_1 \leq a_2 \leq \ldots \leq a_n$. You can access any of them in constant time. Show how to check if the sequence contains a number x in just $\mathcal{O}(\log n)$ steps. Provide either some pseudocode or a clear description of your method.
   (b) You are given an $n \times n$ matrix containing numbers $a_{i,j}$. Each row and each column of the matrix is sorted, i.e., $a_{i,j} \leq a_{i+1,j}$ and $a_{i,j} \leq a_{i,j+1}$. You can access any $a_{i,j}$ in constant time. Show how to check if the matrix contains a number x in just $\mathcal{O}(n)$ steps. For extra credit: show an asymptotically better solution or prove that one cannot beat linear complexity here.

(SPOJ) 3. Extra credit: Given two strings x and y, find the minimum number of characters to be removed from x in order to obtain a string x' that does not contain y as a substring.