- This problemset has *four* questions. The points denote the intended difficulty.
- To pass, you need 2 points (partial points are possible).
- Please send the solutions to gawry1+phdopen@gmail.com
- The deadline is 29.02.2024 anywhere on Earth.

(1) 1. Show how to find, given s and t, a shortest substring of s that DOES NOT occur in t in linear time.

(1+1) 2. We want to store n natural numbers $x_1, x_2, \ldots, x_n$. Show how to save them in $\sum_i 2{+}2\lfloor \log_2 x_i \rfloor + o(n)$ bits, so that given an index i, we can return x_i in $\mathcal{O}(1)$ time. Try to further optimise the space while keeping the query time constant.

(1+1) 3. We are given a weight-balanced SLP of size g describing a string $s[1..n]$. Weight-balanced means that, for any production $A \to BC$, we have that lengths of the strings derived from B and C are within a multiplicative constant factor. The grammar can be preprocessed in $\mathcal{O}(g)$ time and space. Show how to store, for a given position b, $\mathcal{O}(\log^* n)$ words of information that allow us to later extract, given t, the substring $s[b..(b{+}t{-}1)]$ in $\mathcal{O}(t)$ time using the previously stored information. Try to design your algorithm so that it does need to know the value of t, and outputs $s[b], s[b+1], s[b+2], s[b+3], \ldots$ in worst-case constant time per character.

(1) 4. Explain in detail how to efficiently generate all secondary occurrences from the primary occurrences for a LZ77-compressed string $s[1..n]$ described by z phrases. State the space and the total time to generate all occ such occurrences (as functions of n, z, occ).