- This problemset has *three* 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. You are given a LZW parse (so, a sequence of n blocks, each block being either a single letter, or a previously defined block concatenated with a single character). Construct a small structure allowing you to access any letter of the corresponding text efficiently. For full credit, construct a structure of size $\mathcal{O}(n)$ allowing answering any such query in $\mathcal{O}(\log n)$ time, for partial credit the size of your structure can be larger, but should be $o(n^2)$.

2. Prove that for **any** text of length n over a fixed finite alphabet Σ, its LZW parse consists of at most $\mathcal{O}(\frac{n}{\log n})$ blocks (the constant hidden under the big-O depends on $|\Sigma|$, though).

3. A primitive square is a word of the form xx with x being primitive, which means that it is not possible to write $x = y^k$ with $k > 1$. For instance abaaba is a primitive square, but abababab is not. We want to get a good bound on the maximal number of subwords of a word of length n that are primitive squares. Note that if the same primitive square occurs multiple times in the word, we count it multiple times.

   (a) Prove a simplified version of the "three squares" lemma: if there are three primitive words x, y, z such that yy is a proper prefix of xx, and zz is a proper prefix of yy, then $|x| \geq 2|z|$.
   Hint: assume that $|x| < 2|z|$, draw a picture, then try to apply the periodicity lemma to deduce that z is actually not primitive.

   (b) Prove the following "three squares" lemma: if there are three primitive words x, y, z such that yy is a proper prefix of xx, and zz is a proper prefix of yy, then $|x| \geq |y| + |z|$.
   Hint: this is tricky and for extra credit.

   (c) Use the above lemma (in either version) to bound the number of primitive squares that all begin at the same position. Observe that multiplying this bound by n gives you a bound on the total number of primitive squares.

(SPOJ) 4. Given a word $w[1..n]$ find the largest k such that w contains a substring of the form u^k, for some nonempty word u.

   Hint: $\mathcal{O}(n \log n)$ is enough. Guess $|u|$ and try to compute the largest k in $\mathcal{O}(\frac{n}{|u|})$ time. using some longest common prefix/suffix queries.