Exercise 1.
Efficient searching with suffix arrays

In the lecture we discussed two strategies how to reduce the number of redundant character comparisons during a binary search. One uses the mlr values, while the other one makes use of lcp values. The mlr trick in practice already brings the running time to \(O(m + \log n)\).

- Find a pair of pattern and text where the mlr trick still needs time \(O(m \log n)\).
- For the same text and pattern perform the binary search using the lcp values.
- Prove that using the lcp method the search algorithm does at most \(O(m + \log n)\) character comparisons.

Exercise 2.
Given a text \(T\) of length \(n\), let \(suftab'\) be the suffix array of \(T\) where suffixes are lexicographically ordered according to the first \(m\) letters for some \(m < n\). Will the Kasai algorithm still compute the correct lcp values of adjacent suffixes in \(suftab'\)? Justify your answer!

Exercise 3.
Given the text \texttt{halloballo} construct the suffix array using the Skew algorithm.

Exercise 4.
Show that the worst case runtime of the skew algorithm for a text of length \(n\) is \(O(n)\).