Run-length compressed suffix arrays

Nicola Prezza

References and Reading

[1] Sections 11.1, 11.2 of: Navarro, Gonzalo. Compact data structures: A practical approach. Cambridge University Press, 2016.

Exercises

1 RLCSAs Consider the string L on which the FM-index of section 11.2 is based. A well-known fact is that, if the input text T is very repetitive (i.e. the set of distinct substrings is small), then the number r of equal-letter runs in L is very small. For this reason, r is usually considered to be a good measure of repetitiveness of T.

1.1 Explain (informally) why this is true. Hint: think about the origin of L as the list of characters preceding sorted suffixes.

1.2 Find an infinite family of texts with this property: each T in the family has $\Theta(|T|)$ equal-letter runs, but column L has $r \in O(1)$ equal-letter runs.

1.3 Propose an implementation of a run-length FM-index (RLFMI) taking advantage of this source of compressibility (i.e. the index should take advantage of the fact that $r \ll |T|$).

1.4 If L has r equal-letter runs, what can we say about Ψ? Can Ψ be compressed similarly? Propose an implementation of a run-length compressed suffix array (RLCSA) taking advantage of this source of compressibility.

1.5 Suppose $r \in \Theta(\sqrt{n})$, where $n = |T|$ (i.e. our text is polynomially compressible). How fast can we support `locate` queries on RLCSA/RLFMI indexes while using only $O(r)$ words of space? (e.g. with the solutions of exercises 1.3, 1.4)