Problem
The Maximum Number of Runs in a String

Bill Smyth[1,2]

[1] Algorithms Research Group, Department of Computing & Software
McMaster University, Hamilton, Ontario, Canada L8S 4K1
smyth@mcmaster.ca
www.cas.mcmaster.ca/cas/research/algorithms.htm

[2] Digital Ecosystems & Business Intelligence Institute
and Department of Computing, Curtin University, GPO Box U1987
Perth WA 6845, Australia
smyth@computing.edu.au

Given a nonempty string $\boldsymbol{u}$ and an integer $e \geq 2$, we call $\boldsymbol{u}^e$ a ***repetition***; if $\boldsymbol{u}$ itself is not a repetition, then $\boldsymbol{u}^e$ is a ***proper repetition***. Given a string $\boldsymbol{x}$, a ***repetition in x*** is a substring

$$\boldsymbol{x}[i..i{+}e|\boldsymbol{u}|{-}1] = \boldsymbol{u}^e,$$

where $\boldsymbol{u}^e$ is a proper repetition and neither $\boldsymbol{x}\big[i{+}e|\boldsymbol{u}|..i{+}(e{+}1)|\boldsymbol{u}|{-}1)\big]$ nor $\boldsymbol{x}[i{-}|\boldsymbol{u}|..i{-}1]$ equals $\boldsymbol{u}$. We say the repetition has ***period*** $|\boldsymbol{u}|$ and ***exponent*** e; it can be specified by the integer triple $(i, |\boldsymbol{u}|, e)$. It is well known [2] that the maximum number of repetitions in a string $\boldsymbol{x} = \boldsymbol{x}[1..n]$ is $\Theta(n \log n)$, and that the number of repetitions in $\boldsymbol{x}$ can be computed in $\Theta(n \log n)$ time [2, 1, 10].

A string $\boldsymbol{u}$ is a ***run*** iff it is periodic of (minimum) period $p \leq |\boldsymbol{u}|/2$. Thus $\boldsymbol{x} = abaabaabaabaab = (aba)^4ab$ is a run of period $|aba| = 3$. A substring $\boldsymbol{u} = \boldsymbol{x}[i..j]$ of $\boldsymbol{x}$ is a ***run*** or ***maximal periodicity in x*** iff it is a run of period p and neither $\boldsymbol{x}[i{-}1..j]$ nor $\boldsymbol{x}[i..j{+}1]$ is a run of period p. The run $\boldsymbol{u}$ has ***exponent*** $e = \lfloor |\boldsymbol{u}|/p \rfloor$ and possibly empty ***tail*** $\boldsymbol{t} = \boldsymbol{x}[i{+}ep..j]$ (proper prefix of $\boldsymbol{x}[i..i{+}p{-}1]$). Thus

$$\begin{array}{cccccccccccccccc} & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ \boldsymbol{x} & = & b & a & a & a & b & a & a & b & a & a & b & a & b & a \end{array}$$

contains a run $\boldsymbol{x}[3..12]$ of period $p = 3$ and exponent $e = 3$ with tail $\boldsymbol{t} = a$ of length $t = |\boldsymbol{t}| = 1$. It can be specified by a 4-tuple $(i, p, e, t) = (3, 3, 3, 1)$. and it includes the repetitions $(aab)^3$, $(aba)^3$ and $(baa)^2$ of period $p = 3$. In general it is easy to see that for $e = 2$ a run ***encodes*** $t{+}1$ repetitions;

for $e > 2$, p repetitions. Clearly, computing all the runs in $\boldsymbol{x}$ specifies all the repetitions in $\boldsymbol{x}$. The idea of a run was introduced in [9].

Let $r_{\boldsymbol{x}}$ denote the number of runs that actually occur in a given string $\boldsymbol{x}$, and let $\rho(n)$ denote the maximum number of runs that can possibly occur in any string $\boldsymbol{x}$ of given length n. A string $\boldsymbol{x} = \boldsymbol{x}[1..n]$ such that $r_{\boldsymbol{x}} = \rho(n)$ is said to be ***run-maximal***.

In [7, 8] it was shown that there exist universal positive constants k_1 and k_2 such that

$$\rho(n)/n < k_1 - k_2 \log_2 n/\sqrt{n},$$

but the proof was nonconstructive and provided no way of estimating the magnitude of k_1 and k_2. In [7], using a brute force algorithm, a table of $\rho(n)$ was computed for $n = 5, 6, \ldots, 31$, giving also for each n an example of a run-maximal string; for every n in this range, $\rho(n)/n < 1$ and $\rho(n) \leq \rho(n-1)+2$. In [5] an infinite sequence $X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots.\}$ of strings was described, with $|\boldsymbol{x_{i+1}}| > |\boldsymbol{x_i}|$ for every $i \geq 1$, such that

$$\lim_{i \to \infty} r_{\boldsymbol{x_i}}/|\boldsymbol{x_i}| = \frac{3}{2\phi},$$

where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden mean. Moreover, it was conjectured that in fact

$$\lim_{n \to \infty} \rho(n)/n = \frac{3}{2\phi}. \tag{1}$$

Recently a different and simpler construction was found [6] to yield another infinite sequence X of strings for which the ratio $r_{\boldsymbol{x_i}}/|\boldsymbol{x_i}|$ approached the same limit; in addition, it was shown that for every $\epsilon > 0$ and for every sufficiently large $n = n(\epsilon)$, $\frac{3}{2\phi} - \epsilon$ provides an asymptotic lower bound on $\rho(n)/n$.

In 2006 considerable progress was made on the estimation of an upper bound on $\rho(n)/n$:

* $\rho(n)/n \leq 5.0$ [12];
* $\rho(n)/n \leq 3.48$ [11];
* $\rho(n)/n \leq 3.44$ [13];
* $\rho(n)/n \leq 1.6$ [3].

Thus the problem may be stated as follows:

Is conjecture (1) true?
If not, then characterize the function $\rho(n)/n$.

Help may be found in recent work studying the limitations imposed on the existence and length of runs in neighbourhoods of positions where two runs are known to exist [4, 14].

References

1. Alberto Apostolico & Franco P. Preparata, **Optimal off-line detection of repetitions in a string**, *Theoret. Comput. Sci. 22* (1983) 297–315.
2. Maxime Crochemore, **An optimal algorithm for computing the repetitions in a word**, *Inform. Process. Lett. 12–5* (1981) 244–250.
3. Maxime Crochemore & Lucian Ilie, **Maximal repetitions in strings**, submitted for publication (2006).
4. Kangmin Fan, Simon J. Puglisi, W. F. Smyth & Andrew Turpin, **A new periodicity lemma**, *SIAM J. Discrete Math. 20–3* (2006) 656–668.
5. Frantisek Franek, R. J. Simpson & W. F. Smyth, **The maximum number of runs in a string**, *Proc.* 14^{th} *Australasian Workshop on Combinatorial Algs.*, M. Miller & K. Park (eds.) (2003) 26–35.
6. Frantisek Franek & Qian Yang, **An asymptotic lower bound for the maximum-number-of-runs function**, *Proc. Prague Stringology Conference '06*, Jan Holub & Jan Žd'árek (eds.) (2006) 3–8.
7. Roman Kolpakov & Gregory Kucherov, *Maximal Repetitions in Words or How to Find all Squares in Linear Time*, Rapport LORIA 98-R-227, Laboratoire Lorrain de Recherche en Informatique et ses Applications (1998) 22 pp.
8. Roman Kolpakov & Gregory Kucherov, **On maximal repetitions in words**, *J. Discrete Algs. 1* (2000) 159–186.
9. Michael G. Main, **Detecting leftmost maximal periodicities**, *Discrete Applied Maths. 25* (1989) 145–153.
10. Michael G. Main & Richard J. Lorentz, **An $O(n \log n)$ algorithm for finding all repetitions in a string**, *J. Algs. 5* (1984) 422–432.
11. Simon J. Puglisi, R. J. Simpson & W. F. Smyth, **How many runs can a string contain?**, submitted for publication (2006).
12. Wojciech Rytter, **The number of runs in a string: improved analysis of the linear upper bound**, *Proc.* 23rd *Symp. Theoretical Aspects of Computer Science*, B. Durand & W. Thomas (eds.), LNCS 2884, Springer-Verlag (2006) 184–195.
13. Wojciech Rytter, **The number of runs in a string**, submitted for publication (2006).
14. R. J. Simpson, **Intersecting periodic words**, submitted for publication (2006).