

New Lower Bounds for the Maximum Number of Runs in a String

Wataru Matsubara¹, Kazuhiko Kusano¹, Akira Ishino¹, Hideo Bannai², and Ayumi Shinohara¹

¹ Graduate School of Information Science, Tohoku University,
Aramaki aza Aoba 6-6-05, Aoba-ku, Sendai 980-8579, Japan
{matsubara@shino., kusano@shino., ishino@, ayumi@}ecei.tohoku.ac.jp

² Department of Informatics, Kyushu University,
744 Motooka, Nishiku, Fukuoka 819-0395 Japan.
bannai@i.kyushu-u.ac.jp

Abstract. We show a new lower bound for the maximum number of runs in a string. We prove that for any $\varepsilon > 0$, $(\alpha - \varepsilon)n$ is an asymptotic lower bound, where $\alpha = 174719/184973 \approx 0.944565$. It is superior to the previous bound $3/(1 + \sqrt{5}) \approx 0.927$ given by Franěk *et al.* [6,7]. Moreover, our construction of the strings and the proof is much simpler than theirs.

1 Introduction

Repetitions in strings is an important element in the analysis and processing of strings. It was shown in [9] that when considering *maximal repetitions*, or *runs*, the maximum number of runs $\rho(n)$ in any string of length n is $O(n)$, leading to a linear time algorithm for computing all the runs in a string. Although they were not able to give bounds for the constant factor, there have been several works to this end [12,13,11,2,1,8]. The currently known best upper bound³ is $\rho(n) \leq 1.048n$ [3], obtained by calculations based on the proof technique of [2]. The technique bounds the number of runs for each string by considering runs in two parts: runs with long periods, and runs with short periods. The former is more sparse and easier to bound while the latter is bounded by an exhaustive calculation concerning how runs of different periods can overlap in an interval of some length. On the other hand, an asymptotic lower bound on $\rho(n)$ is presented in [7], where it is shown that for any $\varepsilon > 0$, there exists an integer $N > 0$ such that for any $n > N$, $\rho(n) \geq (\alpha - \varepsilon)n$, where $\alpha = \frac{3}{1+\sqrt{5}} \approx 0.927$. It was conjectured in [6] that this bound is optimal.

In this paper, we prove that the conjecture was false, by showing a new lower bound $\alpha = 174719/184973 \approx 0.944565$. First we show a concrete string τ of length 184973, which contains 174697 runs in it. It immediately disproves the conjecture, since $174697/184973 \approx 0.944445$ is already higher than the previous bound 0.927. Then we prove that the string τ^k , which is the string obtained by concatenating k copies of τ , contains $174719k - 21$ runs for any $k \geq 2$. Since $|\tau^k| = 184973k$, it yields the new lower bound $174719/184973$ as $k \rightarrow \infty$.

2 Preliminaries

Let Σ be a finite set of symbols, called an *alphabet*. Strings x , y and z are said to be a *prefix*, *substring*, and *suffix* of the string $w = xyz$, respectively. The length of

³ Presented on the website <http://www.csd.uwo.ca/faculty/ilie/runs.html>

a string w is denoted by $|w|$. The i -th symbol of a string w is denoted by $w[i]$ for $1 \leq i \leq |w|$, and the substring of w that begins at position i and ends at position j is denoted by $w[i : j]$ for $1 \leq i \leq j \leq |w|$. A string w has period p if $w[i] = w[i + p]$ for $1 \leq i \leq |w| - p$. A string w is called *primitive* if w cannot be written as u^k , where k is a positive integer, $k \geq 2$.

A string u is a *run* if it is periodic with (minimum) period $p \leq |u|/2$. A substring $u = w[i : j]$ of w is a *run in w* if it is a run of period p and neither $w[i - 1 : j]$ nor $w[i : j + 1]$ is a run of period p , that means the run is maximal. We denote the run $u = w[i : j]$ in w by the triple $\langle i, j - i + 1, p \rangle$ consisting of the begin position i , the length $|u|$, and the minimum period p of u . A run of w which is a prefix (resp. suffix) of w is called a prefix (resp. suffix) run of w . For a string w , we denote by $run(w)$ the number of runs in w .

For example, the string `aabaabaaaacaacac` contains the following 7 runs: $\langle 1, 2, 1 \rangle = \mathbf{a}^2$, $\langle 4, 2, 1 \rangle = \mathbf{a}^2$, $\langle 7, 4, 1 \rangle = \mathbf{a}^4$, $\langle 12, 2, 1 \rangle = \mathbf{a}^2$, $\langle 13, 4, 2 \rangle = (\mathbf{ac})^2$, $\langle 1, 8, 3 \rangle = (\mathbf{aab})^{\frac{8}{3}}$, and $\langle 9, 7, 3 \rangle = (\mathbf{aac})^{\frac{7}{3}}$. Thus $run(\text{aabaabaaaacaacac}) = 7$.

We are interested in the behavior of the *maxrun function* defined by

$$\rho(n) = \mathbf{max}\{run(w) \mid w \text{ is a string of length } n\}.$$

Franěk, Simpson and Smyth [6] showed a beautiful construction of a series of strings which contains many runs, and later Franěk and Qian Yang [7] formally proved a family of true asymptotic lower bounds arbitrarily close to $\frac{3}{1+\sqrt{5}}n$ as follows.

Theorem 1 ([7]). *For any $\varepsilon > 0$ there exists a positive integer N so that $\rho(n) \geq \left(\frac{3}{1+\sqrt{5}} - \varepsilon\right)n$ for any $n \geq N$.*

3 Basic Properties

In this section, we summarize some basic properties concerning periods and repetitions in strings, which will be utilized in the sequel.

The next Lemma given by Fine and Wilf [5] provides an important property on periods of a string.

Lemma 2 (Periodicity Lemma (see [10,4])). *Let p and q be two periods of a string w . If $p + q - \gcd(p, q) \leq |w|$, then $\gcd(p, q)$ is also a period of w .*

For a string w , let us consider a series of strings $w, w^2, w^3, w^4 \dots$, and observe all runs contained in these strings. There are many cases, which confuse the task of counting the number of runs in these strings.

1. A run in w^k which is neither a suffix nor prefix run of w^k is also a run in w^{k+1} .
2. A suffix run in w^k and a prefix run in w may be merged into one run in w^{k+1} .
3. A suffix run in w^k may be extended to a run in w^{k+1} .
4. A new run may be newly created at the border between w^{k+1} and w .

Concerning case 4, note that a new run that did not appear in w or w^2 may be created in w^3 . For example, consider strings $w = \text{abcacabc}$, and $r = (\text{cabca})^2$. We can verify that r is a run $\langle 8, 10, 5 \rangle$ of $w^3 = \text{abcacabcabcacabcabcacabc}$, while r does not appear in $w^2 = \text{abcacabcabcacabc}$. Moreover, the same argument holds also for binary alphabet $0, 1$; Replace a, b, c into $01, 10, 00$, respectively in the above example.

However, the following lemma shows that the length of such new runs can be bounded.

Lemma 3. *Let w be a string of length n . For any $k \geq 3$, let $r = \langle i, l, p \rangle$ be a run in w^k . If $l \geq 2n$, then $i = 1$ and $l = kn$, that is, $r = w^k$.*

Proof. We assume that $n > 1$, since it is trivial for the case $n = 1$. Since p is the minimum period of the run r , we know $|r| = l \geq 2p$ and $l \geq 2n$. Let u be a primitive string of length m where $w = u^t$ for some integer $t \geq 1$. Then, $|u| = m \leq n$ is also a period of run r . Since $p + m \leq l$, Lemma 2 claims that $\gcd(p, m)$ is also a period of run r . If $p > m$, then $\gcd(p, m) < p$, which contradicts the assumption that p is the minimum period of r . If $p < m$, then it contradicts the assumption that u is primitive. Therefore we have $p = m$. Since m is a period of w^k , we have $r = \langle 1, kn, m \rangle = w^k$.

This lets us prove the following lemma which gives a formula for $run(w^k)$.

Lemma 4. *Let w be a string of length n . For any $k \geq 2$, $run(w^k) = Ak - B$, where $A = run(w^3) - run(w^2)$ and $B = 2run(w^3) - 3run(w^2)$.*

Proof. We think about the increase in the number of runs, when concatenating w^k and w . Let $r = \langle i, l, p \rangle$ be a run of w^{k+1} such that $i+l > nk+1$, that is, r ends somewhere in the last w of w^{k+1} . By Lemma 3, if $i \leq (k-2)n$ then $r = w^{k+1}$. In such a case, r does not increase the number of runs since the run will have already been considered in w^2 . Therefore, the increase in runs can be considered by restricting our attention to runs with $i > (k-2)n$, that is, the increase in runs for the last 3 w 's of w^{k+1} when concatenating w to the last 2 w 's of w^k . This gives us $run(w^{k+1}) - run(w^k) = run(w^3) - run(w^2)$.

$$\begin{aligned} run(w^k) &= run(w^{k-1}) + run(w^3) - run(w^2) \\ &= run(w^{k-2}) + 2(run(w^3) - run(w^2)) \\ &= run(w^2) + (k-2)(run(w^3) - run(w^2)) \\ &= k(run(w^3) - run(w^2)) - (2run(w^3) - 3run(w^2)) \end{aligned}$$

for $k \geq 3$. It is easy to see that the equation also holds for $k = 2$.

Theorem 5. *For any string w and any $\varepsilon > 0$, there exists a positive integer N such that for any $n \geq N$,*

$$\frac{\rho(n)}{n} > \frac{run(w^3) - run(w^2)}{|w|} - \varepsilon.$$

Proof. By Lemma 4, $run(w^k) = Ak - B$, where $A = run(w^3) - run(w^2)$ and $B = 2run(w^3) - 3run(w^2)$.

For any given $\varepsilon > 0$, we choose $N > \frac{A-B}{\varepsilon}$. For any $n \geq N$, let k be the integer satisfying $|w|(k-1) \leq n < |w|k$. Notice that $k > \frac{n}{|w|} \geq \frac{N}{|w|} \geq \frac{A-B}{|w|\varepsilon}$. Since $\rho(i+1) \geq \rho(i)$ for any i , and $|w^{k-1}| = |w|(k-1)$,

$$\begin{aligned} \frac{\rho(n)}{n} &\geq \frac{\rho(|w|(k-1))}{|w|k} \geq \frac{run(w^{k-1})}{|w|k} = \frac{A(k-1) - B}{|w|k} = \frac{Ak - A - B}{|w|k} \\ &= \frac{A}{|w|} - \frac{A-B}{|w|k} > \frac{A}{|w|} - \varepsilon. \end{aligned}$$

□

4 New Lower Bounds

We found some strings which contain many runs, by running a computer program which utilizes a simple heuristic search for run-rich binary strings. Given a buffer size, the search first starts with the single string 0 in the buffer. At each round, two new strings are created from each string in the buffer by appending 0 or 1 to the string. The new strings are then sorted in order of $run(w^3) - run(w^2)$, and only those that fit in the buffer are retained for the next round. Strings that give a high ratio of runs are recorded.

We tried several variations of the algorithm, and found many run-rich strings. Among these strings found so far, the string τ , lets us prove the currently best lower bound on the maximum number of runs in a string. Since τ is too long to include in the paper, we will make τ available on our web site ⁴. Once we have τ , it is straightforward to confirm that the following lemma holds. Any naïve program to count runs in a string would be sufficient.

Lemma 6. *There exists a string τ such that $|\tau| = 184973$, $run(\tau) = 174697$, $run(\tau^2) = 349417$, and $run(\tau^3) = 524136$.*

It immediately disproves the conjecture, since $174697/184973 \approx 0.944445$ is already higher than the previous bound $\frac{3}{1+\sqrt{5}} \approx 0.927$. We now show the main result of this paper.

Theorem 7. *For any $\varepsilon > 0$ there exists a positive integer N so that $\rho(n) > (\alpha - \varepsilon)n$ for any $n \geq N$, where $\alpha = \frac{174719}{184973} \approx 0.944565$.*

Proof. From Theorem 5 and Lemma 6, we have

$$\frac{\rho(n)}{n} > \frac{524136 - 349417}{184973} - \varepsilon = \frac{174719}{184973} - \varepsilon.$$

□

For proof of concept, we present in the Appendix, a shorter string τ_{1558} with $|\tau_{1558}| = 1558$, $run(\tau_{1558}) = 1455$, $run(\tau_{1558}^2) = 2915$, $run(\tau_{1558}^3) = 4374$ that gives a smaller bound $(4374 - 2915)/1558 \approx 0.93645$ compared to τ , but is still better than previously known.

5 Conclusion and Further Research

We presented a new lower bound $174719/184973 \approx 0.944565$ for the maximum number of runs in a string. The proof was very simple, once after we verified that the runs in the string τ is 174697, and noticed some trivial properties of the string. We do not think that the bound is optimal. We believe that our work would revive the interests to push the lower bound higher up, since the previous bound $3/(1 + \sqrt{5}) \approx 0.927$ was conjectured to be the optimal since 2003.

Further research will include trying to find properties of run-rich strings by analyzing strings obtaining from heuristic search. We believe that compression gives a clue to understanding the property of run-rich strings, since while τ has length 184973, it can be represent by mere 24 terms of LZ factors (see Appendix).

⁴ <http://www.shino.ecei.tohoku.ac.jp/runs/>

Acknowledgements: We thank Professor Rytter for useful comments about the compressibility of run-rich strings.

References

1. P. BATURO, M. PIĄTKOWSKI, AND W. RYTTER: *The number of runs in Sturmian words*, in Proc. CIAA 2008, 2008, To appear.
2. M. CROCHEMORE AND L. ILIE: *Maximal repetitions in strings*. J. Comput. Syst. Sci., 74 2008, pp. 796–807.
3. M. CROCHEMORE, L. ILIE, AND L. TINTA: *Towards a solution to the “runs” conjecture*, in Proc. CPM 2008, vol. 5029 of LNCS, 2008, pp. 290–302.
4. M. CROCHEMORE AND W. RYTTER: *Jewels of Stringology*, World Scientific, 2002.
5. N. FINE AND H. WILF: *Uniqueness Theorems for Periodic Functions*. Proceedings of the American Mathematical Society, 16(1) 1965, pp. 109–114.
6. F. FRANĚK, R. SIMPSON, AND W. SMYTH: *The maximum number of runs in a string*, in Proc. 14th Australasian Workshop on Combinatorial Algorithms (AWOCA2003), 2003, pp. 26–35.
7. F. FRANĚK AND Q. YANG: *An asymptotic lower bound for the maximal-number-of-runs function*, in Proc. Prague Stringology Conference (PSC’06), 2006, pp. 3–8.
8. M. GIRAUD: *Not so many runs in strings*, in Proc. LATA 2008, 2008, pp. 245–252.
9. R. KOLPAKOV AND G. KUCHEROV: *Finding maximal repetitions in a word in linear time*, in Proc. 40th Annual Symposium on Foundations of Computer Science (FOCS’99), 1999, pp. 596–604.
10. M. LOTHAIRE: *Algebraic combinatorics on words*, Cambridge University Press New York, 2002.
11. S. J. PUGLISI, J. SIMPSON, AND W. F. SMYTH: *How many runs can a string contain?* Theoretical Computer Science, 401(1–3) 2008, pp. 165–171.
12. W. RYTTER: *The number of runs in a string: Improved analysis of the linear upper bound*, in Proc. 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS 2006), vol. 3884 of LNCS, 2006, pp. 184–195.
13. W. RYTTER: *The number of runs in a string*. Inf. Comput., 205(9) 2007, pp. 1459–1469.

Appendix

The binary string τ_{1558} with $|\tau_{1558}| = 1558$, $run(\tau_{1558}) = 1455$, $run(\tau_{1558}^2) = 2915$, $run(\tau_{1558}^3) = 4374$, giving lower bound $(4374 - 2915)/1558 \approx 0.93645 > 0.927$.

```

110101101001011010110100101101011001101011010010110101101001011010
110010110101101001011010110100101101011001101011010010110101101001
011010110010110101101001011010110010110100101101011010010110101100
101101011010010110101101001011010110010110100101101011010010110101
100101101011010010110101100101101001011010110100101101011001011010
110100101101011010010110101100101101011010010110101100101101001011
010110100101101011001011010110100101101011010010110101100101101001
011010110100101101011001011010110100101101011001011010110010110101101
001011010110010110101101001011010110100101101011001011010110100101
101011001011010010110101101001011010110010110101101001011010110010
110100101101011010010110101100101101011010010110101101001011010110
010110100101101011010010110101100101101011010010110101100101101001
011010110100101101011001011010110100101101011010010110101100101101
011010010110101100101101001011010110100101101011001011010110100101
101011010010110101100101101001011010110100101101011001011010110100
101101011001011010010110101101001011010110010110101101001011010110
100101101011001011010110100101101011001011010010110101101001011010
110010110101101001011010110010110100101101011010010110101100101101
011010010110101101001011010110010110100101101011010010110101100101
101011010010110101100101101001011010110100101101011001011010110100
101101011010010110101100101101011010010110101100101101001011010110
100101101011001011010110100101101011010010110101100101101001011010
110100101101011001011010110100101101011001011010010110101101001011
010110010110101101001011010110100101101011010010110101101001011
    
```

By interpreting τ_{1558} as a binary representation of an integer, it can be expressed in hexadecimal representation by:

```

0x35A5AD2D66B4B5A5ACB5A5AD2D66B4B5A5ACB5A5ACB4B5A5ACB5A5AD2D65A5AD
2D65AD2D65A5AD2D65AD2D696B2D696B2D2D696B2D696B4B59696B4B596B4B5969
6B4B596B4B5A5ACB5A5ACB4B5A5ACB5A5ACB4B5A5ACB5A5AD2D65A5AD2D65AD2D6
5A5AD2D65AD2D696B2D696B2D2D696B2D696B4B59696B4B596B4B59696B4B596B4
B5A5ACB5A5ACB4B5A5ACB5A5ACB4B5A5ACB5A5AD2D65A5AD2D65AD2D65A5AD2D65
AD2D696B2D696B2D2D696B2D696B4B59696B4B596B4B59696B4B59696B4B5A5A
    
```

The string τ of Lemma 6 can be represented by 24 terms of LZ factors.
 $\tau = a / (0, 1) / b / (1, 3) / (1, 4) / (2, 8) / (5, 13) / (12, 19) /$
 $(26, 31) / (49, 38) / (50, 63) / (89, 93) / (113, 162) / (57, 317) /$
 $(249, 693) / (275, 984) / (879, 2120) / (942, 3041) / (2811, 6521) /$
 $(2999, 9374) / (8764, 20072) / (9332, 28878) / (27096, 45341) /$
 $(38210, 67195)$