The Expected Time to Find a String in a Random Binary Sequence

Terry R. McConnell
Syracuse University

January 29, 2001

1 Introduction

The proverbial monkey typing at random will, given sufficient time, produce the complete works of Shakespeare (along with near misses, good tries, and reams of outright garbage.) How long might this be expected to take? At the risk of being overly reductive, the works of Shakespeare can be viewed as one particular long string of characters. So, more generally, exactly how long is the expected waiting time for a given string to appear in a stream of random characters?

The first definitive answer seems to have been given by P.T. Nielsen [11], and rediscovered about 10 years later by G. Blom [2]. Given that the question is interesting and can be solved by relatively elementary means, one suspects that the solution has been rediscovered many times since Nielsen's paper (and possibly before.) Indeed, the author of one intermediate probability text (see [12, pp 186-187]) discusses the problem, and even hints at its general solution, without providing a reference.

In this paper we describe an algorithm for computing the expected time until the first appearance of a given string in random data, and also discuss some related problems. There is little that is new in the arguments we give – they are, for the most part, adapted from the ones given in the papers cited above. We also discuss the connection with some important problems in computation, such as searching efficiently for a given string in arbitrary but non-random text. To focus the ideas we shall confine our attention to the alphabet of binary digits, but all results can be generalized easily to the case of an arbitrary finite alphabet.

2 Expected Waiting Times

Consider the problem of finding the first '011' in a stream of random binary digits, or, equivalently, the first time a 'tails' is followed by two 'heads' in a sequence of fair coin-tosses. A simple procedure would run as follows: toss the coin 3 times and see whether the tosses produced 011; if not, toss the coin

another 3 times, and repeat until one of the three toss trials yields the desired sequence. Each such trial has probability $\frac{1}{8}$ of succeeding, and each trial is independent of the others. It follows that the number of trials, N, until the first successful one has a geometric distribution,

$$P(N = j) \quad = \quad \left(\frac{1}{8}\right)\left(\frac{7}{8}\right)^{j-1},$$

hence the expected value of N is 8.

Unfortunately, this simple analysis is not correct for the original problem. Suppose, for example, the first trial produced 010. This is a failure, but it is now possible that the target sequence can be obtained after the next *two* tosses (if they are 11). In other words, since success can sometimes occur before the end of a complete trial, trials are not independent. It easy to see that this argument does yield an upper bound of 24, the extra factor of three occurring because each trial requires 3 digits. In fact, as we shall see below, 8 is the correct result for this string, but there are other target strings (111, e.g.) for which 8 is not correct. The complete story has to do with the internal structure of the target string.

To state and prove a precise result, it is necessary to introduce some notation. Let $b_1, b_2, \ldots$ be a sequence of i.i.d. Bernoulli digits, i.e, each b_i takes only the values 0 and 1 with equal probability. If σ denotes a fixed finite sequence of binary digits, let X_n denote the Markov chain whose initial state is σ and whose successive states are obtained by adding the next binary digit on the right and dropping the leftmost digit. (Thus the state space of X_n comprises all finite binary sequences of length $|\sigma| = n$.)

If τ is a binary sequence of length $n-1$ and x is any binary digit, then the transition matrix

$$P_{x\tau,\sigma} \quad = \quad \frac{1}{2}$$

if either $\sigma = \tau 0$ or $\sigma = \tau 1$, and is 0 in all other cases. Thus, the transition matrix is doubly stochastic. It is well known that this implies a uniform stationary distribution, and hence we have

$$E_\sigma T_\sigma \quad = \quad 2^n, \tag{1}$$

where T_σ denotes the time X_n first returns to σ.

After these preliminary observations, it is now easy to prove the following theorem:

Theorem: Let τ be a binary string of length n and let T_τ be the least j such that τ occurs in $b_1 b_2 \ldots b_j$. Let σ be the longest proper suffix of τ that is also a prefix of τ. Then

$$E\,T_\tau \quad = \quad E\,T_\sigma \quad + \quad 2^n.$$

If τ has no proper suffix which is also a prefix, then we interpret σ as the empty string and the first term on the right-hand side as zero.

Proof: First suppose σ is non-empty. Let $\nu_1, \nu_2, \ldots$ be the successive times that σ occurs in $b_1, b_2, \ldots$. (We say that σ occurs at time t if $b_{t-k+1}b_{t-k+2}\ldots b_t = \sigma$, where $k = |\sigma|$.) Let

$$Y_j = \begin{cases} 1, & \text{if } \tau \text{ occurs during } \nu_j + 1 \ldots \nu_{j+1} \\ 0, & \text{else.} \end{cases}$$

Let $d_1, d_2, \ldots, d_l$ be the digits that follow σ in τ. Then $Y_j = 1$ if and only if $b_{\nu_j+1} = d_1, b_{\nu_j+2} = d_2, \ldots b_{\nu_j+l} = d_l$. Thus, by the strong Markov property of the sequence $b_1, b_2, \ldots$ the Y_j are independent and

$$P(Y_j = 1) = \left(\frac{1}{2}\right)^l = \left(\frac{1}{2}\right)^{n-|\sigma|}.$$

If N is the first $j \geq 1$ such that $Y_j = 1$, then N has a geometric distribution and

$$E\,N \;=\; 2^{n-|\sigma|}.$$

Next, observe that on $\{Y_j = 1\}$ we must have $\nu_{j+1} = \nu_j + l$ by the definition of σ. Thus

$$T_\tau \;=\; T_\sigma \;+\; \sum_{j=1}^{N}(\nu_{j+1} - \nu_j).$$

The $\nu_{j+1} - \nu_j$ are also i.i.d, and by (1) we have $E(\nu_{j+1} - \nu_j) = 2^{|\sigma|}$. Finally, by Wald's first lemma

$$E\sum_{j=1}^{N}(\nu_{j+1} - \nu_j) = E\,N\;E\,(\nu_2 - \nu_1) = 2^{n-|\sigma|}2^{|\sigma|} = 2^n,$$

and the result follows in this case.

Strangely, the case of an empty σ seems to require a completely different argument. Let us adopt some of Nielsen's terminology and call a proper suffix of τ which is also a prefix a *bifix* of τ. Thus, assume for the rest of the argument that τ is bifix free. Suppose x is the first digit of τ. Then x is the longest bifix of τx, so by the result for a nonempty bifix,

$$E\,T_{\tau x} \;=\; E\,T_x \;+\; 2^{n+1} \;=\; 2^{n+1} + 2. \tag{2}$$

Denote $x' = 1 - x$. Then by conditioning on the first digit,

$$E_\tau T_{\tau x} \;=\; 1 \;+\; \frac{1}{2}E_{\tau x'}T_{\tau x}. \tag{3}$$

But no suffix of $\tau x'$ is a prefix of τx, so using (2),

$$E_{\tau x'}T_{\tau x} \;=\; E\,T_{\tau x} \;=\; 2^{n+1} + 2.$$

Combining this with (3), we have $E_\tau T_{\tau x} = 2^n + 2$. Since τ must occur before τx can occur,

$$E\,T_{\tau x} = E\,T_\tau + E_\tau T_{\tau x} = E\,T_\tau + 2 + 2^n.$$

Thus, $2^{n+1} + 2 = E\,T_\tau + 2^n + 2$, and we conclude that $E\,T_\tau = 2^n$.

With the result of the theorem it is a simple matter to design a recursive algorithm to compute $E\,T_\tau$ for any given finite string τ. See, e.g., [6] for an implementation in the C programming language. Another approach is based upon the concept of "failure function" from the theory of lexical analysis. (See, e.g., [1 ,exercises 3.26 and 3.27], which in turn are based on a paper of Knuth, Morris and Pratt [9].) Consider the problem of designing an algorithm to find the first occurrence of τ in an input stream of binary digits. A common approach is to build (or simulate) a finite state machine whose states correspond to prefixes of τ. There is also an initial state, and a terminal state that is entered at the moment the full string τ is found in the input stream.

For example, suppose τ is 01001. In addition to the starting and terminal states, we would construct 5 additional states corresponding to the strings $0, 01, 010$, and 0100. Call these states 1,2,3, and 4, with the initial state being labeled 0 and the terminal state being labeled 5. When the current state is 3 (corresponding to 010,) and a 0 is received, the new state will be 4 (corresonding to 0100.) On the other hand, if a 1 is received the new state would be 2 (corresponding to 01.) In general, the current state measures the progress towards constructing the entire target string.

If the states are numbered as described, the *failure function, f(n),* is the length of the longest proper suffix of the string labeled by n that is also a prefix of the target string. (One defines $f(n) = 0$ if there is no such suffix.) For example, with the target string as above, we have $f(1) = 0, f(2) = 0, f(3) = 1$, and $f(4) = 1$. Knuth, Morris, and Pratt *op. cit.* show how to compute the failure function and use it to implement an efficient search algorithm.

By iterating the result of the theorem, one easily obtains the following expression for $E\,T_\tau$ in terms of the failure function:

$$E\,T_\tau \;=\; 2^n + 2^{f(n)} + 2^{f(f(n))} + \ldots,$$

with the sum being continued as long as the exponent remains positive.

3 Variations

A number of authors have considered variations on the problem discussed in the previous section. Blom and Thorburn [3], e.g, consider the first time any in a set of target strings is obtained in a stream of random data. The corresponding deterministic problem, i.e, implementing algorithms to search for keywords in a stream of data, is obviously important in computer science and there is an extensive literature. (Compilers of computer languages must be able to recognize programming language keywords such as while, if, for, etc.) See [1].

E. Karnin [8] considers the first time, τ_n, that *any* sequence of a given length n repeats in random data. Among other results, he finds the asymptotic behavior of the expected time:

$$E\,\tau_n \sim \sqrt{\pi 2^n}, \quad \text{as } n \to \infty.$$

Karnin points out that this result is closely related to the classical "birthday problem." Suppose there are N distinct objects that are sampled one by one with replacement. How long, on average, until some object is obtained for the second time? If the objects in question are binary strings of length n, then $N = 2^n$. When such strings are generated independently, unlike the case considered above, then the same asymptotic analysis as given in Karnin's paper shows that the expected time until the first repeat is asymptotic to $\sqrt{\pi 2^{n-1}}$ as $n \to \infty$. Thus, it takes roughly $\sqrt{2}$ times longer until the first repeat in a stream of random digits, apparently due to the dependence amongst successive n-tuples.

There are many interesting issues concerned with the probability that one string of a given length will occur before another given string of the same length. See, e.g, [5], which discusses curious non-transitive effects that occur when bettors place wagers on which strings will occur first.

How long does it take, on average, before *all* strings of a given length have been observed in the input stream? More generally, how long does it take a given Markov chain to visit all of its states? If we denote this random time by T, then it is not difficult to write down an explicit formula for $E\,T$ using some elementary facts about Markov chains. Assume the chain is irreducible and has a finite state space. It is easy to see that

$$E_i(\text{number of visits to j}) = \delta_{ij} + P_{ij} + P_{ij}^2 + \dots \tag{4}$$

Suppose A is a given collection of states, the initial state, i, does not belong to A, and we want to find E_i(number of visits to j before reaching A). The previous formula can be used if we replace all transitions into and out of states in A with transitions to a new absorbing state, Δ. Its transition matrix $\tilde{Q}$ can be constructed from P as follows: first construct a matrix Q from P by replacing all entries in rows and columns corresponding to states in A with zero entries. Next add a bottom row to Q consisting of all zeros, and finally adjoin a last column to Q so that the row sums equal 1. (Entries in the new last column thus correspond to the probabilities of transitions into A for the original chain, and transitions into Δ for the new chain.) If $j \notin A$ the right side of (4) with P replaced by $\tilde{Q}$ then gives the expected number of visits to j before reaching A. Moreover, because of the form of $\tilde{Q}$, we can use Q in place of $\tilde{Q}$. Since the chain is irreducible, states in A will eventually be reached with probability one. From this, it is easy to show that the series converges and equals $(I-Q)^{-1}_{ij}$. To summarize, we have that

$$E_i(\text{number of visits to j before reaching A}) = (I-Q)^{-1}_{ij}.$$

Now suppose that $A = \{i_1, \dots, i_k\}$, that T_A denotes the time to reach some state in A for the first time, and that T_j denotes the time to reach an individual state j for the first time. Then

$$T_A \;\; = \;\; T_{i_1} \wedge T_{i_2} \wedge \cdots \wedge T_{i_k},$$

where $a \wedge b$ denotes the smaller of numbers a and b. On the other hand, if the entire state space is $S = \{i_1, \ldots, i_n\}$, then

$$T \;=\; T_{i_1} \vee T_{i_2} \vee \cdots \vee T_{i_n},$$

where we use the standard notation $a \vee b$ for the larger of the numbers. Finally, a version of the "inclusion-exclusion formula" gives

$$T_{i_1} \vee \cdots \vee T_{i_n} = \sum_{j=1}^{n} T_j - \sum_{i<j} T_i \wedge T_j + \sum_{i<j<k} T_i \wedge T_j \wedge T_k - \cdots + (-1)^{n+1} T_{i_1} \wedge \cdots \wedge T_{i_n}.$$

Combining these results, and taking expectations, we arrive at the following general formula: if i is a particular state and T is the time it takes to visit all the states, then

$$E_i\, T \;=\; \sum_{j=1}^{n} (-1)^{j+1} \sum_{C=\{i_1,\ldots,i_j\}} \sum_{k \notin C} (I - P_C)^{-1}_{ik},$$

where the second sum is over distinct subsets C of size j of the state space, and P_C is the matrix obtained from P by replacing with 0 all entries in rows and columns corresponding to states in C.

To apply this to the problem of finding all binary strings of length n, note that after the first n digits the resulting string is uniformly distributed on the set S of all strings σ of length n. By conditioning on the string σ it follows that

$$E\, T \;=\; n \;+\; \frac{1}{2^n} \sum_{\sigma} E_\sigma\, T,$$

where the sum extends over all strings σ of length n. The above formula can then be applied to calculate each term on the right.

For small values of n these formulas can be used to compute the expected value exactly (See, e.g, the Maple worksheet [7].) For $n = 1$, we have $E\,T = 3$. For $n = 2, 3$, and 4 the values are, respectively, 9.5, $\frac{82959}{3640} \approx 22.79\ldots$, and

$$\frac{15196470103027446764838236318296131920851968094230950060807620630943693}{25918001389871207439459590474165228239254323748667152552605683561440 0},$$

which is approximately equal to 58.63287788. (We reproduce the exact value to discourage those who might look for a simple formula.)

For larger values of n the formula becomes impractical. For example, with $n = 5$ the state space has size 32, and an unsophisticated application of the above formula would involve inverting some 2^{32} large matrices.

Failing a practical method for calculating $E\,T$, one can still ask for the asymptotic behavior of this quantity as $n \to \infty$. This is closely related to the

classical "coupon collector's problem:" Suppose there are N objects, each of a different type, and a collector receives one item, selected at random with replacement, at each unit of time. How long, on average, until the collector will have obtained at least one of each type of object? It is well known that as $N \to \infty$, the expected time is asymptotic to $N \log(N)$. (See, e.g, [4, p. 39].)

We can easily use this result to find an upper bound for ET : Group the stream of random digits into blocks of length n, and view each such block as one in a sequence of random samples from the set of binary strings of length n. Since $N = 2^n$ here, we immediatly obtain the asymptotic upper bound that

$$ET \quad \leq \quad (\log(2) + \epsilon) n^2 2^n,$$

once n is sufficiently large, for any $\epsilon > 0$.

Of course, this result is quite crude. Indeed, Mori [10] has proved that the correct rate of growth is exactly the same as for the coupon collector's problem:

$$ET \quad \sim \log(2) n 2^n,$$

as $n \to \infty$. (I thank David Aldous for pointing out this reference.) It is perhaps surprising that the dependence amongst successive substrings of length n in a random stream does not affect the result of the coupon collector's problem, since it apparently has a significant effect in the case of the birthday problem.

References

1. A.V. Aho, R. Sethi, and J.D. Ullman, *Compilers, Principles, Techniques, and Tools,* Addison-Wesley, Reading, 1988.

2. G. Blom, On the mean number of random digits until a given sequence occurs, *J. Appl. Prob.* **19**(1982), 136-143.

3. G. Blom and D. Thorburn, How many random digits are required until given sequences are obtained? *J. Appl. Prob.* **19**(1982),518-531.

4. Richard Durrett, *Probability: Theory and Examples,* 2nd Edition, Wadsworth, Belmont, CA, 1996.

5. M. Gardner, Mathematical games, *Scientific American* **231**(1974), 120-125.

6. http://barnyard.syr.edu/quickies/cover.c

7. http://barnyard.syr.edu/cover.ma

8. E.D. Karnin, The first repetition of a pattern in a symmetric bernoulli sequence, *J. Appl. Prob.* **20** (1983),413-418.

9. D.E. Knuth, J.H. Morris, and V.R. Pratt, Fast pattern matching in strings, *SIAM J. Comput.* **6**(1977), 323-350.

10. T.F. Mori, On the expectation of the maximum waiting time, *Ann. Univ. Sci. Budapest. Sect. Comput.* **7**(1987), 111-115.

11. P.T. Nielsen, On the expected duration of a search for a fixed pattern in random data, *IEEE Trans. Inform. Theory* **19**(1973), 702-704.

12. Sheldon M. Ross, *Introduction to Probability Models,* 7th Edition, Academic Press, San Diego, 2000.