Hitting a prime in 2.43 dice rolls (on average)

Noga Alon
Department of Mathematics, Princeton University
Princeton, NJ 08544, USA
and
Schools of Mathematics and Computer Science
Tel Aviv University
Tel Aviv 6997801, Israel

Yaakov Malinovsky*
Department of Mathematics and Statistics
University of Maryland, Baltimore County
Baltimore, MD 21250, USA

April 4, 2023

Abstract

What is the number of rolls of fair 6-sided dice until the first time the total sum of all rolls is a prime? We compute the expectation and the variance of this random variable up to an additive error of less than 10^{-4}. This is a solution to a puzzle suggested by DasGupta (2017) in the Bulletin of the Institute of Mathematical Statistics, where the published solution is incomplete. The proof is simple, combining a basic dynamic programming algorithm with a quick Matlab computation and basic facts about the distribution of primes.

Keywords: dynamic-programming, prime number theorem, stopping time

*Corresponding author

1 The Problem and Monte-Carlo Simulation

The following puzzle appears in the Bulletin of the Institute of Mathematical Statistics (DasGupta, 2017): Let $X_1, X_2, \ldots$ be independent uniform random variables on the integers $1, 2, \ldots, 6$, and define $S_n = X_1 + \ldots + X_n$ for $n = 1, 2, \ldots$. Denote by τ the discrete time in which S_n first hits the set of prime numbers P:

$$\tau = \min \{n \geq 1 : S_n \in P\} .$$

The contributing editor (DasGupta, 2017) provides a lower bound of 2.34 for the expectation $E(\tau)$ and mentions the following heuristic approximation for it: $E(\tau) \approx 7.6$. He also adds that it is unknown whether τ has a finite variance.

In this note, we compute the value of $E(\tau)$ up to an additive error of less than 10^{-7}, showing it is much closer to the lower bound mentioned above than to 7.6. We also show the variance is finite and compute its value up to an additive error of less than 10^{-4}. It will be clear from the discussion that it is not difficult to get a better approximation for both quantities by increasing the amount of computation performed.

Before describing the rigorous argument, we present in Table 1 below the outcomes of Monte-Carlo simulations of the process.

Table 1: Monte-Carlo simulations

number of repetitions	$mean(\tau)$	$variance(\tau)$	$max(\tau)$
10^6	2.4316	6.2735	49
2×10^6	2.4274	6.2572	67
3×10^6	2.4305	6.2372	70
5×10^6	2.4287	6.2418	64
10^7	2.4286	6.2463	65

We provide Matlab code for the Monte-Carlo simulation in the supplementary materials, Section A.

In the next sections, we proceed with a rigorous computation of $E(\tau)$ and $Var(\tau)$ up to an additive error smaller than $1/10,000$. Not surprisingly, this computation shows that the simulations supply accurate values.

2 Expectation and Variance of the Hitting Time

First, we present the formulas for calculating the expectation and variance of the hitting time τ as a function of the probability that τ equals or exceeds a certain value k, for $k = 1, 2, 3, \ldots$, which we denote by $p(k) = P(\tau \geq k)$. We have

$$E(\tau) = \sum_{k \geq 1} p(k) \tag{1}$$

and

$$E(\tau^2) = \sum_{k \geq 1} (2k-1)p(k). \tag{2}$$

We remind in the supplementary materials, Section B how to obtain the formulas (1) and (2).

Obviously, by the definition of variance we have

$$Var(\tau) = E(\tau^2) - [E(\tau)]^2. \tag{3}$$

In section 3 we develop a dynamic programming algorithm to compute $p(k)$ exactly and use the first 1000 values ($k = 1, \ldots, 1000$) to estimate $E(\tau)$ and $Var(\tau)$ with reference to expressions (1), (2), and (3).

3 Dynamic Programming Algorithm and Estimates

In this section we develop a dynamic programming algorithm to compute the first K values $p(1), p(2), \ldots, p(K)$, then use these values to estimate $E(\tau)$ and $Var(\tau)$.

3.1 Dynamic Programming Algorithm

For each integer $k \geq 1$ and for each non-prime n satisfying $k \leq n \leq 6k$, let $p(k, n)$ denote the probability that $X_1 + \cdots + X_k = n$ and that for every $i < k$, $X_1 + \cdots + X_i$ is non-prime. Fix a parameter K (in our computation, we later take $K = 1000$). By the definition of $p(k, n)$ and the rule of total probability, we have the following dynamic programming (DP) algorithm for computing $p(k, n)$ precisely for all $1 \leq k \leq K$ and $k \leq n \leq 6k$:

1. $p(1, 1) = p(1, 4) = p(1, 6) = 1/6$.

2. For $k = 2, \ldots, K$ and for any non-prime n between k and $6k$,

$$p(k, n) = \frac{1}{6} \sum_{i} p(k-1, n-i), \tag{4}$$

where the sum ranges over all i between 1 and 6 so that $n - i$ is non-prime.
From the definitions of $p(k)$ and $p(k, n)$, we obtain the following identity:

$$p(k+1) = \sum_{\{n:\ k \leq n \leq 6k\}} p(k, n). \tag{5}$$

We apply this DP algorithm to get the values $p(k, n)$ for $k = 1, \ldots, K$. Then, using identity (5), we obtain the values $p(1), p(2), \ldots, p(K)$, which consequently provide us the partial sums in (1) and (2). These partial sums are the lower bounds to $E(\tau)$ and $E(\tau^2)$, which allows us to estimate the expectation and variance of the hitting time, which we discuss in the next section.

3.2 Estimators of the Expectation and Variance of Hitting Time

Denote by E_K and $E_K^{(2)}$ the estimators (lower bounds) of $E(\tau)$ and $E(\tau^2)$ based on the respective values of $p(k)$ for the first K values in (1) and (2):

$$E_K = \sum_{k=1}^{K} p(k) \quad \text{and} \quad E_K^{(2)} = \sum_{k=1}^{K} (2k-1) p(k). \tag{6}$$

We estimate $E(\tau)$ by E_K and consequently $Var(\tau)$ by V_K, defined as follows

$$V_K = E_K^{(2)} - (E_K)^2.$$

Setting $K = 1000$ and applying the dynamic programming algorithm in Matlab (provided in the supplementary materials, Section C), with an execution time of less than five seconds, we obtain $E_{1000} = 2.4284$ and $V_{1000} = 6.2427$.

The "quality" of these estimators can be measured as the difference between $E(\tau)$ and $Var(\tau)$ and their corresponding estimators:

$$RE_K = E(\tau) - E_K \quad \text{and} \quad RV_K = Var(\tau) - V_K. \tag{7}$$

In the next section, we bound RE_K and RV_K and show that for $K = 1000$, $RE_{1000} < 10^{-7}$ and $RV_{1000} < 10^{-4}$.

4 Bounding the Remainders

In this section, we provide the bounds for the remainder terms defined in (7). To accomplish this, we will use basic facts about the distribution of primes and prove the following simple result by induction on k.

Proposition 1. *For every k and for every non-prime n,*

$$p(k,n) < \frac{1}{3}\left(\frac{5}{6}\right)^{\pi(n)}, \tag{8}$$

where $\pi(n)$ is the number of primes smaller than n.

Proof. Note first that (8) holds for $k = 1$, as $1/6 = p(1,6) < (1/3)(5/6)^3$, $1/6 = p(1,4) < (1/3)(5/6)^2$ and $1/6 = p(1,1) < (1/3)(5/6)^0$, with room to spare. Assuming the inequality holds for $k-1$ (and every relevant n) we prove it for k. Suppose there are q primes in the set $\{n-6,\ldots,n-1\}$, then $\pi(n-i) \geq \pi(n) - q$ for all non-prime $n-i$ in this set. Thus, by the induction hypothesis, and using (4), we obtain

$$p(k,n) \leq \frac{1}{6}(6-q)\frac{1}{3}\left(\frac{5}{6}\right)^{\pi(n)-q} \leq \left(\frac{5}{6}\right)^{q}\frac{1}{3}\left(\frac{5}{6}\right)^{\pi(n)-q} = \frac{1}{3}\left(\frac{5}{6}\right)^{\pi(n)}.$$

□

By the prime number theorem (cf., e.g., Hardy and Wright (2008)), for every $n > 1000$ $\pi(n) > 0.9\dfrac{n}{\ln n}$ (again, with room to spare). Therefore, from the above estimate, we arrive at the following result.

Corollary 1. *For every $k > 1000$ and every non-prime $n (n \geq k)$,*

$$p(k,n) < \frac{1}{3}\left(\frac{5}{6}\right)^{0.9\frac{n}{\ln n}}.$$

Corollary 1 is the crucial result in obtaining the upper bounds of the remainders RE_{1000} and RV_{1000}, which are given in the following proposition.

Proposition 2. *The remainder terms, which are defined in* (7)*, are bounded as follows:*

$$(a)\ RE_{1000} < 10^{-7},$$

$$(b)\ RV_{1000} < 10^{-4}.$$

Proof. Recall that

$$P(\tau \geq k+1) = p(k+1) = \sum_{\{n:\ k \leq n \leq 6k\}} p(k,n).$$

For part (a), we have

$$RE_{1000} = \sum_{k>1000} P(\tau \geq k) = \sum_{k>999} P(\tau \geq k+1) = \sum_{k>999} \sum_{\{n:\ k \leq n \leq 6k\}} p(k,n)$$

$$< \sum_{k>999} \sum_{\{n:\ k \leq n \leq 6k\}} \frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n} = \sum_{n \geq 1000} \sum_{k=\max(1000, n/6)}^{n} \frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n} < \sum_{n \geq 1000} \sum_{k=1000}^{n} \frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n}$$

$$= \sum_{n \geq 1000} (n-999)\frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n}, \tag{9}$$

where the first inequality is obtained from Corollary 1.

Define

$$f(n) = (n-999)\frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n}, \tag{10}$$

where n is an integer ≥ 1000. The first part of Proposition 2 follows by noting $\sum_{n \geq 1000} f(n) < 10^{-7}$, shown in the supplementary materials, Section D.

For part (b),

$$R^{(2)}_{1000} := \sum_{k>1000} (2k-1)P(\tau \geq k) = \sum_{k>1000} \sum_{\{n:\ k-1 \leq n \leq 6(k-1)\}} (2k-1)p(k-1,n)$$

$$< \sum_{k>1000} \sum_{\{n:\ k-1 \leq n \leq 6(k-1)\}} (2k-1)\left(\frac{5}{6}\right)^{0.9n/\ln n} = \sum_{n \geq 1000} \frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n} \sum_{k=\max(1001, n/6+1)}^{n+1} (2k-1)$$

$$< \sum_{n \geq 1000} \frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n} \sum_{k=1001}^{n+1} (2k-1) = \sum_{n \geq 1000} \frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n} \left[(n+1)^2 - 1000^2\right], \tag{11}$$

where the first inequality is also obtained from Corollary 1.

Denote by

$$g(n) = \left[(n+1)^2 - 1000^2\right]\frac{1}{3}\left(\frac{5}{6}\right)^{0.9n/\ln n}, \tag{12}$$

where n is an integer ≥ 1000. Likewise, the second part of Proposition 2 follows by noting $\sum_{n \geq 10{,}000} g(n) < 3.4 \times 10^{-68}$, shown in the supplementary materials, Section E.

Combining this with (11), we obtain

$$R^{(2)}_{1000} < \sum_{n=1000}^{9999} g(n) + \sum_{n\geq 10{,}000} g(n) < 8.5 \times 10^{-5} + 3.4 \times 10^{-68} < 1/10,000. \qquad (13)$$

Now, from (6) and (7) it follows that $RV_K = R^{(2)}_K - 2E_K(RE_K) - (RE_K)^2$, where $R^{(2)}_K := \sum_{k>K}(2k-1)p(k)$. Combining this with (13) and part (a) of Proposition 2, we conclude that $RV_{1000} < 10^{-4}$, i.e., the error of the variance estimation based on the first 1000 values of k is below $1/10,000$. □

5 Final Remarks

The problem considered in this work deals with a random process for generating primes, and its investigation ties together a simple algorithmic idea with basic facts about the distribution of prime numbers. Although the solution is tailored to this specific situation, it offers a method for studying problems of this type. This note can also be used as a motivator in upper-level undergraduate or graduate classes by introducing and illustrating the power of combining a simple dynamic programming algorithm with a quick computer-aided computation and basic facts about the distribution of primes.

Supplementary Materials

Section A Matlab Code for the Monte-Carlo Simulation.

Section B The First Two Moments of the Hitting Time.

Section C Matlab Code for the Dynamic-Programming Algorithm.

Section D Upper Bound of $\sum_{n\geq 1000} f(n)$.

Section E Upper Bound of $\sum_{n\geq 10{,}000} g(n)$.

Acknowledgments

We wish to thank the Editor, the Associate Editor and the two referees for helpful comments and suggestions.

Funding

Research of Noga Alon is supported in part by NSF grant DMS-2154082 and by BSF grant 2018267. Research of Yaakov Malinovsky is supported in part by BSF grant 2020063.

References

DasGupta, A. (2017). Solution to Puzzle 17. IMS Bulletin **46**(5), p.9. https://imstat.org/2017/07/14/student-puzzle-corner-18-and-solution-to-puzzle-17/

Hardy, G. H. and Wright, E. M. (2008). An Introduction to The Theory of Numbers. Sixth edition. Revised by D. R. Heath-Brown and J. H. Silverman, with a foreword by Andrew Wiles. Oxford University Press, Oxford.