Probability and Computing

Randomization and Probabilistic Techniques in Algorithms and Data Analysis

Second Edition

Michael Mitzenmacher Eli Upfal

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

4843/24, 2nd Floor, Ansari Road, Daryaganj, Delhi - 110002, India

79 Anson Road, #06-04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107154889
10.1017/9781316651124

First published 2017

Printed in the United Kingdom by Clays, St Ives plc

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging in Publication Data
Names: Mitzenmacher, Michael, 1969– author. | Upfal, Eli, 1954– author.
Title: Probability and computing / Michael Mitzenmacher Eli Upfal.
Description: Second edition. | Cambridge, United Kingdom ;
New York, NY, USA : Cambridge University Press, [2017] |
Includes bibliographical references and index.
Identifiers: LCCN 2016041654 | ISBN 9781107154889
Subjects: LCSH: Algorithms. | Probabilities. | Stochastic analysis.
Classification: LCC QA274.M574 2017 | DDC 518/.1 – dc23
LC record available at https://lccn.loc.gov/2016041654

ISBN 978-1-107-15488-9 Hardback

Additional resources for this publication at www.cambridge.org/Mitzenmacher.

CHAPTER TWO

Discrete Random Variables and Expectation

In this chapter, we introduce the concepts of discrete random variables and expectation and then develop basic techniques for analyzing the expected performance of algorithms. We apply these techniques to computing the expected running time of the well-known Quicksort algorithm. In analyzing two versions of Quicksort, we demonstrate the distinction between the analysis of randomized algorithms, where the probability space is defined by the random choices made by the algorithm, and the *probabilistic analysis* of deterministic algorithms, where the probability space is defined by some probability distribution on the inputs.

Along the way we define the Bernoulli, binomial, and geometric random variables, study the expected size of a simple branching process, and analyze the expectation of the coupon collector's problem – a probabilistic paradigm that reappears throughout the book.

2.1. Random Variables and Expectation

When studying a random event, we are often interested in some value associated with the random event rather than in the event itself. For example, in tossing two dice we are often interested in the sum of the two dice rather than the separate value of each die. The sample space in tossing two dice consists of 36 events of equal probability, given by the ordered pairs of numbers $\{(1,1),(1,2),\ldots,(6,5),(6,6)\}$. If the quantity we are interested in is the sum of the two dice, then we are interested in 11 events (of unequal probability): the 11 possible outcomes of the sum. Any such function from the sample space to the real numbers is called a random variable.

Definition 2.1: *A random variable X on a sample space Ω is a real-valued (measurable) function on Ω; that is, $X : \Omega \to \mathbb{R}$. A discrete random variable is a random variable that takes on only a finite or countably infinite number of values.*

Since random variables are functions, they are usually denoted by a capital letter such as X or Y, while real numbers are usually denoted by lowercase letters.

For a discrete random variable X and a real value a, the event "$X = a$" includes all the basic events of the sample space in which the random variable X assumes the value a. That is, "$X = a$" represents the set $\{s \in \Omega \mid X(s) = a\}$. We denote the probability of that event by

$$\Pr(X = a) = \sum_{s \in \Omega : X(s) = a} \Pr(s).$$

If X is the random variable representing the sum of the two dice, then the event $X = 4$ corresponds to the set of basic events $\{(1, 3), (2, 2), (3, 1)\}$. Hence

$$\Pr(X = 4) = \frac{3}{36} = \frac{1}{12}.$$

The definition of independence that we developed for events extends to random variables.

Definition 2.2: *Two random variables X and Y are independent if and only if*

$$\Pr((X = x) \cap (Y = y)) = \Pr(X = x) \cdot \Pr(Y = y)$$

for all values x and y. Similarly, random variables $X_1, X_2, \ldots, X_k$ are mutually independent if and only if, for any subset $I \subseteq [1, k]$ and any values x_i, $i \in I$,

$$\Pr\left(\bigcap_{i \in I}(X_i = x_i)\right) = \prod_{i \in I} \Pr(X_i = x_i).$$

A basic characteristic of a random variable is its *expectation*, which is also often called the *mean*. The expectation of a random variable is a weighted average of the values it assumes, where each value is weighted by the probability that the variable assumes that value.

Definition 2.3: *The expectation of a discrete random variable X, denoted by* $\mathbf{E}[X]$, *is given by*

$$\mathbf{E}[X] = \sum_i i \Pr(X = i),$$

where the summation is over all values in the range of X. The expectation is finite if $\sum_i |i| \Pr(X = i)$ converges; otherwise, the expectation is unbounded.

For example, the expectation of the random variable X representing the sum of two dice is

$$\mathbf{E}[X] = \frac{1}{36} \cdot 2 + \frac{2}{36} \cdot 3 + \frac{3}{36} \cdot 4 + \cdots + \frac{1}{36} \cdot 12 = 7.$$

You may try using symmetry to give simpler argument for why $\mathbf{E}[X] = 7$.

As an example of where the expectation of a discrete random variable is unbounded, consider a random variable X that takes on the value 2^i with probability $1/2^i$ for $i = 1, 2, \ldots$. The expected value of X is

$$\mathbf{E}[X] = \sum_{i=1}^{\infty} \frac{1}{2^i} 2^i = \sum_{i=1}^{\infty} 1 = \infty.$$

Here we use the somewhat informal notation $\mathbf{E}[X] = \infty$ to express that $\mathbf{E}[X]$ is unbounded.

2.1.1. *Linearity of Expectations*

A key property of expectation that significantly simplifies its computation is the *linearity of expectations*. By this property, the expectation of the sum of random variables is equal to the sum of their expectations. Formally, we have the following theorem.

Theorem 2.1 [Linearity of Expectations]: *For any finite collection of discrete random variables $X_1, X_2, \ldots, X_n$ with finite expectations,*

$$\mathbf{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbf{E}[X_i].$$

Proof: We prove the statement for two random variables X and Y; the general case follows by induction. The summations that follow are understood to be over the ranges of the corresponding random variables:

$$\begin{aligned}
\mathbf{E}[X+Y] &= \sum_i \sum_j (i+j)\Pr((X=i)\cap(Y=j)) \\
&= \sum_i \sum_j i\Pr((X=i)\cap(Y=j)) + \sum_i \sum_j j\Pr((X=i)\cap(Y=j)) \\
&= \sum_i i \sum_j \Pr((X=i)\cap(Y=j)) + \sum_j j \sum_i \Pr((X=i)\cap(Y=j)) \\
&= \sum_i i\Pr(X=i) + \sum_j j\Pr(Y=j) \\
&= \mathbf{E}[X] + \mathbf{E}[Y].
\end{aligned}$$

The first equality follows from Definition 1.2. In the penultimate equation we have used Theorem 1.6, the Law of Total Probability. ■

We now use this property to compute the expected sum of two standard dice. Let $X = X_1 + X_2$, where X_i represents the outcome of die i for $i = 1, 2$. Then

$$\mathbf{E}[X_i] = \frac{1}{6}\sum_{j=1}^{6} j = \frac{7}{2}.$$

Applying the linearity of expectations, we have

$$\mathbf{E}[X] = \mathbf{E}[X_1] + \mathbf{E}[X_2] = 7.$$

It is worth emphasizing that linearity of expectations holds for any collection of random variables, even if they are *not* independent! For example, consider again the previous example and let the random variable $Y = X_1 + X_1^2$. We have

$$\mathbf{E}[Y] = \mathbf{E}[X_1 + X_1^2] = \mathbf{E}[X_1] + \mathbf{E}[X_1^2],$$

even though X_1 and X_1^2 are clearly dependent. As an exercise, you may verify this identity by considering the six possible outcomes for X_1.

Linearity of expectations also holds for countably infinite summations in certain cases. Specifically, it can be shown that

$$\mathbf{E}\left[\sum_{i=1}^{\infty} X_i\right] = \sum_{i=1}^{\infty} \mathbf{E}[X_i]$$

whenever $\sum_{i=1}^{\infty} \mathbf{E}[|X_i|]$ converges. The issue of dealing with the linearity of expectations with countably infinite summations is further considered in Exercise 2.29.

This chapter contains several examples in which the linearity of expectations significantly simplifies the computation of expectations. One result related to the linearity of expectations is the following simple lemma.

Lemma 2.2: *For any constant c and discrete random variable X,*

$$\mathbf{E}[cX] = c\mathbf{E}[X].$$

Proof: The lemma is obvious for $c = 0$. For $c \neq 0$,

$$\begin{aligned}
\mathbf{E}[cX] &= \sum_j j \Pr(cX = j) \\
&= c \sum_j (j/c) \Pr(X = j/c) \\
&= c \sum_k k \Pr(X = k) \\
&= c\mathbf{E}[X].
\end{aligned}$$

■

2.1.2. *Jensen's Inequality*

Suppose that we choose the length X of a side of a square uniformly at random from the range $[1, 99]$. What is the expected value of the area? We can write this as $\mathbf{E}[X^2]$. It is tempting to think of this as being equal to $\mathbf{E}[X]^2$, but a simple calculation shows that this is not correct. In fact, $\mathbf{E}[X]^2 = 2500$ whereas $\mathbf{E}[X^2] = 9950/3 > 2500$.

More generally, we can prove that $\mathbf{E}[X^2] \geq (\mathbf{E}[X])^2$. Consider $Y = (X - \mathbf{E}[X])^2$. The random variable Y is nonnegative and hence its expectation must also be nonnegative. Therefore,

$$\begin{aligned}
0 \leq \mathbf{E}[Y] &= \mathbf{E}[(X - \mathbf{E}[X])^2] \\
&= \mathbf{E}[X^2 - 2X\mathbf{E}[X] + (\mathbf{E}[X])^2] \\
&= \mathbf{E}[X^2] - 2\mathbf{E}[X\mathbf{E}[X]] + (\mathbf{E}[X])^2 \\
&= \mathbf{E}[X^2] - (\mathbf{E}[X])^2.
\end{aligned}$$

To obtain the penultimate line, we used the linearity of expectations. To obtain the last line we used Lemma 2.2 to simplify $\mathbf{E}[X\mathbf{E}[X]] = \mathbf{E}[X] \cdot \mathbf{E}[X]$.

The fact that $\mathbf{E}[X^2] \geq (\mathbf{E}[X])^2$ is an example of a more general theorem known as Jensen's inequality. Jensen's inequality shows that, for any convex function f, we have $\mathbf{E}[f(X)] \geq f(\mathbf{E}[X])$.

Definition 2.4: *A function $f : \mathbf{R} \to \mathbf{R}$ is said to be convex if, for any x_1, x_2 and $0 \leq \lambda \leq 1$,*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2).$$

Visually, a convex function f has the property that, if you connect two points on the graph of the function by a straight line, this line lies on or above the graph of the function. The following fact, which we state without proof, is often a useful alternative to Definition 2.4.

Lemma 2.3: *If f is a twice differentiable function, then f is convex if and only if $f''(x) \geq 0$.*

Theorem 2.4 [Jensen's Inequality]: *If f is a convex function, then*

$$\mathbf{E}[f(X)] \geq f(\mathbf{E}[X]).$$

Proof: We prove the theorem assuming that f has a Taylor expansion. Let $\mu = \mathbf{E}[X]$. By Taylor's theorem there is a value c such that

$$\begin{aligned} f(x) &= f(\mu) + f'(\mu)(x-\mu) + \frac{f''(c)(x-\mu)^2}{2} \\ &\geq f(\mu) + f'(\mu)(x-\mu), \end{aligned}$$

since $f''(c) > 0$ by convexity. Taking expectations of both sides and applying linearity of expectations and Lemma 2.2 yields the result:

$$\begin{aligned} \mathbf{E}[f(X)] &\geq \mathbf{E}[f(\mu) + f'(\mu)(X-\mu)] \\ &= \mathbf{E}[f(\mu)] + f'(\mu)(\mathbf{E}[X] - \mu) \\ &= f(\mu) = f(\mathbf{E}[X]). \end{aligned}$$ ■

An alternative proof of Jensen's inequality, which holds for any random variable X that takes on only finitely many values, is presented in Exercise 2.10.

2.2. The Bernoulli and Binomial Random Variables

Suppose that we run an experiment that succeeds with probability p and fails with probability $1 - p$.

Let Y be a random variable such that

$$Y = \begin{cases} 1 & \text{if the experiment succeeds,} \\ 0 & \text{otherwise.} \end{cases}$$

The variable Y is called a *Bernoulli* or an *indicator* random variable. Note that, for a Bernoulli random variable,

$$\mathbf{E}[Y] = p \cdot 1 + (1 - p) \cdot 0 = p = \Pr(Y = 1).$$

For example, if we flip a fair coin and consider the outcome "heads" a success, then the expected value of the corresponding indicator random variable is $1/2$.

Consider now a sequence of n independent coin flips. What is the distribution of the number of heads in the entire sequence? More generally, consider a sequence of n independent experiments, each of which succeeds with probability p. If we let X represent the number of successes in the n experiments, then X has a *binomial distribution*.

Definition 2.5: *A* binomial random variable X *with parameters n and p, denoted by $B(n, p)$, is defined by the following probability distribution on $j = 0, 1, 2, \ldots, n$:*

$$\Pr(X = j) = \binom{n}{j} p^j (1 - p)^{n-j}.$$

That is, the binomial random variable X equals j when there are exactly j successes and $n - j$ failures in n independent experiments, each of which is successful with probability p.

As an exercise, you should show that Definition 2.5 ensures that $\sum_{j=0}^{n} \Pr(X = j) = 1$. This is necessary for the binomial random variable to have a valid probability function, according to Definition 1.2.

The binomial random variable arises in many contexts, especially in sampling. As a practical example, suppose that we want to gather data about the packets going through a router by postprocessing them. We might want to know the approximate fraction of packets from a certain source or of a certain data type. We do not have the memory available to store all of the packets, so we choose to store a random subset – or *sample* – of the packets for later analysis. If each packet is stored with probability p and if n packets go through the router each day, then the number of sampled packets each day is a binomial random variable X with parameters n and p. If we want to know how much memory is necessary for such a sample, a natural starting point is to determine the expectation of the random variable X.

Sampling in this manner arises in other contexts as well. For example, by sampling the program counter while a program runs, one can determine what parts of a program are taking the most time. This knowledge can be used to aid dynamic program optimization techniques such as binary rewriting, where the executable binary form of a program is modified while the program executes. Since rewriting the executable as the program runs is expensive, sampling helps the optimizer to determine when it will be worthwhile.

What is the expectation of a binomial random variable X? We can compute it directly from the definition as

$$\begin{aligned}
\mathbf{E}[X] &= \sum_{j=0}^{n} j \binom{n}{j} p^j (1-p)^{n-j} \\
&= \sum_{j=0}^{n} j \frac{n!}{j!\,(n-j)!} p^j (1-p)^{n-j} \\
&= \sum_{j=1}^{n} \frac{n!}{(j-1)!\,(n-j)!} p^j (1-p)^{n-j} \\
&= np \sum_{j=1}^{n} \frac{(n-1)!}{(j-1)!\,((n-1)-(j-1))!} p^{j-1} (1-p)^{(n-1)-(j-1)} \\
&= np \sum_{k=0}^{n-1} \frac{(n-1)!}{k!\,((n-1)-k)!} p^k (1-p)^{(n-1)-k} \\
&= np \sum_{k=0}^{n-1} \binom{n-1}{k} p^k (1-p)^{(n-1)-k} \\
&= np,
\end{aligned}$$

where the last equation uses the binomial identity

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}.$$

The linearity of expectations allows for a significantly simpler argument. If X is a binomial random variable with parameters n and p, then X is the number of successes in n trials, where each trial is successful with probability p. Define a set of n indicator random variables $X_1, \dots, X_n$, where $X_i = 1$ if the ith trial is successful and 0 otherwise. Clearly, $\mathbf{E}[X_i] = p$ and $X = \sum_{i=1}^{n} X_i$ and so, by the linearity of expectations,

$$\mathbf{E}[X] = \mathbf{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbf{E}[X_i] = np.$$

The linearity of expectations makes this approach of representing a random variable by a sum of simpler random variables, such as indicator random variables, extremely useful.

2.3. Conditional Expectation

Just as we have defined conditional probability, it is useful to define the *conditional expectation* of a random variable. The following definition is quite natural.

Definition 2.6: $$\mathbf{E}[Y \mid Z = z] = \sum_y y \Pr(Y = y \mid Z = z),$$

where the summation is over all y in the range of Y.

The definition states that the conditional expectation of a random variable is, like the expectation, a weighted sum of the values it assumes. The difference is that now each value is weighted by the *conditional probability* that the variable assumes that value. One can similarly define the conditional expectation of a random variable Y conditioned on an event $\mathcal{E}$ as

$$\mathbf{E}[Y \mid \mathcal{E}] = \sum_y y \Pr(Y = y \mid \mathcal{E}).$$

For example, suppose that we independently roll two standard six-sided dice. Let X_1 be the number that shows on the first die, X_2 the number on the second die, and X the sum of the numbers on the two dice. Then

$$\mathbf{E}[X \mid X_1 = 2] = \sum_x x \Pr(X = x \mid X_1 = 2) = \sum_{x=3}^{8} x \cdot \frac{1}{6} = \frac{11}{2}.$$

As another example, consider $\mathbf{E}[X_1 \mid X = 5]$:

$$\begin{aligned}
\mathbf{E}[X_1 \mid X = 5] &= \sum_{x=1}^{4} x \Pr(X_1 = x \mid X = 5) \\
&= \sum_{x=1}^{4} x \frac{\Pr(X_1 = x \cap X = 5)}{\Pr(X = 5)} = \sum_{x=1}^{4} x \frac{1/36}{4/36} = \frac{5}{2}.
\end{aligned}$$

The following natural identity follows from Definition 2.6.

Lemma 2.5: *For any random variables X and Y,*

$$\mathbf{E}[X] = \sum_y \Pr(Y = y)\mathbf{E}[X \mid Y = y],$$

where the sum is over all values in the range of Y and all of the expectations exist.

Proof:
$$\begin{aligned}
\sum_y \Pr(Y = y)\mathbf{E}[X \mid Y = y] &= \sum_y \Pr(Y = y) \sum_x x \Pr(X = x \mid Y = y) \\
&= \sum_x \sum_y x \Pr(X = x \mid Y = y) \Pr(Y = y) \\
&= \sum_x \sum_y x \Pr(X = x \cap Y = y) \\
&= \sum_x x \Pr(X = x) = \mathbf{E}[X].
\end{aligned}$$
■

The linearity of expectations also extends to conditional expectations. This is clarified in Lemma 2.6, whose proof is left as Exercise 2.11.

Lemma 2.6: *For any finite collection of discrete random variables $X_1, X_2, \ldots, X_n$ with finite expectations and for any random variable Y,*

$$\mathbf{E}\left[\sum_{i=1}^{n} X_i \mid Y = y\right] = \sum_{i=1}^{n} \mathbf{E}[X_i \mid Y = y].$$

Perhaps somewhat confusingly, the conditional expectation is also used to refer to the following random variable.

Definition 2.7: *The expression* $\mathbf{E}[Y \mid Z]$ *is a random variable* $f(Z)$ *that takes on the value* $\mathbf{E}[Y \mid Z = z]$ *when* $Z = z$.

We emphasize that $\mathbf{E}[Y \mid Z]$ is *not* a real value; it is actually a function of the random variable Z. Hence $\mathbf{E}[Y \mid Z]$ is itself a function from the sample space to the real numbers and can therefore be thought of as a random variable.

In the previous example of rolling two dice,

$$\mathbf{E}[X \mid X_1] = \sum_{x} x \Pr(X = x \mid X_1) = \sum_{x=X_1+1}^{X_1+6} x \cdot \frac{1}{6} = X_1 + \frac{7}{2}.$$

We see that $\mathbf{E}[X \mid X_1]$ is a random variable whose value depends on X_1.

If $\mathbf{E}[Y \mid Z]$ is a random variable, then it makes sense to consider its expectation $\mathbf{E}[\mathbf{E}[Y \mid Z]]$. In our example, we found that $\mathbf{E}[X \mid X_1] = X_1 + 7/2$. Thus

$$\mathbf{E}[\mathbf{E}[X \mid X_1]] = \mathbf{E}\left[X_1 + \frac{7}{2}\right] = \frac{7}{2} + \frac{7}{2} = 7 = \mathbf{E}[X].$$

More generally, we have the following theorem.

Theorem 2.7:

$$\mathbf{E}[Y] = \mathbf{E}[\mathbf{E}[Y \mid Z]].$$

Proof: From Definition 2.7 we have $\mathbf{E}[Y \mid Z] = f(Z)$, where $f(Z)$ takes on the value $\mathbf{E}[Y \mid Z = z]$ when $Z = z$. Hence

$$\mathbf{E}[\mathbf{E}[Y \mid Z]] = \sum_{z} \mathbf{E}[Y \mid Z = z] \Pr(Z = z).$$

The right-hand side equals $\mathbf{E}[Y]$ by Lemma 2.5. ■

We now demonstrate an interesting application of conditional expectations. Consider a program that includes one call to a process $\mathcal{S}$. Assume that each call to process $\mathcal{S}$ recursively spawns new copies of the process $\mathcal{S}$, where the number of new copies is a binomial random variable with parameters n and p. We assume that these random variables are independent for each call to $\mathcal{S}$. What is the expected number of copies of the process $\mathcal{S}$ generated by the program?

To analyze this recursive spawning process, we introduce the idea of *generations*. The initial process $\mathcal{S}$ is in generation 0. Otherwise, we say that a process $\mathcal{S}$ is in generation i if it was spawned by another process $\mathcal{S}$ in generation $i - 1$. Let Y_i denote the number of $\mathcal{S}$ processes in generation i. Since we know that $Y_0 = 1$, the number of processes in generation 1 has a binomial distribution. Thus,

$$\mathbf{E}[Y_1] = np.$$

Similarly, suppose we knew that the number of processes in generation $i-1$ was y_{i-1}, so $Y_{i-1} = y_{i-1}$. Let Z_k be the number of copies spawned by the kth process spawned in the $(i-1)$th generation for $1 \le k \le y_{i-1}$. Each Z_k is a binomial random variable with parameters n and p. Then

$$\begin{aligned}
\mathbf{E}[Y_i \mid Y_{i-1} = y_{i-1}] &= \mathbf{E}\left[\sum_{k=1}^{y_{i-1}} Z_k \mid Y_{i-1} = y_{i-1}\right] \\
&= \sum_{j \ge 0} j \Pr\left(\sum_{k=1}^{y_{i-1}} Z_k = j \mid Y_{i-1} = y_{i-1}\right) \\
&= \sum_{j \ge 0} j \Pr\left(\sum_{k=1}^{y_{i-1}} Z_k = j\right) \\
&= \mathbf{E}\left[\sum_{k=1}^{y_{i-1}} Z_k\right] \\
&= \sum_{k=1}^{y_{i-1}} \mathbf{E}[Z_k] \\
&= y_{i-1} np.
\end{aligned}$$

In the third line we have used that the Z_k are all independent binomial random variables; in particular, the value of each Z_k is independent of Y_{i-1}, allowing us to remove the conditioning. In the fifth line, we have applied the linearity of expectations.

Applying Theorem 2.7, we can compute the expected size of the ith generation inductively. We have

$$\mathbf{E}[Y_i] = \mathbf{E}[\mathbf{E}[Y_i \mid Y_{i-1}]] = \mathbf{E}[Y_{i-1} np] = np\mathbf{E}[Y_{i-1}].$$

By induction on i, and using the fact that $Y_0 = 1$, we then obtain

$$\mathbf{E}[Y_i] = (np)^i.$$

The expected total number of copies of process $\mathcal{S}$ generated by the program is given by

$$\mathbf{E}\left[\sum_{i \ge 0} Y_i\right] = \sum_{i \ge 0} \mathbf{E}[Y_i] = \sum_{i \ge 0} (np)^i.$$

If $np \ge 1$ then the expectation is unbounded; if $np < 1$, the expectation is $1/(1-np)$. Thus, the expected number of processes generated by the program is bounded if and only if the expected number of processes spawned by each process is less than 1.

The process analyzed here is a simple example of a *branching process*, a probabilistic paradigm extensively studied in probability theory.

2.4. The Geometric Distribution

Suppose that we flip a coin until it lands on heads. What is the distribution of the number of flips? This is an example of a *geometric distribution*, which arises in the following situation: we perform a sequence of independent trials until the first success, where each trial succeeds with probability p.

Definition 2.8: *A* geometric *random variable X with parameter p is given by the following probability distribution on $n = 1, 2, \ldots,$:*

$$\Pr(X = n) = (1 - p)^{n-1} p.$$

That is, for the geometric random variable X to equal n, there must be $n - 1$ failures, followed by a success.

As an exercise, you should show that the geometric random variable satisfies

$$\sum_{n \geq 1} \Pr(X = n) = 1.$$

Again, this is necessary for the geometric random variable to have a valid probability function, according to Definition 1.2.

In the context of our example from Section 2.2 of sampling packets on a router, if packets are sampled with probability p, then the number of packets transmitted after the last sampled packet until and including the next sampled packet is given by a geometric random variable with parameter p.

Geometric random variables are said to be *memoryless* because the probability that you will reach your first success n trials from now is independent of the number of failures you have experienced. Informally, one can ignore past failures because they do not change the distribution of the number of future trials until first success. Formally, we have the following statement.

Lemma 2.8: *For a geometric random variable X with parameter p and for $n > 0$,*

$$\Pr(X = n + k \mid X > k) = \Pr(X = n).$$

Proof:

$$\begin{aligned}
\Pr(X = n + k \mid X > k) &= \frac{\Pr((X = n + k) \cap (X > k))}{\Pr(X > k)} \\
&= \frac{\Pr(X = n + k)}{\Pr(X > k)} \\
&= \frac{(1 - p)^{n+k-1} p}{\sum_{i=k}^{\infty} (1 - p)^i p} \\
&= \frac{(1 - p)^{n+k-1} p}{(1 - p)^k} \\
&= (1 - p)^{n-1} p \\
&= \Pr(X = n).
\end{aligned}$$

The fourth equality uses the fact that, for $0 < x < 1$, $\sum_{i=k}^{\infty} x^i = x^k/(1 - x)$. ■

We now turn to computing the expectation of a geometric random variable. When a random variable takes values in the set of natural numbers $\mathbf{N} = \{0, 1, 2, 3, \ldots\}$, there is an alternative formula for calculating its expectation.

Lemma 2.9: *Let X be a discrete random variable that takes on only nonnegative integer values. Then*

$$\mathbf{E}[X] = \sum_{i=1}^{\infty} \Pr(X \geq i).$$

Proof:

$$\begin{aligned}\sum_{i=1}^{\infty} \Pr(X \geq i) &= \sum_{i=1}^{\infty} \sum_{j=i}^{\infty} \Pr(X = j) \\ &= \sum_{j=1}^{\infty} \sum_{i=1}^{j} \Pr(X = j) \\ &= \sum_{j=1}^{\infty} j \Pr(X = j) \\ &= \mathbf{E}[X].\end{aligned}$$

The interchange of (possibly) infinite summations is justified, since the terms being summed are all nonnegative. ∎

For a geometric random variable X with parameter p,

$$\Pr(X \geq i) = \sum_{n=i}^{\infty} (1-p)^{n-1} p = (1-p)^{i-1}.$$

Hence

$$\begin{aligned}\mathbf{E}[X] &= \sum_{i=1}^{\infty} \Pr(X \geq i) \\ &= \sum_{i=1}^{\infty} (1-p)^{i-1} \\ &= \frac{1}{1-(1-p)} \\ &= \frac{1}{p}.\end{aligned}$$

Thus, for a fair coin where $p = 1/2$, on average it takes two flips to see the first heads.

There is another approach to finding the expectation of a geometric random variable X with parameter p – one that uses conditional expectations and the memoryless property of geometric random variables. Recall that X corresponds to the number of flips until the first heads given that each flip is heads with probability p. Let $Y = 0$ if the first flip is tails and $Y = 1$ if the first flip is heads. By the identity from Lemma 2.5,

$$\begin{aligned}\mathbf{E}[X] &= \Pr(Y = 0)\mathbf{E}[X \mid Y = 0] + \Pr(Y = 1)\mathbf{E}[X \mid Y = 1] \\ &= (1 - p)\mathbf{E}[X \mid Y = 0] + p\mathbf{E}[X \mid Y = 1].\end{aligned}$$

If $Y = 1$ then $X = 1$, so $\mathbf{E}[X \mid Y = 1] = 1$. If $Y = 0$, then $X > 1$. In this case, let the number of remaining flips (after the first flip until the first heads) be Z. Then, by the linearity of expectations,

$$\mathbf{E}[X] = (1 - p)\mathbf{E}[Z + 1] + p \cdot 1 = (1 - p)\mathbf{E}[Z] + 1.$$

By the memoryless property of geometric random variables, Z is also a geometric random variable with parameter p. Hence $\mathbf{E}[Z] = \mathbf{E}[X]$, since they both have the same distribution. We therefore have

$$\mathbf{E}[X] = (1 - p)\mathbf{E}[Z] + 1 = (1 - p)\mathbf{E}[X] + 1,$$

which yields $\mathbf{E}[X] = 1/p$.

This method of using conditional expectations to compute an expectation is often useful, especially in conjunction with the memoryless property of a geometric random variable.

2.4.1. *Example: Coupon Collector's Problem*

The coupon collector's problem arises from the following scenario. Suppose that each box of cereal contains one of n different coupons. Once you obtain one of every type of coupon, you can send in for a prize. Assuming that the coupon in each box is chosen independently and uniformly at random from the n possibilities and that you do not collaborate with others to collect coupons, how many boxes of cereal must you buy before you obtain at least one of every type of coupon? This simple problem arises in many different scenarios and will reappear in several places in the book.

Let X be the number of boxes bought until at least one of every type of coupon is obtained. We now determine $\mathbf{E}[X]$. If X_i is the number of boxes bought while you had exactly $i - 1$ different coupons, then clearly $X = \sum_{i=1}^{n} X_i$.

The advantage of breaking the random variable X into a sum of n random variables X_i, $i = 1, \ldots, n$, is that each X_i is a geometric random variable. When exactly $i - 1$ coupons have been found, the probability of obtaining a new coupon is

$$p_i = 1 - \frac{i - 1}{n}.$$

Hence, X_i is a geometric random variable with parameter p_i, and

$$\mathbf{E}[X_i] = \frac{1}{p_i} = \frac{n}{n - i + 1}.$$

Using the linearity of expectations, we have that

$$\begin{aligned}\mathbf{E}[X] &= \mathbf{E}\left[\sum_{i=1}^{n} X_i\right] \\ &= \sum_{i=1}^{n} \mathbf{E}[X_i] \\ &= \sum_{i=1}^{n} \frac{n}{n-i+1} \\ &= n \sum_{i=1}^{n} \frac{1}{i}.\end{aligned}$$

The summation $\sum_{i=1}^{n} 1/i$ is known as the *harmonic number* $H(n)$, and as we show next, $H(n) = \ln n + \Theta(1)$. Thus, for the coupon collector's problem, the expected number of random coupons required to obtain all n coupons is $n \ln n + \Theta(n)$.

Lemma 2.10: *The harmonic number* $H(n) = \sum_{i=1}^{n} 1/i$ *satisfies* $H(n) = \ln n + \Theta(1)$.

Proof: Since $1/x$ is monotonically decreasing, we can write

$$\ln n = \int_{x=1}^{n} \frac{1}{x}\, dx \leq \sum_{k=1}^{n} \frac{1}{k}$$

and

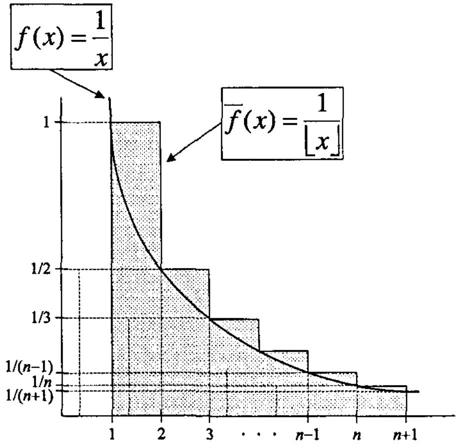
$$\sum_{k=2}^{n} \frac{1}{k} \leq \int_{x=1}^{n} \frac{1}{x}\, dx = \ln n.$$

This is clarified in Figure 2.1, where the area below the curve $f(x) = 1/x$ corresponds to the integral and the areas of the shaded regions correspond to the summations $\sum_{k=1}^{n} 1/k$ and $\sum_{k=2}^{n} 1/k$.

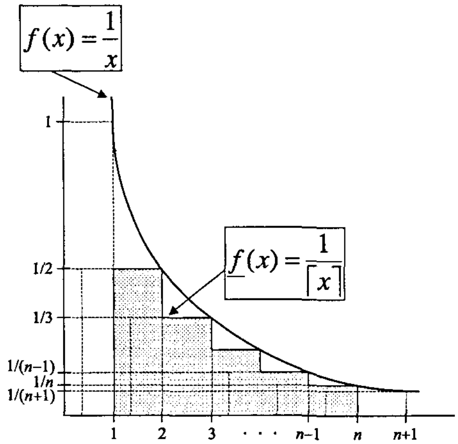
Hence $\ln n \leq H(n) \leq \ln n + 1$, proving the claim. ■

As a simple application of the coupon collector's problem, suppose that packets are sent in a stream from a source host to a destination host along a fixed path of routers. The host at the destination would like to know which routers the stream of packets has passed through, in case it finds later that some router damaged packets that it processed. If there is enough room in the packet header, each router can append its identification number to the header, giving the path. Unfortunately, there may not be that much room available in the packet header.

Suppose instead that each packet header has space for exactly one router identification number, and this space is used to store the identification of a router chosen uniformly at random from all of the routers on the path. This can actually be accomplished easily; we consider how in Exercise 2.18. Then, from the point of view of the destination host, determining all the routers on the path is like a coupon collector's problem. If there are n routers along the path, then the expected number of packets in



(a) Approximating $1/x$ from above.



(b) Approximating $1/x$ from below.

Figure 2.1: Approximating the area above and below $f(x) = 1/x$.

the stream that must arrive before the destination host knows all of the routers on the path is $nH(n) = n \ln n + \Theta(n)$.

2.5. Application: The Expected Run-Time of Quicksort

Quicksort is a simple – and, in practice, very efficient – sorting algorithm. The input is a list of n numbers $x_1, x_2, \dots, x_n$. For convenience, we will assume that the numbers are distinct. A call to the Quicksort function begins by choosing a *pivot* element from the set. Let us assume the pivot is x. The algorithm proceeds by comparing every other element to x, dividing the list of elements into two sublists: those that are less than x and those that are greater than x. Notice that if the comparisons are performed in the natural order, from left to right, then the order of the elements in each sublist is the same as in the initial list. Quicksort then recursively sorts these sublists.

In the worst case, Quicksort requires $\Omega(n^2)$ comparison operations. For example, suppose our input has the form $x_1 = n$, $x_2 = n - 1$, ..., $x_{n-1} = 2$, $x_n = 1$. Suppose also that we adopt the rule that the pivot should be the first element of the list. The first pivot chosen is then n, so Quicksort performs $n - 1$ comparisons. The division has yielded one sublist of size 0 (which requires no additional work) and another of size $n - 1$, with the order $n - 1, n - 2, \dots, 2, 1$. The next pivot chosen is $n - 1$, so Quicksort performs $n - 2$ comparisons and is left with one group of size $n - 2$ in the order $n - 2, n - 3, \dots, 2, 1$. Continuing in this fashion, Quicksort performs

$$(n-1) + (n-2) + \cdots + 2 + 1 = \frac{n(n-1)}{2} \text{comparisons.}$$

This is not the only bad case that leads to $\Omega(n^2)$ comparisons; similarly poor performance occurs if the pivot element is chosen from among the smallest few or the largest few elements each time.

Quicksort Algorithm:

Input: A list $S = \{x_1, \ldots, x_n\}$ of n distinct elements over a totally ordered universe.

Output: The elements of S in sorted order.

1. If S has one or zero elements, return S. Otherwise continue.
2. Choose an element of S as a pivot; call it s.
3. Compare every other element of S to x in order to divide the other elements into two sublists:
   (a) S_1 has all the elements of S that are less than x;
   (b) S_2 has all those that are greater than x.
4. Use Quicksort to sort S_1 and S_2.
5. Return the list S_1, x, S_2.

Algorithm 2.1: Quicksort.

We clearly made a bad choice of pivots for the given input. A reasonable choice of pivots would require many fewer comparisons. For example, if our pivot always splits the list into two sublists of size at most $\lceil n/2 \rceil$, then the number of comparisons $C(n)$ would obey the following recurrence relation:

$$C(n) \leq 2C(\lceil n/2 \rceil) + \Theta(n).$$

The solution to this equation yields $C(n) = O(n \log n)$, which is the best possible result for comparison-based sorting. In fact, any sequence of pivot elements that always split the input list into two sublists each of size at least cn for some constant c would yield an $O(n \log n)$ running time.

This discussion provides some intuition for how we would like pivots to be chosen. In each iteration of the algorithm there is a good set of pivot elements that split the input list into two almost equal sublists; it suffices if the sizes of the two sublists are within a constant factor of each other. There is also a bad set of pivot elements that do not split up the list significantly. If good pivots are chosen sufficiently often, Quicksort will terminate quickly. How can we guarantee that the algorithm chooses good pivot elements sufficiently often? We can resolve this problem in one of two ways.

First, we can change the algorithm to choose the pivots randomly. This makes Quicksort a randomized algorithm; the randomization makes it extremely unlikely that we repeatedly choose the wrong pivots. We demonstrate shortly that the expected number of comparisons made by a simple randomized Quicksort is $2n \ln n + O(n)$, matching (up to constant factors) the $\Omega(n \log n)$ bound for comparison-based sorting. Here, the expectation is over the random choice of pivots.

A second possibility is that we can keep our deterministic algorithm, using the first list element as a pivot, but consider a probabilistic model of the inputs. A *permutation* of a set of n distinct items is just one of the $n!$ orderings of these items. Instead of looking for the worst possible input, we assume that the input items are given to us in a random order. This may be a reasonable assumption for some applications; alternatively, this could be accomplished by ordering the input list according to a randomly chosen permutation before running the deterministic Quicksort algorithm. In this case, we have a deterministic algorithm but a *probabilistic analysis* based on a model of the inputs. We again show in this setting that the expected number of comparisons made is $2n \ln n + O(n)$. Here, the expectation is over the random choice of inputs.

The same techniques are generally used both in analyses of randomized algorithms and in probabilistic analyses of deterministic algorithms. Indeed, in this application the analysis of the randomized Quicksort and the probabilistic analysis of the deterministic Quicksort under random inputs are essentially the same.

Let us first analyze Random Quicksort, the randomized algorithm version of Quicksort.

Theorem 2.11: *Suppose that, whenever a pivot is chosen for Random Quicksort, it is chosen independently and uniformly at random from all possibilities. Then, for any input, the expected number of comparisons made by Random Quicksort is* $2n \ln n + O(n)$.

Proof: Let $y_1, y_2, \ldots, y_n$ be the same values as the input values $x_1, x_2, \ldots, x_n$ but sorted in increasing order. For $i < j$, let X_{ij} be a random variable that takes on the value 1 if y_i and y_j are compared at any time over the course of the algorithm, and 0 otherwise. Then the total number of comparisons X satisfies

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij},$$

and

$$\begin{aligned} \mathbf{E}[X] &= \mathbf{E}\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right] \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{E}[X_{ij}] \end{aligned}$$

by the linearity of expectations.

Since X_{ij} is an indicator random variable that takes on only the values 0 and 1, $\mathbf{E}[X_{ij}]$ is equal to the probability that X_{ij} is 1. Hence all we need to do is compute the probability that two elements y_i and y_j are compared. Now, y_i and y_j are compared if and only if either y_i or y_j is the first pivot selected by Random Quicksort from the set $Y^{ij} = \{y_i, y_{i+1}, \ldots, y_{j-1}, y_j\}$. This is because if y_i (or y_j) is the first pivot selected from this set, then y_i and y_j must still be in the same sublist, and hence they will be compared. Similarly, if neither is the first pivot from this set, then y_i and y_j will be separated into distinct sublists and so will not be compared.

Since our pivots are chosen independently and uniformly at random from each sublist, it follows that, the first time a pivot is chosen from Y^{ij}, it is equally likely to be any element from this set. Thus the probability that y_i or y_j is the first pivot selected from Y^{ij}, which is the probability that $X_{ij} = 1$, is $2/(j - i + 1)$. Using the substitution

$k = j - i + 1$ then yields

$$\begin{aligned}
\mathbf{E}[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} \\
&= \sum_{k=2}^{n} \sum_{i=1}^{n+1-k} \frac{2}{k} \\
&= \sum_{k=2}^{n} (n+1-k)\frac{2}{k} \\
&= \left((n+1)\sum_{k=2}^{n} \frac{2}{k}\right) - 2(n-1) \\
&= (2n+2)\sum_{k=1}^{n} \frac{1}{k} - 4n.
\end{aligned}$$

Notice that we used a rearrangement of the double summation to obtain a clean form for the expectation.

Recalling that the summation $H(n) = \sum_{k=1}^{n} 1/k$ satisfies $H(n) = \ln n + \Theta(1)$, we have $\mathbf{E}[X] = 2n \ln n + \Theta(n)$. ■

Next we consider the deterministic version of Quicksort, on random input. We assume that the order of the elements in each recursively constructed sublist is the same as in the initial list.

Theorem 2.12: *Suppose that, whenever a pivot is chosen for Quicksort, the first element of the sublist is chosen. If the input is chosen uniformly at random from all possible permutations of the values, then the expected number of comparisons made by Deterministic Quicksort is* $2n \ln n + O(n)$.

Proof: The proof is essentially the same as for Random Quicksort. Again, y_i and y_j are compared if and only if either y_i or y_j is the first pivot selected by Quicksort from the set Y^{ij}. Since the order of elements in each sublist is the same as in the original list, the first pivot selected from the set Y^{ij} is just the first element from Y^{ij} in the input list, and since all possible permutations of the input values are equally likely, every element in Y^{ij} is equally likely to be first. From this, we can again use linearity of expectations in the same way as in the analysis of Random Quicksort to obtain the same expression for $\mathbf{E}[X]$. ■

2.6. Exercises

Exercise 2.1: Suppose we roll a fair k-sided die with the numbers 1 through k on the die's faces. If X is the number that appears, what is $\mathbf{E}[X]$?

Exercise 2.2: A monkey types on a 26-letter keyboard that has lowercase letters only. Each letter is chosen independently and uniformly at random from the alphabet. If the monkey types 1,000,000 letters, what is the expected number of times the sequence "proof" appears?

Exercise 2.3: Give examples of functions f and random variables X where $\mathbf{E}[f(X)] < f(\mathbf{E}[X])$, $\mathbf{E}[f(X)] = f(\mathbf{E}[X])$, and $\mathbf{E}[f(X)] > f(\mathbf{E}[X])$.

Exercise 2.4: Prove that $\mathbf{E}[X^k] \geq \mathbf{E}[X]^k$ for any even integer $k \geq 1$.

Exercise 2.5: If X is a $B(n, 1/2)$ random variable with $n \geq 1$, show that the probability that X is even is $1/2$.

Exercise 2.6: Suppose that we independently roll two standard six-sided dice. Let X_1 be the number that shows on the first die, X_2 the number on the second die, and X the sum of the numbers on the two dice.

(a) What is $\mathbf{E}[X \mid X_1 \text{ is even}]$?
(b) What is $\mathbf{E}[X \mid X_1 = X_2]$?
(c) What is $\mathbf{E}[X_1 \mid X = 9]$?
(d) What is $\mathbf{E}[X_1 - X_2 \mid X = k]$ for k in the range $[2, 12]$?

Exercise 2.7: Let X and Y be independent geometric random variables, where X has parameter p and Y has parameter q.

(a) What is the probability that $X = Y$?
(b) What is $\mathbf{E}[\max(X, Y)]$?
(c) What is $\Pr(\min(X, Y) = k)$?
(d) What is $\mathbf{E}[X \mid X \leq Y]$?

You may find it helpful to keep in mind the memoryless property of geometric random variables.

Exercise 2.8: **(a)** Alice and Bob decide to have children until either they have their first girl or they have $k \geq 1$ children. Assume that each child is a boy or girl independently with probability $1/2$ and that there are no multiple births. What is the expected number of female children that they have? What is the expected number of male children that they have?

(b) Suppose Alice and Bob simply decide to keep having children until they have their first girl. Assuming that this is possible, what is the expected number of boys that they have?

Exercise 2.9: **(a)** Suppose that we roll twice a fair k-sided die with the numbers 1 through k on the die's faces, obtaining values X_1 and X_2. What is $\mathbf{E}[\max(X_1, X_2)]$? What is $\mathbf{E}[\min(X_1, X_2)]$?

(b) Show from your calculations in part (a) that

$$\mathbf{E}[\max(X_1, X_2)] + \mathbf{E}[\min(X_1, X_2)] = \mathbf{E}[X_1] + \mathbf{E}[X_2]. \tag{2.1}$$

(c) Explain why Eqn. (2.1) must be true by using the linearity of expectations instead of a direct computation.

Exercise 2.10: **(a)** Show by induction that if $f : \mathbf{R} \to \mathbf{R}$ is convex then, for any $x_1, x_2, \ldots, x_n$ and $\lambda_1, \lambda_2, \ldots, \lambda_n$ with $\sum_{i=1}^{n} \lambda_i = 1$,

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i). \tag{2.2}$$

(b) Use Eqn. (2.2) to prove that if $f : \mathbf{R} \to \mathbf{R}$ is convex then

$$\mathbf{E}[f(X)] \geq f(\mathbf{E}[X])$$

for any random variable X that takes on only finitely many values.

Exercise 2.11: Prove Lemma 2.6.

Exercise 2.12: We draw cards uniformly at random with replacement from a deck of n cards. What is the expected number of cards we must draw until we have seen all n cards in the deck? If we draw $2n$ cards, what is the expected number of cards in the deck that are not chosen at all? Chosen exactly once?

Exercise 2.13: **(a)** Consider the following variation of the coupon collector's problem. Each box of cereal contains one of $2n$ different coupons. The coupons are organized into n pairs, so that coupons 1 and 2 are a pair, coupons 3 and 4 are a pair, and so on. Once you obtain one coupon from every pair, you can obtain a prize. Assuming that the coupon in each box is chosen independently and uniformly at random from the $2n$ possibilities, what is the expected number of boxes you must buy before you can claim the prize?

(b) Generalize the result of the problem in part (a) for the case where there are kn different coupons, organized into n disjoint sets of k coupons, so that you need one coupon from every set.

Exercise 2.14: The geometric distribution arises as the distribution of the number of times we flip a coin until it comes up heads. Consider now the distribution of the number of flips X until the kth head appears, where each coin flip comes up heads independently with probability p. Prove that this distribution is given by

$$\Pr(X = n) = \binom{n-1}{k-1} p^k (1-p)^{n-k}$$

for $n \geq k$. (This is known as the *negative binomial* distribution.)

Exercise 2.15: For a coin that comes up heads independently with probability p on each flip, what is the expected number of flips until the kth heads?

Exercise 2.16: Suppose we flip a coin n times to obtain a sequence of flips X_1, $X_2, \ldots, X_n$. A *streak* of flips is a consecutive subsequence of flips that are all the same. For example, if X_3, X_4, and X_5 are all heads, there is a streak of length 3 starting at the third flip. (If X_6 is also heads, then there is also a streak of length 4 starting at the third flip.)

(a) Let n be a power of 2. Show that the expected number of streaks of length $\log_2 n + 1$ is $1 - o(1)$.

(b) Show that, for sufficiently large n, the probability that there is no streak of length at least $\lfloor \log_2 n - 2\log_2 \log_2 n \rfloor$ is less than $1/n$. (*Hint:* Break the sequence of flips up into disjoint blocks of $\lfloor \log_2 n - 2\log_2 \log_2 n \rfloor$ consecutive flips, and use that the event that one block is a streak is independent of the event that any other block is a streak.)

Exercise 2.17: Recall the recursive spawning process described in Section 2.3. Suppose that each call to process $\mathcal{S}$ recursively spawns new copies of the process $\mathcal{S}$, where the number of new copies is 2 with probability p and 0 with probability $1 - p$. If Y_i denotes the number of copies of $\mathcal{S}$ in the ith generation, determine $\mathbf{E}[Y_i]$. For what values of p is the expected total number of copies bounded?

Exercise 2.18: The following approach is often called *reservoir sampling*. Suppose we have a sequence of items passing by one at a time. We want to maintain a sample of one item with the property that it is uniformly distributed over all the items that we have seen at each step. Moreover, we want to accomplish this without knowing the total number of items in advance or storing all of the items that we see.

Consider the following algorithm, which stores just one item in memory at all times. When the first item appears, it is stored in the memory. When the kth item appears, it replaces the item in memory with probability $1/k$. Explain why this algorithm solves the problem.

Exercise 2.19: Suppose that we modify the reservoir sampling algorithm of Exercise 2.18 so that, when the kth item appears, it replaces the item in memory with probability $1/2$. Describe the distribution of the item in memory.

Exercise 2.20: A permutation on the numbers $[1, n]$ can be represented as a function $\pi : [1, n] \to [1, n]$, where $\pi(i)$ is the position of i in the ordering given by the permutation. A fixed point of a permutation $\pi : [1, n] \to [1, n]$ is a value for which $\pi(x) = x$. Find the expected number of fixed points of a permutation chosen uniformly at random from all permutations.

Exercise 2.21: Let $a_1, a_2, \ldots, a_n$ be a random permutation of $\{1, 2, \ldots, n\}$, equally likely to be any of the $n!$ possible permutations. When sorting the list $a_1, a_2, \ldots, a_n$, the element a_i must move a distance of $|a_i - i|$ places from its current position to reach its position in the sorted order. Find

$$\mathbf{E}\left[\sum_{i=1}^{n} |a_i - i|\right],$$

the expected total distance that elements will have to be moved.

Exercise 2.22: Let $a_1, a_2, \ldots, a_n$ be a list of n distinct numbers. We say that a_i and a_j are *inverted* if $i < j$ but $a_i > a_j$. The *Bubblesort* sorting algorithm swaps pairwise adjacent inverted numbers in the list until there are no more inversions, so the list is in sorted order. Suppose that the input to Bubblesort is a random permutation, equally likely to be any of the $n!$ permutations of n distinct numbers. Determine the expected number of inversions that need to be corrected by Bubblesort.

Exercise 2.23: *Linear insertion* sort can sort an array of numbers in place. The first and second numbers are compared; if they are out of order, they are swapped so that they are in sorted order. The third number is then placed in the appropriate place in the sorted order. It is first compared with the second; if it is not in the proper order, it is swapped and compared with the first. Iteratively, the kth number is handled by swapping it downward until the first k numbers are in sorted order. Determine the expected number of swaps that need to be made with a linear insertion sort when the input is a random permutation of n distinct numbers.

Exercise 2.24: We roll a standard fair die over and over. What is the expected number of rolls until the first pair of consecutive sixes appears? (*Hint:* The answer is not 36.)

Exercise 2.25: A blood test is being performed on n individuals. Each person can be tested separately, but this is expensive. Pooling can decrease the cost. The blood samples of k people can be pooled and analyzed together. If the test is negative, this one test suffices for the group of k individuals. If the test is positive, then each of the k persons must be tested separately and thus $k + 1$ total tests are required for the k people.

Suppose that we create n/k disjoint groups of k people (where k divides n) and use the pooling method. Assume that each person has a positive result on the test independently with probability p.

(a) What is the probability that the test for a pooled sample of k people will be positive?
(b) What is the expected number of tests necessary?
(c) Describe how to find the best value of k.
(d) Give an inequality that shows for what values of p pooling is better than just testing every individual.

Exercise 2.26: A permutation $\pi : [1, n] \to [1, n]$ can be represented as a set of cycles as follows. Let there be one vertex for each number i, $i = 1, \ldots, n$. If the permutation maps the number i to the number $\pi(i)$, then a directed arc is drawn from vertex i to vertex $\pi(i)$. This leads to a graph that is a set of disjoint cycles. Notice that some of the cycles could be self-loops. What is the expected number of cycles in a random permutation of n numbers?

Exercise 2.27: Consider the following distribution on the integers $x \geq 1$: $\Pr(X = x) = (6/\pi^2)x^{-2}$. This is a valid distribution, since $\sum_{k=1}^{\infty} k^{-2} = \pi^2/6$. What is its expectation?

Exercise 2.28: Consider a simplified version of roulette in which you wager x dollars on either red or black. The wheel is spun, and you receive your original wager plus another x dollars if the ball lands on your color; if the ball doesn't land on your color, you lose your wager. Each color occurs independently with probability $1/2$. (This is a simplification because real roulette wheels have one or two spaces that are neither red nor black, so the probability of guessing the correct color is actually less than $1/2$.)

The following gambling strategy is a popular one. On the first spin, bet 1 dollar. If you lose, bet 2 dollars on the next spin. In general, if you have lost on the first $k-1$ spins, bet 2^{k-1} dollars on the kth spin. Argue that by following this strategy you will eventually win a dollar. Now let X be the random variable that measures your maximum loss before winning (i.e., the amount of money you have lost before the play on which you win). Show that $\mathbf{E}[X]$ is unbounded. What does it imply about the practicality of this strategy?

Exercise 2.29: Prove that, if $X_0, X_1, \ldots$ is a sequence of random variables such that

$$\sum_{j=0}^{\infty} \mathbf{E}[|X_j|]$$

converges, then the linearity of expectations holds:

$$\mathbf{E}\left[\sum_{j=0}^{\infty} X_j\right] = \sum_{j=0}^{\infty} \mathbf{E}[X_j].$$

Exercise 2.30: In the roulette problem of Exercise 2.28, we found that with probability 1 you eventually win a dollar. Let X_j be the amount you win on the jth bet. (This might be 0 if you have already won a previous bet.) Determine $\mathbf{E}[X_j]$ and show that, by applying the linearity of expectations, you find your expected winnings are 0. Does the linearity of expectations hold in this case? (Compare with Exercise 2.29.)

Exercise 2.31: A variation on the roulette problem of Exercise 2.28 is the following. We repeatedly flip a fair coin. You pay j dollars to play the game. If the first head comes up on the kth flip, you win $2^k/k$ dollars. What are your expected winnings? How much would you be willing to pay to play the game?

Exercise 2.32: You need a new staff assistant, and you have n people to interview. You want to hire the best candidate for the position. When you interview a candidate, you can give them a score, with the highest score being the best and no ties being possible.

You interview the candidates one by one. Because of your company's hiring practices, after you interview the kth candidate, you either offer the candidate the job before the next interview or you forever lose the chance to hire that candidate. We suppose the candidates are interviewed in a random order, chosen uniformly at random from all $n!$ possible orderings.

We consider the following strategy. First, interview m candidates but reject them all; these candidates give you an idea of how strong the field is. After the mth candidate, hire the first candidate you interview who is better than all of the previous candidates you have interviewed.

(a) Let E be the event that we hire the best assistant, and let E_i be the event that ith candidate is the best and we hire him. Determine $\Pr(E_i)$, and show that

$$\Pr(E) = \frac{m}{n} \sum_{j=m+1}^{n} \frac{1}{j-1}.$$

(b) Bound $\sum_{j=m+1}^{n} \frac{1}{j-1}$ to obtain

$$\frac{m}{n}(\ln n - \ln m) \leq \Pr(E) \leq \frac{m}{n}(\ln(n-1) - \ln(m-1)).$$

(c) Show that $m(\ln n - \ln m)/n$ is maximized when $m = n/e$, and explain why this means $\Pr(E) \geq 1/e$ for this choice of m.