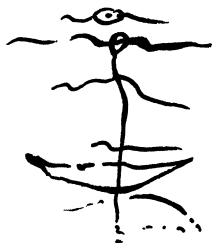


Introduction



*To doubt everything or to believe everything –
these are two equally easy solutions,
because both of them relieve us of the necessity of thinking.*



Jules Henri Poincaré

1.1 What Is Randomness and Does There Exist True Randomness?

The notion of “randomness” is one of the most fundamental and most discussed terms in science. Following the definition used in dictionaries, an event is considered to be *random* when it happens unpredictably. An object is called *random*, when it is created without any plan. The fundamental question is whether randomness really exists, or whether we use this term only to model objects and events with unknown lawfulness. Philosophers and scientists have disputed the answer to this question since ancient times. Democritus believed that

*the randomness is the unknown,
and that the nature is determined
in its fundamentals.*

Thus, Democritus asserted that order conquers the world and this order is governed by unambiguous laws. Following Democritus’s opinion, one uses the notion of “randomness” only in the subjective sense in order to veil one’s inability to truly understand the nature of events and things. Hence the existence of the notion of randomness is only a consequence of the incompleteness of our knowledge. To present his opinion transparently, Democritus liked to use the following example. Two men agreed on sending their slaves to bring water at the same time in order to cause the slaves to meet. The slaves really met at the source and said, “Oh, this is randomness that we have met.”

In contrast to Democritus, Epikurus claimed that

*the randomness is objective,
it is the proper nature of events.*

Thus, Epikurus claimed that there exists a true randomness that is completely independent of our knowledge. Epikurus’s opinion was that there exist processes whose development is ambiguous rather than unambiguous, and an

unpredictable choice from the existing possibilities is what we call randomness.

One could simply say, “Epikurus was right because there are games of chance, such as rolling dice or roulette, that can have different outcomes, and the results are determined by chance. Unfortunately, the story is not so simple, and discussing gambling one gets the support for the opinion of Democritus rather than for Epikurus’s view on the nature of events. Rolling dice is a very complex activity, but if one knows the direction, the speed and the surface on which a die is tossed, then it may be possible to compute and predict the result. Obviously, the movement of the hand controlled by the human brain is too complex to allow us to estimate the values of all important parameters. But we may not consider the process of rolling a die as an objectively random process only because it is too complex for us to predict the outcome. The same is true of roulette and other games of chance. Physics also often uses random models to describe and analyze physical processes that are not inherently or necessarily random (and are sometimes clearly deterministic), but which are too complex to have a realistic possibility of modeling them in a fully deterministic way. It is interesting to note that based on this observation even Albert Einstein accepted the notion of randomness only in relation to an incomplete knowledge, and strongly believed in the existence of clear, deterministic laws for all processes in nature.¹

Before the 20th century, the world view of people was based on causality and determinism. The reasons for that were, first, religion, which did not accept the existence of randomness in a world created by God², and, later, the optimism created by the success of natural sciences and mechanical engineering in the 19th century, which gave people hope that everything could be discovered, and everything discovered could be explained by deterministic causalities of cause and resulting effect.³

This belief in determinism also had emotional roots, because people connected randomness (and even identified it) with chaos, uncertainty, and unpredictability, which were always related to fear; and so the possibility of random events was not accepted. To express the strongly negative connotation of randomness in the past, one can consider the following quotation of Marcus Aurelius:

*There are only two possibilities,
either a big chaos conquers the world,
or order and law.*

¹“God does not roll dice” is a famous quotation of Albert Einstein. The equally famous reply of Niels Bohr is, “The true God does not allow anybody to prescribe what He has to do.”

²Today we know that this strong interpretation is wrong and that the existence of true randomness does not contradict the existence of God.

³Take away the cause, and the effect must cease.

Because randomness was undesirable, it may not be surprising that philosophers and researchers performed their investigations without allowing the existence of random events in their concepts or even tried to prove the nonexistence of randomness by focusing on deterministic causalities. Randomness was in a similarly poor situation with Galileo Galilei, who claimed that the earth is not a fixed center of the whole universe. Though he was able to prove his claim by experimental observations, he did not have any chance to convince people about it because they were very afraid of such a reality. Life in the medieval world was very hard, and so people clung desperately to the very few assurances they had. And the central position of the earth in the universe supported the belief that the poor man is at the center of God's attention. The terrible fear of losing this assurance was the main reason for the situation, with nobody willing to verify the observations of Galileo Galilei. And the "poor" randomness had the same trouble gaining acceptance⁴.

Finally, scientific discoveries in the 20th century (especially in physics and biology) returned the world to Epikurus's view on randomness. The mathematical models of evolutionary biology show that random mutations of DNA have to be considered a crucial instrument of evolution. The essential reason for accepting the existence of randomness was one of the deepest discoveries in the history of science: the theory of quantum mechanics. The mathematical model of the behavior of particles is related to ambiguity, which can be described in terms of random events. All important predictions of this theory were proved experimentally, and so some events in the world of particles are considered as truly random events. For accepting randomness (or random events) it was very important to overcome the restricted interpretation of randomness, identifying it with chaos and uncertainty. A very elegant, modern view on randomness is given by the Hungarian mathematician Alfréd Rényi:

*Randomness and order do not contradict each other;
more or less both may be true at once.
The randomness controls the world
and due to this in the world there are order and law,
which can be expressed in measures of random events
that follow the laws of probability theory.*

For us, as computer scientists, it is important to realize that there is also another reason to deal with randomness than "only" the modeling of natural processes. Surprisingly, this reason was already formulated 200 years ago by the great German poet Johann Wolfgang von Goethe as follows:

*The tissue of the world
is built from necessities and randomness;
the intellect of men places itself between both*

⁴One does not like to speak about emotions in the so-called exact (hard) sciences, but this is a denial of the fact that the emotions of researchers (the subjects in the research) are the aggregates of the development and the progress.

*and can control them;
it considers the necessity
as the reason of its existence;
it knows how randomness can be
managed, controlled, and used...*

In this context Johann Wolfgang von Goethe is the first “computer scientist”, who recognized randomness as a useful source for performing some activities. The use of randomness as a resource of an unbelievable, phenomenal efficiency is the topic of this book. We aim to convince the reader that it can be very profitable to design and implement randomized algorithms and systems instead of completely deterministic ones. This realization is nothing other than the acceptance of nature as teacher. It seems to be the case that nature always uses the most efficient and simplest way to achieve its goal, and that randomization of a part of the control is an essential concept of nature’s strategy. Computer science practice confirms this point of view. In many everyday applications, simple randomized systems and algorithms do their work efficiently with a high degree of reliability, and we do not know any deterministic algorithms that would do the same with a comparable efficiency. We even know of examples where the design and use of deterministic counterparts of some randomized algorithms is beyond physical limits. This is also the reason why currently one does not relate tractability (practical solvability) with the efficiency of deterministic algorithms, but with efficient randomized algorithms.

To convince the reader of the enormous usefulness of randomization, the next section presents a randomized protocol that solves a concrete communication task within communication complexity that is substantially smaller than the complexity of the best possible deterministic protocol.

We close this section by calling attention to the fact that we did not give a final answer to the question of whether or not true randomness exists, and it is very improbable that science will be able to answer this question in the near future. The reason for this pessimism is that the question about the existence of randomness lies in the very fundamentals of science, i.e., on the level of axioms, and not on the level of results. And, on the level of axioms (basic assumptions), even the exact sciences like mathematics and physics do not have any generally valid assertion, but only assumptions expressed in the form of axioms. The only reason to believe in axioms is that they fully correspond to our experience and knowledge. An example of an axiom of mathematics (viewed as a formal language of science) is that our way of thinking is correct, and so all our formal arguments are reliable. Starting with the axioms, one builds the building of science very carefully, in such a way that all results achieved are true provided the axioms are valid. If an axiom

is shown to be not generally valid, one has to revise the entire theory built upon it⁵.

Here, we allow ourselves to believe in the existence of randomness, and not only because the experience and knowledge of physics and evolutionary theory support this belief. For us as computer scientists, the main reason to believe in randomness is that randomness can be a source of efficiency. Randomness enables us to reach aims incomparably faster, and it would be very surprising for us if nature left this great possibility unnoticed.

1.2 Randomness as a Source of Efficiency – an Exemplary Application

The aim of this section is to show that randomized algorithms can be essentially more efficient than their deterministic counterparts.

Let us consider the following scenario. We have two computers R_I and R_{II} (Figure 1.1) that are very far apart⁶. At the beginning both have a database with the same content. In the meantime the contents of these databases dynamically developed in such a way that one now tries to perform all changes simultaneously in both databases with the aim of getting the same database, with complete information about the database subject (for instance, genome sequences), in both locations. After some time, we want to check whether this process is successful, i.e., whether R_I and R_{II} contain the same data.

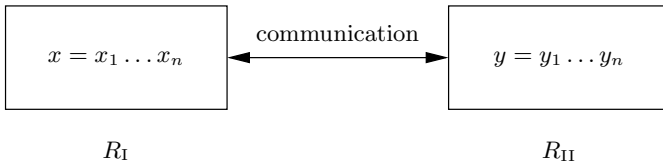


Fig. 1.1.

Let n be the size of the database in bits. For instance, n can be approximately $n = 10^{16}$, which is realistic for biological applications. Our goal is to design a communication protocol between R_I and R_{II} that is able to determine whether the data saved on both computers is the same or not. The complexity of the communication protocol is the number of bits that have to

⁵Disproving the general validity of an axiom should not be considered a “tragedy.” Such events are part of the development of science and they are often responsible for the greatest discoveries. The results built upon the old, disproved axiom usually need not be rejected; it is sufficient to relativize their validity, because they are true in frameworks where the old axiom is valid.

⁶For instance, one in Europe and one in America.

be exchanged between R_I and R_{II} in order to solve this decision problem, and we obviously try to minimize this complexity.

One can prove that every deterministic communication protocol solving this task must exchange at least n bits⁷ between R_I and R_{II} , i.e., there exists no deterministic protocol that solves this task by communicating $n-1$ or lower bits. Sending $n = 10^{16}$ bits and additionally assuring that all arrive safely⁸ at the other side is a practically nontrivial task, so one would probably not do it in this way.

A reasonable solution can be given by the following randomized protocol. Let $x = x_1x_2 \dots x_n \in \{0,1\}^*$, $x_i \in \{0,1\}$ for all $i = 1, \dots, n$. We denote by

$$\text{Number}(x) = \sum_{i=1}^n 2^{n-i} \cdot x_i$$

the natural number whose binary representation is the string x .

$R = (R_I, R_{II})$ (Randomized Protocol for Equality)

Initial situation: R_I has a sequence x of n bits, $x = x_1 \dots x_n$, and R_{II} has a sequence y of n bits $y = y_1 \dots y_n$.

Phase 1: R_I chooses uniformly⁹ a prime p from the interval $[2, n^2]$ at random.

Phase 2: R_I computes the integer

$$s = \text{Number}(x) \bmod p$$

and sends the binary representations of s and p to R_{II} .

{Observe that $s \leq p < n^2$ and so each of these integers can be represented by $\lceil \log_2 n^2 \rceil$ bits.}

Phase 3: After reading s and p , R_{II} computes the number

$$q = \text{Number}(y) \bmod p.$$

If $q \neq s$, then R_{II} outputs “ $x \neq y$ ”.

If $q = s$, then R_{II} outputs “ $x = y$ ”.

Now we analyze the work of $R = (R_I, R_{II})$. First, we look at the complexity measured in the number of communication bits, and then we analyze the reliability (error probability) of the randomized protocol $R = (R_I, R_{II})$.

The only communication of the protocol involves submitting the binary representations of the positive integers s and p . As we have already observed, $s \leq p < n^2$; hence the length of the message is at most¹⁰

⁷This means that sending all data of R_I to R_{II} for the comparison is an optimal communication strategy.

⁸without flipping a bit

⁹This means that every prime from the interval $[2, n^2]$ has the same probability of being chosen.

¹⁰Every positive integer m can be represented by $\lceil \log_2(m+1) \rceil$ bits.

$$2 \cdot \lceil \log_2 n^2 \rceil \leq 4 \cdot \lceil \log_2 n \rceil.$$

For $n = 10^{16}$, the binary length of the message is at most $4 \cdot 16 \cdot \lceil \log_2 10 \rceil = 256$. This is a very short message that can be safely transferred.

Now we show not only that for most inputs (initial situations) the randomized strategy works, but also show that the probability of providing the right answer is high for every input. Let us first recognize that the randomized protocol may err¹¹. For instance, if $x = 01111$ and $y = 10110$, i.e., $\text{Number}(x) = 15$ and $\text{Number}(y) = 22$, then the choice of the prime 7 from the set $\{2, 3, 5, 7, 11, 13, 17, 19, 23\}$ yields the wrong answer, because

$$15 \bmod 7 = 1 = 22 \bmod 7.$$

To analyze the error probability for any input (x, y) , with $x = x_1 \dots x_n$, and $y = y_1 \dots y_n$, we partition the set

$$\text{PRIM}(n^2) = \{p \text{ is a prime} \mid p \leq n^2\}$$

into two subsets (Figure 1.2). The first subset contains the **bad primes**, where a prime p is **bad for (x, y)** if the random choice of p results in the wrong output of the protocol R . The second subset of $\text{PRIM}(n^2)$ is the complementary subset to the subset of bad primes and we call the primes in this subset **good for (x, y)** because the choice of any of them results in the right answer for the input (x, y) .

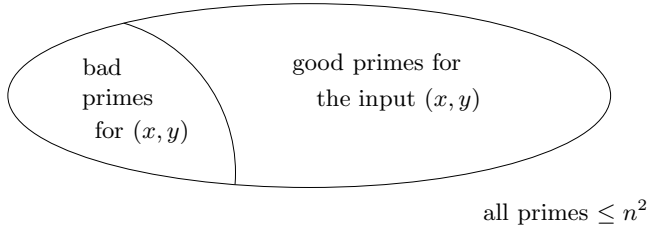


Fig. 1.2.

Since every prime in $\text{PRIM}(n^2)$ has the same probability of being chosen, the error probability¹² for the input (x, y) is

$$\frac{\text{the number of bad primes for } (x, y)}{\text{Prim}(n^2)},$$

¹¹In the sense that the randomized protocol outputs “ $x = y$ ” for different x and y .

¹²Here we work with an informal understanding of the notion of probability. The exact definition of probability and related notions will be presented in the next chapter, and we will then repeat this argument formally.

where $\text{Prim}(n^2)$ denotes the cardinality of $\text{Prim}(n^2)$. The famous Prime Number Theorem says that

$$\lim_{m \rightarrow \infty} \frac{\text{Prim}(m)}{m/\ln m} = 1,$$

and we know that

$$\text{Prim}(m) > \frac{m}{\ln m}$$

for all positive integers $m > 67$. Hence, we have

$$\text{Prim}(n^2) > \frac{n^2}{2 \ln n}$$

for all $n \geq 9$. Our aim is now to show that

for any input (x, y) , the number of bad primes for (x, y) is at most $n - 1$,

i.e., that the number of primes that are bad for (x, y) is essentially smaller than $n^2/2 \ln n$.

Analyzing the error probability, we distinguish two possibilities with respect to the real relation between x and y .

(i) Let $x = y$.

Then one has

$$\text{Number}(x) \bmod p = \text{Number}(y) \bmod p$$

for all primes p , i.e., there are no bad primes for the input (x, y) . Therefore R_{II} outputs “ $x = y$ ” with certainty, i.e., the error probability is equal to 0 in this case.

(ii) Let $x \neq y$.

One gets the wrong answer “ $x = y$ ” only if R_{I} has chosen a prime p such that

$$s = \text{Number}(x) \bmod p = \text{Number}(y) \bmod p.$$

In other words, p is a bad prime for (x, y) when

$$\text{Number}(x) = x' \cdot p + s \text{ and } \text{Number}(y) = y' \cdot p + s$$

for some nonnegative integers x' and y' .

A consequence is that

$$\text{Number}(x) - \text{Number}(y) = x' \cdot p - y' \cdot p = (x' - y') \cdot p,$$

i.e., that

$$p \text{ divides the number } |\text{Number}(x) - \text{Number}(y)|.$$

Thus, the protocol R outputs the wrong answer “ $x = y$ ” only if the chosen prime p divides the number $|\text{Number}(x) - \text{Number}(y)|$. This way, we have the following new definition of bad primes:

a prime p is bad for (x, y) iff

p divides the number $w = |\text{Number}(x) - \text{Number}(y)|$.

Thus, to estimate the error probability, it is sufficient to estimate how many primes from the $\text{Prim}(n^2) \sim n^2 / \ln n^2$ primes divide the number w . Since the length of the binary representations of x and y is equal to n ,

$$w = |\text{Number}(x) - \text{Number}(y)| < 2^n.$$

Obviously¹³, we can factorize w to get

$$w = p_1^{i_1} \cdot p_2^{i_2} \cdot \dots \cdot p_k^{i_k},$$

where $p_1 < p_2 < \dots < p_k$ are primes and i_1, i_2, \dots, i_k are positive integers. Our aim is to prove that

$$k \leq n - 1.$$

We prove it by contradiction. Assume $k \geq n$. Then,

$$w = p_1^{i_1} \cdot p_2^{i_2} \cdot \dots \cdot p_k^{i_k} \geq p_1 \cdot p_2 \cdot \dots \cdot p_n > 1 \cdot 2 \cdot 3 \cdot \dots \cdot n = n! > 2^n,$$

which contradicts the fact that $w < 2^n$. In this way we have proved that w has at most $n - 1$ different prime factors. Since every prime in $\{2, 3, \dots, n^2\}$ has the same probability of being chosen, the probability of choosing a bad prime p dividing w is at most

$$\frac{n - 1}{\text{Prim}(n^2)} \leq \frac{n - 1}{n^2 / \ln n^2} \leq \frac{\ln n^2}{n}$$

for all $n \geq 9$.

Thus the error probability of R for an input (x, y) with $x \neq y$ is at most

$$\frac{\ln n^2}{n},$$

which is at most

$$0.36892 \cdot 10^{-14}$$

for $n = 10^{16}$.

An error probability of this size is no real risk, but let us assume that a pessimist is not satisfied with this error probability and wants to have an error probability below all physical limits. In such a case one can execute the work of the protocol R ten times, always with an independent, new choice of a prime.

¹³We know from number theory that every positive integer has a unique factorization.

Protocol R_{10}

Initial situation: R_I has n bits $x = x_1 \dots x_n$ and R_{II} has n bits $y = y_1 \dots y_n$.

Phase 1: R_I chooses 10 uniformly random primes

$$p_1, p_2, \dots, p_{10}$$

from $\{2, 3, \dots, n^2\}$.

Phase 2: R_I computes

$$s_i = \text{Number}(x) \bmod p_i$$

for $i = 1, 2, \dots, 10$ and sends the binary representations of

$$p_1, p_2, \dots, p_{10}, s_1, s_2, \dots, s_{10}$$

to R_{II} .

Phase 3: Upon receiving $p_1, p_2, \dots, p_{10}, s_1, s_2, \dots, s_{10}$ R_{II} computes

$$q_i = \text{Number}(y) \bmod p_i$$

for $i = 1, 2, \dots, 10$.

If there exists an $i \in \{1, 2, \dots, 10\}$ such that $q_i \neq s_i$, then R_{II} outputs “ $x \neq y$ ”.

Else (if $q_j = s_j$ for all $j \in \{1, 2, \dots, 10\}$) R_{II} outputs “ $x = y$ ”.

We observe that the communication complexity of R_{10} is 10 times larger than that of R . But, for $n = 10^{16}$, the message consists of at most 2560 bits, which is no issue for discussion.

What is the gain with respect to error probability?

If $x = y$, then we again have the situation that the protocol R_{10} provides the right answer “ $x = y$ ” with certainty, i.e., the error probability is equal to 0.

However, if $x \neq y$, R_{10} outputs the wrong answer “ $x = y$ ” only if all 10 chosen primes belong to the maximal $n - 1$ bad primes that divide the difference $w = |\text{Number}(x) - \text{Number}(y)|$. Since the 10 bad primes are chosen in 10 independent experiments, the error probability is at most¹⁴

$$\left(\frac{n-1}{\text{Prim}(n^2)} \right)^{10} \leq \left(\frac{\ln n^2}{n} \right)^{10} = \frac{2^{10} \cdot (\ln n)^{10}}{n^{10}}.$$

For $n = 10^{16}$, the error probability is smaller than

$$0.4717 \cdot 10^{-141}.$$

¹⁴Why the probability of independently choosing two bad primes is equal to the multiplication of the probabilities of choosing a bad prime is carefully explained in Section 2.3.

If one takes into account the fact that the number of microseconds since the Big Bang is a number of 24 digits, and that the number of protons in the known universe is a number of 79 digits, an event with a probability below 10^{-141} is a real wonder. Note also that in the case where a deterministic protocol communication of 10^{16} bits would be executable, the costs speak clearly in favor of the implementation of the above randomized protocol.

We can learn a lot from the construction of the protocol R_{10} that consists of independent repetitions of R . We see that the error probability of a randomized algorithm A can be substantially pushed down by executing several independent runs of A . In cases such as the above communication protocol, even a few repetitions result in an enormous decrease in error probability.

We have observed that using randomization one can gain phenomenally in the efficiency by paying a very small price in reliability. Here we call attention to the fact that in practice randomized algorithms with very small error probability can be even more reliable than their best deterministic counterparts. What do we mean by this? Theoretically, all deterministic algorithms are absolutely correct, and randomized algorithms may err. But the nature of the story is that deterministic programs are not absolutely reliable because during their runs on a computer a hardware error may occur and then they may produce wrong results. Clearly, the probability of the occurrence of a hardware error grows proportionally with the running time of the program. Therefore a fast randomized algorithm can be more reliable than a slow deterministic algorithm. For instance, if a randomized algorithm computes a result in 10 seconds with an error probability 10^{-30} , then it is more reliable than a deterministic algorithm that computes the result in one week. Another good example is our randomized protocol R for Equality. For $n = 10^{16}$, the protocol R has to communicate 256 bits only and the error probability is at most $0.4 \cdot 10^{-14}$. On the other hand, every deterministic protocol has to safely communicate at least 10^{16} bits and the probability of flipping some of them because of a hardware error is essentially larger than the error probability of R .

1.3 Concept of the Book

The aim of this book is to provide an elementary course on the design and analysis of efficient randomized algorithms. We focus not on giving an overview of the deepest contributions to this area, but on a transparent presentation of the most successful design methods and concepts, and we try to contribute to understanding why randomized approaches can be essentially more efficient than their best deterministic counterparts. In this way, we aim to contribute to capturing formal methods as instruments for problem solving and to developing a feeling for the computer scientist's way of thinking.

The only presumed background for reading this textbook is a basic knowledge of introductory courses, such as “programming,” “algorithms and data

structures” and introduction to the “theory of computation.” Thus, we assume that the reader is familiar with terms such as computing task (or problem), decision problem, optimization problem, algorithm, and complexity of algorithms. We use the formal definitions of these basic terms, and the same notation as that presented in our textbook *Theoretical Computer Science* [Hro03]. From mathematics, we assume some elementary knowledge of combinatorics and linear algebra. All other concepts and assertions of probability theory, algebra, and number theory are either presented whenever they are needed or surveyed in the Appendix.

The book is divided into eight chapters, including this introduction. In order to support the iterative way of teaching, these chapters are organized as follows. Every chapter opens with a section “Objectives,” in which the motivations, teaching objectives, and relations to topics of the previous chapters are presented. The core of the chapter is dedicated to the formalization, development, and application of the ideas presented in the “Objectives.” For every essential development and achievement, we will pinpoint its relevance to our objectives. We end each chapter with a short summary and outlook. Here the major highlights of the chapter are informally summarized, and the chapter’s relevance to other parts of the book is once again reviewed.

Chapter 2 provides the fundamentals. One learns here what randomized algorithms are, and how to design and analyze them. The core of Chapter 2 begins with Section 2.2, with elementary fundamentals of probability theory, reduced to a simple kernel that is sufficient for our purposes. In Section 2.4 we explain what randomized algorithms are and how to model and analyze them by means of probability theory. Section 2.5 presents the fundamental classification of randomized algorithms with respect to their error probabilities.¹⁵ Section 2.5 shows how to model and classify randomized algorithms in the areas of discrete optimization, where we usually do not speak about error probability but about a probability of getting a good approximation of an optimal solution. From a contextual point of view Section 2.5 is central to this textbook. Here we introduce the most successful and recognized paradigms of the design of randomized algorithms such as “Fooling an Adversary,” “Fingerprinting,” “Amplification,” “Random Sampling,” “Abundance of Witnesses,” and “Random Rounding.” In this way, we start not only to build the methodology and the machinery for the design of efficient and simple randomized algorithms, but also to capture the nature of the fascinating computational power of randomization in many applications. The paradigms introduced here determine the structure of this book because each of the following chapters (apart from the Appendix) is devoted to the study of one of these paradigms.

Chapter 3 provides a deeper insight into the application of the method of fooling an adversary, which is also called the method of eliminating worst-case

¹⁵More precisely, with respect to the speedup of reducing the error probability with the number of independently executed runs of the randomized algorithm on the same input.

problem instances. Here, one views a randomized algorithm as a probability distribution over a set of deterministic algorithms (strategies). The crucial point is creating a set of deterministic strategies such that, for any problem instance, most of these strategies efficiently compute the correct result¹⁶. This can be possible even when there does not exist any efficient deterministic algorithm for solving the problem¹⁷ considered. First, we make this approach transparent by presenting hashing, where universal hashing is nothing other than an application of the method of fooling an adversary. A deeper insight into the power of this method is given by applying it in the area of online algorithms.

The fingerprinting method is successfully applied several times in Chapter 4. The idea of this method is to solve equivalence problems in such a way that instead of trying to compare full complex representations of given objects one compares rather their randomly chosen partial representations called fingerprints. The design of our randomized protocol presented in Section 2.2 can also be viewed as an application of this design paradigm. In Section 4.2 we apply fingerprinting in order to solve other communication problems that can be viewed as generalizations of the equality problem. Section 4.3 uses our motivation example once again in order to design an efficient randomized algorithm for searching for a string (pattern) in a longer string (text). Section 4.4 shows how one can apply fingerprinting in order to verify the correctness of the multiplication of two matrices in a more efficient way than the matrix multiplication.¹⁸ In Section 4.5 we generalize the idea of Section 4.4 in order to develop a polynomial randomized algorithm for deciding the equivalence of two polynomials. This application of fingerprinting is of special importance, because a deterministic polynomial algorithm for this decision problem is not known.

Because amplification and random sampling are often combined, or even indistinguishably mixed, we present them together in Chapter 5. The paradigm of success probability amplification is common to all randomized algorithms and it says that one can increase the success probability of any randomized algorithm by several independent runs of the algorithm on the same input. Section 5.2 shows a more clever application of this paradigm by repeating only some critical parts of a computation instead of repeating all the random runs. Random sampling enables us to create objects with some required properties by a simple random choice from a set of objects, despite the fact that one does not know how to efficiently construct such objects in the deterministic way. In Section 5.3 we combine amplification with random sampling in order to successfully attack the NP-complete satisfiability problem. Section

¹⁶This means that some of these strategies are allowed to compute wrong results on some inputs.

¹⁷I.e., a deterministic algorithm that is correct and efficient on any input.

¹⁸I.e., one can verify whether $A \cdot B = C$ for three matrices A , B , and C without computing $A \cdot B$.

5.4 shows an application of random sampling for efficiently generating non-quadratic residues, which one does not know how to generate deterministically in polynomial time.

Chapter 6 is devoted to the method of abundance of witnesses, which can be viewed as the deepest paradigm of randomization. A witness is additional information to an input, whose knowledge makes a hard problem efficiently solvable. The idea of this method is to generate such witnesses at random. The art of successfully applying this method lies in searching for a suitable kind of witness for the given problem. Here we present a part of such a search for a convenient definition of witnesses for primality testing, which results in the design of efficient randomized primality testing algorithms.

Chapter 7 is devoted to the design of randomized approximation algorithms for the NP-hard maximum satisfiability problem (MAX-SAT). We show how one can round a real solution of the relaxed version of MAX-SAT at random in such a way that a good approximation of an optimal solution to the original discrete optimization problem can be expected.

Appendix A provides some fundamentals of mathematics sufficient for the purposes of the previous chapters. The mathematics is viewed here as a formal language and as a machinery (sets of instruments and methods) for designing, modeling and analyzing randomized algorithms, and it is also presented in this way. Section A.2 provides a short, concise introduction to fundamentals of group theory and number theory. Section A.3 presents some basic facts and methods of combinatorics.

1.4 To the Student

This textbook has been written primarily for you. The aim of this book is not only to introduce and explain some basic methods for the design of efficient randomized algorithms, but also to inspire you for the study of theoretical computer science. In the previous sections of this chapter we have attempted to convince you that randomization is a fascinating area of computer science, because due to randomization one can efficiently perform things that were not considered possible before, and so one can enjoy work on a topic that offers a lot of exciting surprises.

But to teach an exciting topic is not sufficient to fill the lecture room with many interested students. A good didactic presentation of the topic for the success of a course is at least as important as the attractiveness of the subject. Therefore, our presentation of this topic is based on the following three concepts:

(i) *Simplicity and transparency*

We explain simple notions, concepts, and methods in simple terms. We avoid the use of unnecessary mathematical abstractions by attempting to be as concrete as possible. Through this we develop an introduction to the

design of randomized algorithms on elementary mathematical knowledge. Presenting complicated arguments or proofs, we first explain the ideas in a simple and transparent way, and then provide the formal, detailed proofs. Sections and theorems marked with a “*” are more involved and technical. Undergraduates are advised to skip these parts when reading the material for the first time.

Clarity takes priority over the presentation of the best known results. When a transparent argument of a weaker result can bring across the idea succinctly, we opt for it instead of presenting a strong but technically demanding and confusing argument of the best known result.

Throughout this book, we work systematically, taking small steps to journey from the simple to the complicated. We avoid any interruption in thoughts.

(ii) *Less is sometimes more, or a context-sensitive presentation*

Many study guides and textbooks falsely assume that the first and foremost aim is the delivery of a quantum of information to the reader. Hence, they often go down the wrong track: maximum knowledge in minimum time, presented in minimal space. This haste usually results in the presentation of a great amount of individual results, thus neglecting the context of the entire course. The philosophy behind this book is different. We would like to build and influence the student’s way of thinking. Hence, we are not overly concerned about the amount of information, and are prepared to sacrifice 10% to 20% of the teaching material. In return we dedicate more time to the informal ideas, motivations, connections between practice and theory, and, especially, to internal contexts of the presented research area. We place special emphasis on the creation of new terms. The notions and definitions do not appear out of the blue, as seemingly so in some lectures using the formal language of mathematics. The formally defined terms are always an approximation or an abstraction of intuitive ideas. The formalization of these ideas enables us to make accurate statements and conclusions about certain objects and events. They also allow for formal and direct argumentation. We strive to explain our choice of the formalization of terms and models used, and to point out the limitations of their usage. To learn to work on the level of terms creation (basic definitions) is very important, because most of the essential progress happens exactly at this level.

(iii) *Support of iterative teaching*

The strategy of this book is also tailored to cultivate repetitive reconsideration of presented concepts. As already mentioned, every chapter opens with a section “Objectives” in which the objectives are presented in an informal way and in the context of knowledge from the previous chapters. Every essential development in the main body of a chapter is accomplished with a discussion about its importance in the context of already presented knowledge. The conclusion of each chapter informally summarizes its major highlights and weighs its contribution on a contex-

tual level. As usual, the learning process is supported by exercises. The exercises are not allocated to special subsections, but are distributed in the text, with our recommendation to work through them immediately after you have encountered them while reading the book. They help learn how to successfully apply presented concepts and methods and to deepen your understanding of the material.

But the most important point is that this textbook is self-contained, with all formal and informal details presented in the lectures, and so one can use it for a complete review of all explanations given in the teacher's lecture. In fact, one can master the subject of this book by only reading it (without attending the lecture).

1.5 To the Teacher

The aim of this textbook is to support you in creating an introductory course on randomized algorithms. The advantage of this book is that it provides a lot of space for informal development of concepts and ideas, which unfortunately are often presented only orally in lectures, and are not included in the written supporting materials. Therefore, if the teacher followed this book in her or his lecture, the student would have the possibility to review the complete lecture, or a part of it as many times as she or he wanted. Additionally, the students are not required to write all technical details presented during the lecture, and can concentrate on the explanations given by the teacher.

Finally, we allow ourselves to formulate four rules which can be very helpful in inducting a successful course on any topic. All have been very well known for many years (there is no original idea of ours behind them), but teachers often forget about their consistent application, which is the main problem with education quality.

- (i) Make sure that your students can review the topic of your lectures any time and as often as they need to. For instance, you can save the entire presentation on the Internet or write (use) detailed supporting materials.
- (ii) Provide with your lectures, especially if you have many students in the course, one more public discussion hour per week. In this additional hour students may ask anything related to topics already presented. Typically they ask for more careful repetitions of some complex parts of the lecture or for alternative explanations. Anonymous, written questions should be allowed also.
- (iii) Do not save time when one needs to develop concepts and ideas on an informal level or to create new terms. This often underestimated part of the lecture is at least as important as the correct, detailed presentation of results and their proofs. Exactly telling the development of ideas in a scientific discipline in a broad context essentially contributes to a deeper understanding of the subject and motivates the student to deal with the topic.

- (iv) Organize small groups for exercises. Take care in choosing exercises for homework in order to fix and deepen the understanding of the actual topic of your lecture. The solutions of the homework have to be made public before meeting the students for exercises in order to prevent the exercises from becoming a presentation of correct solutions. Alternative solutions and the most frequent mistakes have to be discussed. All homework has to be individually corrected and given back to the students.



<http://www.springer.com/978-3-540-23949-9>

Design and Analysis of Randomized Algorithms

Introduction to Design Paradigms

Hromkovic, J.

2005, XII, 277 p., Hardcover

ISBN: 978-3-540-23949-9